DocCode - SCORE

Placeholder Sheet for Supplemental Content

Application Number: 18779029 Document Date: 07/21/2024

The presence of this form indicates that the following document type was received in electronic format on the date identified above. This document is stored separately from the typical case contents which are stored in image format.

Since this was an electronic submission, there is no physical artifact folder, no artifact folder is recorded and no paper documents or physical media exist.

- Drawing

At the time of document entry (noted above):

- USPTO employees may access this supplemental content via Docket Application Viewer (DAV) or Supplemental Content Viewer (SCV).
- External customers may access this supplemental content via Patent Center.

MICROSOFT CORP.
EXHIBIT 1002

Doc Code: ECOMM.AUTH/ECOMM.WTDW
Doc Description: Internet Communications Authorized/Internet Communications Authorization Withdrawn

PTO/SB/439 (11-15)

AUTHORIZATION FOR INTERNET COMMUNICATIONS IN A PATENT APPLICATION OR REQUEST TO WITHDRAW AUTHORIZATION FOR INTERNET COMMUNICATIONS		
	Application No.	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Practitioner Docket No.	15305FTNK

To: Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

I. To authorize permission for Internet Communications.

☑ Recognizing that Internet communications are not secure, I hereby authorize the USPTO to communicate with the undersigned and practitioners in accordance with 37 CFR 1.33 and 37 CFR 1.34 concerning any subject matter of this application via video conferencing, instant messaging, or electronic mail. I understand that a copy of these communications will be made of record in the application file. (MPEP 502.03)

II. To withdraw authorization for Internet Communications.

☐ The authorization given on _____, to the USPTO to communicate with the undersigned and any practitioner in accordance with 37 CFR 1.33 and 37 CFR 1.34 concerning any subject matter of this application via Internet communications such as video conferencing, instant messaging, or electronic mail is hereby withdrawn. I understand that the withdrawal is effective when approved rather than when received.

I am the

☐ applicant.

☑ attorney or agent of record. Registration number 50966.

☐ attorney or agent acting under 37 CFR 1.34. Registration number _____.

/BRIAN R. GALVIN/ — Signature

July 19, 2024 — Date

Brian R. Galvin — Typed or printed name

(360) 674-6233 — Telephone Number

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. Juristic entities must be represented by a patent practitioner (see 37 CFR 1.31, which is applicable to any paper filed on or after September 16, 2012, that is presented on behalf of a juristic entity, regardless of application filing date). Submit multiple forms if more than one signature is required, see below*.

☑ * Total of 1 forms are submitted.

Privacy Act Statement

The **Privacy Act of 1974 (P.L. 93-579)** requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (*i.e.*, GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

Doc Code: PA..
Document Description: Power of Attorney

PTO/AIA/82B (07-13)
Approved for use through 11/30/2014. OMB 0651-0051
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number

POWER OF ATTORNEY BY APPLICANT

I hereby revoke all previous powers of attorney given in the application identified in either the attached transmittal letter or the boxes below.

Application Number	Filing Date

(Note: The boxes above may be left blank if information is provided on form PTO/AIA/82A.)

☑ I hereby appoint the Patent Practitioner(s) associated with the following Customer Number as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above: 82,318

OR

☐ I hereby appoint Practitioner(s) named in the attached list (form PTO/AIA/82C) as my/our attorney(s) or agent(s), and to transact all business in the United States Patent and Trademark Office connected therewith for the patent application referenced in the attached transmittal letter (form PTO/AIA/82A) or identified above. (Note: Complete form PTO/AIA/82C.)

Please recognize or change the correspondence address for the application identified in the attached transmittal letter or the boxes above to:

☑ The address associated with the above-mentioned Customer Number

OR

☐ The address associated with Customer Number:

OR

Firm or Individual Name					
Address					
City		State		Zip	
Country					
Telephone		Email			

I am the Applicant (if the Applicant is a juristic entity, list the Applicant name in the box):

QOMPLX LLC

☐ Inventor or Joint Inventor (title not required below)

☐ Legal Representative of a Deceased or Legally Incapacitated Inventor (title not required below)

☑ Assignee or Person to Whom the Inventor is Under an Obligation to Assign (provide signer's title if applicant is a juristic entity)

☐ Person Who Otherwise Shows Sufficient Proprietary Interest (e.g., a petition under 37 CFR 1.46(b)(2) was granted in the application or is concurrently being filed with this document) (provide signer's title if applicant is a juristic entity)

SIGNATURE of Applicant for Patent

The undersigned (whose title is supplied below) is authorized to act on behalf of the applicant (e.g., where the applicant is a juristic entity).

Signature	Jason Crabtree	Date (Optional)	
Name	Jason Crabtree		
Title	CEO		

NOTE: Signature - This form must be signed by the applicant in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications. If more than one applicant, use multiple forms.

☑ Total of 1 forms are submitted.

This collection of information is required by 37 CFR 1.131, 1.32, and 1.33. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 3 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

Doc ID: e879b180855e2db05f1c6bd7a0ded5c424589276

PTO/SB/08a (01-22)
Approved for use through 05/31/2024. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (**Not for submission under 37 CFR 1.99)**		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

U.S.PATENTS						Remove
Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	3370192	A	1968-02-20	SCHWARTZ JAMES W et al.	Entire Document
	2	5669000	A	1997-09-16	Jessen et al.	Entire Document
	3	5953011	A	1999-09-14	Matsuoka	Entire Document
	4	6256544	B1	2001-07-03	Weissinger	Entire Document
	5	6477572	B1	2002-11-05	Elderton et al.	Entire Document
	6	6857073	B2	2005-02-15	French et al.	Entire Document
	7	6906709	B1	2005-06-14	Larkin et al.	Entire Document
	8	7072863	B1	2006-07-04	Phillips et al.	Entire Document

	9	7139747	B1	2006-11-21	Najork	Entire Document
	10	7171515	B2	2007-01-30	Ohta et al.	Entire Document
	11	7222366	B2	2007-05-22	Bruton, III et al.	Entire Document
	12	7227948	B2	2007-06-05	Ohkuma et al.	Entire Document
	13	7266821	B2	2007-09-04	Polizzi et al.	Entire Document
	14	7281125	B2	2007-10-09	Challener et al.	Entire Document
	15	7310632	B2	2007-12-18	Meek et al.	Entire Document
	16	7373524	B2	2008-05-13	Motsinger et al.	Entire Document
	17	7437718	B2	2008-10-14	Fournet et al.	Entire Document
	18	7448046	B2	2008-11-04	Navani et al.	Entire Document
	19	7480940	B1	2009-01-20	Agbabian et al.	Entire Document

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	20	7493593	B2	2009-02-17	Koehler	Entire Document
	21	7530105	B2	2009-05-05	Gilbert et al.	Entire Document
	22	7546207	B2	2009-06-09	Nix et al.	Entire Document
	23	7546333	B2	2009-06-09	Alon et al.	Entire Document
	24	7546637	B1	2009-06-09	Agbabian et al.	Entire Document
	25	7603709	B2	2009-10-13	Lewis et al.	Entire Document
	26	7603714	B2	2009-10-13	Johnson et al.	Entire Document
	27	7653188	B2	2010-01-26	Kloberdans et al.	Entire Document
	28	7657406	B2	2010-02-02	Tolone et al.	Entire Document
	29	7660815	B1	2010-02-09	Scofield et al.	Entire Document
	30	7685296	B2	2010-03-23	Brill et al.	Entire Document

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	31	7698213	B2	2010-04-13	Lancaster	Entire Document
	32	7702821	B2	2010-04-20	Feinberg et al.	Entire Document
	33	7739653	B2	2010-06-15	Venolia	Entire Document
	34	7743421	B2	2010-06-22	Cosquer et al.	Entire Document
	35	7774335	B1	2010-08-10	Scofield et al.	Entire Document
	36	7818224	B2	2010-10-19	Boerner	Entire Document
	37	7818417	B2	2010-10-19	Ginis et al.	Entire Document
	38	7840677	B2	2010-11-23	Li et al.	Entire Document
	39	7925561	B2	2011-04-12	Xu	Entire Document
	40	7933926	B2	2011-04-26	Ebert	Entire Document
	41	8006303	B1	2011-08-23	Dennerline et al.	Entire Document

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	42	8055712	B2	2011-11-08	Kagawa et al.	Entire Document
	43	8065257	B2	2011-11-22	Kuecuekyan	Entire Document
	44	8069190	B2	2011-11-29	McColl et al.	Entire Document
	45	8116450	B2	2012-02-14	Agrawal et al.	Entire Document
	46	8132260	B1	2012-03-06	Mayer et al.	Entire Document
	47	8156029	B2	2012-04-10	Szydlo	Entire Document
	48	8205259	B2	2012-06-19	Stute	Entire Document
	49	8209274	B1	2012-06-26	Lin et al.	Entire Document
	50	8245302	B2	2012-08-14	Evans et al.	Entire Document
	51	8346753	B2	2013-01-01	Hayes	Entire Document
	52	8352347	B2	2013-01-08	Howard et al.	Entire Document

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	53	8352412	B2	2013-01-08	Alba et al.	Entire Document
	54	8370192	B2	2013-02-05	Deo et al.	Entire Document
	55	8380843	B2	2013-02-19	Loizeaux et al.	Entire Document
	56	8386519	B2	2013-02-26	Kenedy et al.	Entire Document
	57	8407800	B2	2013-03-26	Schloegel et al.	Entire Document
	58	8417656	B2	2013-04-09	Beg et al.	Entire Document
	59	8457996	B2	2013-06-04	Winkler et al.	Entire Document
	60	8495521	B2	2013-07-23	Fried	Entire Document
	61	8516594	B2	2013-08-20	Bennett et al.	Entire Document
	62	8516596	B2	2013-08-20	Sandoval et al.	Entire Document
	63	8548777	B2	2013-10-01	Sturrock et al.	Entire Document

	64	8566945	B2	2013-10-22	Sima	Entire Document
	65	8583639	B2	2013-11-12	Chitnis et al.	Entire Document
	66	8595240	B1	2013-11-26	Otey et al.	Entire Document
	67	8601554	B2	2013-12-03	Gordon et al.	Entire Document
	68	8601587	B1	2013-12-03	Powell et al.	Entire Document
	69	8607197	B2	2013-12-10	Barcia et al.	Entire Document
	70	8615800	B2	2013-12-24	Baddour et al.	Entire Document
	71	8654127	B2	2014-02-18	Kenttala et al.	Entire Document
	72	8677473	B2	2014-03-18	Dennerline et al.	Entire Document
	73	8707275	B2	2014-04-22	Mascaro et al.	Entire Document
	74	8712596	B2	2014-04-29	Scott	Entire Document

	75	8725597	B2	2014-05-13	Mauseth et al.	Entire Document
	76	8726393	B2	2014-05-13	Macy et al.	Entire Document
	77	8751867	B2	2014-06-10	Marvasti et al.	Entire Document
	78	8752178	B2	2014-06-10	Coates et al.	Entire Document
	79	8781990	B1	2014-07-15	de Alfaro et al.	Entire Document
	80	8782080	B2	2014-07-15	Lee et al.	Entire Document
	81	8788306	B2	2014-07-22	Delurgio et al.	Entire Document
	82	8793758	B2	2014-07-29	Raleigh et al.	Entire Document
	83	8806361	B1	2014-08-12	Noel et al.	Entire Document
	84	8813234	B1	2014-08-19	Bowers et al.	Entire Document
	85	8819772	B2	2014-08-26	Bettini et al.	Entire Document

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	86	8826426	B1	2014-09-02	Dubey	Entire Document
	87	8839440	B2	2014-09-16	Yun et al.	Entire Document
	88	8897900	B2	2014-11-25	Smith et al.	Entire Document
	89	8898442	B2	2014-11-25	Stoitsev	Entire Document
	90	8914878	B2	2014-12-16	Burns et al.	Entire Document
	91	8949960	B2	2015-02-03	Berkman et al.	Entire Document
	92	8959494	B2	2015-02-17	Howard	Entire Document
	93	8990392	B1	2015-03-24	Stamos	Entire Document
	94	8997233	B2	2015-03-31	Green et al.	Entire Document
	95	9009837	B2	2015-04-14	Nunez Di Croce	Entire Document
	96	9015708	B2	2015-04-21	Choudhury et al.	Entire Document

	97	9021477	B2	2015-04-28	Choudhury et al.	Entire Document
	98	9031870	B2	2015-05-12	Kenedy et al.	Entire Document
	99	9043332	B2	2015-05-26	Noel et al.	Entire Document
	100	9049207	B2	2015-06-02	Hugard, IV et al.	Entire Document
	101	9069725	B2	2015-06-30	Jones	Entire Document
	102	9092616	B2	2015-07-28	Kumar et al.	Entire Document
	103	9100430	B1	2015-08-04	Seiver et al.	Entire Document
	104	9110706	B2	2015-08-18	Yu et al.	Entire Document
	105	9129108	B2	2015-09-08	Drissi et al.	Entire Document
	106	9134966	B2	2015-09-15	Brock et al.	Entire Document
	107	9137024	B2	2015-09-15	Swingler et al.	Entire Document

	108	9152727	B1	2015-10-06	Balducci et al.	Entire Document
	109	9166990	B2	2015-10-20	Eswaran et al.	Entire Document
	110	9171079	B2	2015-10-27	Banka et al.	Entire Document
	111	9185124	B2	2015-11-10	Chakraborty	Entire Document
	112	9202040	B2	2015-12-01	Rosenblatt et al.	Entire Document
	113	9203827	B2	2015-12-01	Srinivasan et al.	Entire Document
	114	9210185	B1	2015-12-08	Pinney Wood et al.	Entire Document
	115	9231962	B1	2016-01-05	Yen et al.	Entire Document
	116	9235732	B2	2016-01-12	Eynon et al.	Entire Document
	117	9253643	B2	2016-02-02	Pattar et al.	Entire Document
	118	9256735	B2	2016-02-09	Stute	Entire Document

	119	9262787	B2	2016-02-16	Binion et al.	Entire Document
	120	9264395	B1	2016-02-16	Stamos	Entire Document
	121	9276951	B2	2016-03-01	Choi et al.	Entire Document
	122	9286103	B2	2016-03-15	Acharya et al.	Entire Document
	123	9292692	B2	2016-03-22	Wallrabenstein	Entire Document
	124	9292699	B1	2016-03-22	Stuntebeck et al.	Entire Document
	125	9294498	B1	2016-03-22	Yampolskiy et al.	Entire Document
	126	9300682	B2	2016-03-29	Burnham et al.	Entire Document
	127	9319430	B2	2016-04-19	Bell, Jr. et al.	Entire Document
	128	9336481	B1	2016-05-10	Ionson	Entire Document
	129	9338061	B2	2016-05-10	Chen et al.	Entire Document

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	130	9344444	B2	2016-05-17	Lippmann et al.	Entire Document
	131	9349103	B2	2016-05-24	Eberhardt, III et al.	Entire Document
	132	9369482	B2	2016-06-14	Borohovski et al.	Entire Document
	133	9384345	B2	2016-07-05	Dixon et al.	Entire Document
	134	9390376	B2	2016-07-12	Harrison et al.	Entire Document
	135	9400962	B2	2016-07-26	Prasad	Entire Document
	136	9438616	B2	2016-09-06	Singla et al.	Entire Document
	137	9461876	B2	2016-10-04	Van Dusen et al.	Entire Document
	138	9466041	B2	2016-10-11	Simitsis et al.	Entire Document
	139	9467461	B2	2016-10-11	Balderas	Entire Document
	140	9495188	B1	2016-11-15	Ettema et al.	Entire Document

	141	9501647	B2	2016-11-22	Yampolskiy et al.	Entire Document
	142	9503467	B2	2016-11-22	Lefebvre et al.	Entire Document
	143	9503472	B2	2016-11-22	Laidlaw et al.	Entire Document
	144	9509716	B2	2016-11-29	Shabtai et al.	Entire Document
	145	9515826	B2	2016-12-06	Whelan et al.	Entire Document
	146	9516053	B1	2016-12-06	Muddu et al.	Entire Document
	147	9521166	B2	2016-12-13	Wilson	Entire Document
	148	9541982	B2	2017-01-10	Lipasti et al.	Entire Document
	149	9558220	B2	2017-01-31	Nixon et al.	Entire Document
	150	9560065	B2	2017-01-31	Neil et al.	Entire Document
	151	9565204	B2	2017-02-07	Chesla	Entire Document

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	152	9571517	B2	2017-02-14	Vallone et al.	Entire Document
	153	9578046	B2	2017-02-21	Baker	Entire Document
	154	9596141	B2	2017-03-14	McDowall	Entire Document
	155	9600792	B2	2017-03-21	Foehr et al.	Entire Document
	156	9602513	B2	2017-03-21	Gamage et al.	Entire Document
	157	9602529	B2	2017-03-21	Jones et al.	Entire Document
	158	9602530	B2	2017-03-21	Ellis et al.	Entire Document
	159	9609009	B2	2017-03-28	Muddu et al.	Entire Document
	160	9609015	B2	2017-03-28	Natarajan et al.	Entire Document
	161	9619291	B2	2017-04-11	Pueyo et al.	Entire Document
	162	9639575	B2	2017-05-02	Leida et al.	Entire Document

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	163	9652538	B2	2017-05-16	Shivaswamy et al.	Entire Document
	164	9652604	B1	2017-05-16	Johansson et al.	Entire Document
	165	9654495	B2	2017-05-16	Hubbard et al.	Entire Document
	166	9661019	B2	2017-05-23	Liu	Entire Document
	167	9667600	B2	2017-05-30	Piqueras Jover et al.	Entire Document
	168	9667641	B2	2017-05-30	Muddu et al.	Entire Document
	169	9672283	B2	2017-06-06	Pappas et al.	Entire Document
	170	9672355	B2	2017-06-06	Titonis et al.	Entire Document
	171	9674211	B2	2017-06-06	Curcic et al.	Entire Document
	172	9674249	B1	2017-06-06	Kekre et al.	Entire Document
	173	9679125	B2	2017-06-13	Bailor et al.	Entire Document

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	174	9680867	B2	2017-06-13	Hughes et al.	Entire Document
	175	9686293	B2	2017-06-20	Golshan et al.	Entire Document
	176	9690645	B2	2017-06-27	Samuni et al.	Entire Document
	177	9699205	B2	2017-07-04	Muddu et al.	Entire Document
	178	9712553	B2	2017-07-18	Nguyen et al.	Entire Document
	179	9721086	B2	2017-08-01	Shear et al.	Entire Document
	180	9729421	B2	2017-08-08	Brech et al.	Entire Document
	181	9729538	B2	2017-08-08	Plotnik et al.	Entire Document
	182	9734169	B2	2017-08-15	Redlich et al.	Entire Document
	183	9734220	B2	2017-08-15	Karpi{hacek over (s)}t{hacek over (s)}enko et al.	Entire Document
	184	9736173	B2	2017-08-15	Li et al.	Entire Document

	185	9749343	B2	2017-08-29	Watters et al.	Entire Document
	186	9749344	B2	2017-08-29	Watters et al.	Entire Document
	187	9753796	B2	2017-09-05	Mahaffey et al.	Entire Document
	188	9756067	B2	2017-09-05	Boyadjiev et al.	Entire Document
	189	9762443	B2	2017-09-12	Dickey	Entire Document
	190	9771225	B2	2017-09-26	Stone et al.	Entire Document
	191	9772934	B2	2017-09-26	Maag et al.	Entire Document
	192	9774407	B2	2017-09-26	Hudson et al.	Entire Document
	193	9774522	B2	2017-09-26	Vasseur et al.	Entire Document
	194	9774616	B2	2017-09-26	Flores et al.	Entire Document
	195	9781144	B1	2017-10-03	Otvagin et al.	Entire Document

	196	9807104	B1	2017-10-31	Sarra	Entire Document
	197	9832213	B2	2017-11-28	Underwood et al.	Entire Document
	198	9842000	B2	2017-12-12	Bishop et al.	Entire Document
	199	9858322	B2	2018-01-02	Theimer et al.	Entire Document
	200	9860208	B1	2018-01-02	Ettema et al.	Entire Document
	201	9875360	B1	2018-01-23	Grossman et al.	Entire Document
	202	9882929	B1	2018-01-30	Ettema et al.	Entire Document
	203	9887933	B2	2018-02-06	Lawrence, III	Entire Document
	204	9910993	B2	2018-03-06	Grossman et al.	Entire Document
	205	9911088	B2	2018-03-06	Nath et al.	Entire Document
	206	9917860	B2	2018-03-13	Senanayake et al.	Entire Document

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	207	9928366	B2	2018-03-27	Ladnai et al.	Entire Document
	208	9930058	B2	2018-03-27	Carpenter et al.	Entire Document
	209	9942295	B2	2018-04-10	Rider et al.	Entire Document
	210	9946517	B2	2018-04-17	Talby et al.	Entire Document
	211	9952899	B2	2018-04-24	Novaes	Entire Document
	212	9954879	B1	2018-04-24	Sadaghiani et al.	Entire Document
	213	9954884	B2	2018-04-24	Hassell et al.	Entire Document
	214	9965627	B2	2018-05-08	Ray et al.	Entire Document
	215	9967264	B2	2018-05-08	Harris et al.	Entire Document
	216	9967265	B1	2018-05-08	Peer et al.	Entire Document
	217	9967282	B2	2018-05-08	Thomas et al.	Entire Document

	218	9967283	B2	2018-05-08	Ray et al.	Entire Document
	219	9967625	B2	2018-05-08	Korst et al.	Entire Document
	220	9984129	B2	2018-05-29	Patel et al.	Entire Document
	221	9992228	B2	2018-06-05	Ray et al.	Entire Document
	222	10009378	B2	2018-06-26	Chiviendacz et al.	Entire Document
	223	10027711	B2	2018-07-17	Gill et al.	Entire Document
	224	10038559	B2	2018-07-31	Burrows et al.	Entire Document
	225	10044675	B1	2018-08-07	Ettema et al.	Entire Document
	226	10050985	B2	2018-08-14	Mhatre et al.	Entire Document
	227	10055473	B2	2018-08-21	Allen et al.	Entire Document
	228	10061635	B2	2018-08-28	Ellwein	Entire Document

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	229	10074052		2018-09-11	Banerjee et al.	Entire Document
	230	10078664	B2	2018-09-18	Gustafson et al.	Entire Document
	231	10083236	B2	2018-09-25	Crosby	Entire Document
	232	10102480	B2	2018-10-16	Dirac et al.	Entire Document
	233	10108907	B2	2018-10-23	Bugay et al.	Entire Document
	234	10109014	B1	2018-10-23	Bischoff et al.	Entire Document
	235	10110415	B2	2018-10-23	Radivojevic et al.	Entire Document
	236	10120907	B2	2018-11-06	de Castro Alves et al.	Entire Document
	237	10122687	B2	2018-11-06	Thomas et al.	Entire Document
	238	10122764	B1	2018-11-06	Obaidi	Entire Document
	239	10146592	B2	2018-12-04	Bishop et al.	Entire Document

	240	10152676	B1	2018-12-11	Strom	Entire Document
	241	10154049	B2	2018-12-11	Sancheti et al.	Entire Document
	242	10162969	B2	2018-12-25	Knapp	Entire Document
	243	10168691	B2	2019-01-01	Zornio et al.	Entire Document
	244	10180780	B2	2019-01-15	Ainalem	Entire Document
	245	10185832	B2	2019-01-22	Cam	Entire Document
	246	10191768	B2	2019-01-29	Bishop et al.	Entire Document
	247	10205735	B2	2019-02-12	Apostolopoulos	Entire Document
	248	10210246	B2	2019-02-19	Stojanovic et al.	Entire Document
	249	10210470	B2	2019-02-19	Partha Ray	Entire Document
	250	10212176	B2	2019-02-19	Wang	Entire Document

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	251	10212184	B2	2019-02-19	Sweeney et al.	Entire Document
	252	10216485	B2	2019-02-26	Misra et al.	Entire Document
	253	10217348	B2	2019-02-26	Poder et al.	Entire Document
	254	10261763	B2	2019-04-16	Fink et al.	Entire Document
	255	10275545	B2	2019-04-30	Yeager et al.	Entire Document
	256	10277629	B1	2019-04-30	Guntur	Entire Document
	257	10284570	B2	2019-05-07	Schmidtler et al.	Entire Document
	258	10289841	B2	2019-05-14	Tang et al.	Entire Document
	259	10290141	B2	2019-05-14	Kennedy et al.	Entire Document
	260	10298607	B2	2019-05-21	Tang et al.	Entire Document
	261	10305902	B2	2019-05-28	Kim	Entire Document

	262	10318739	B2	2019-06-11	Brucker et al.	Entire Document
	263	10318882	B2	2019-06-11	Brueckner et al.	Entire Document
	264	10320828	B1	2019-06-11	Derbeko et al.	Entire Document
	265	10321278	B2	2019-06-11	Proctor	Entire Document
	266	10324773	B2	2019-06-18	Wing et al.	Entire Document
	267	10338913	B2	2019-07-02	Franchitti	Entire Document
	268	10367829	B2	2019-07-30	Huang et al.	Entire Document
	269	10380140	B2	2019-08-13	Sherman	Entire Document
	270	10387124	B2	2019-08-20	Chaudhuri et al.	Entire Document
	271	10387631	B2	2019-08-20	Duggal et al.	Entire Document
	272	10410113	B2	2019-09-10	Clayton et al.	Entire Document

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	273	10410214	B2	2019-09-10	Doyle	Entire Document
	274	10438001	B1	2019-10-08	Hariprasad	Entire Document
	275	10440054	B2	2019-10-08	Robertson	Entire Document
	276	10445482	B2	2019-10-15	Ren	Entire Document
	277	10452664	B2	2019-10-22	Le Mouel et al.	Entire Document
	278	10462112	B1	2019-10-29	Makmel et al.	Entire Document
	279	10505954	B2	2019-12-10	Stokes, III et al.	Entire Document
	280	10511498	B1	2019-12-17	Narayan et al.	Entire Document
	281	10515062	B2	2019-12-24	Tidwell et al.	Entire Document
	282	10515366	B1	2019-12-24	Gorelik et al.	Entire Document
	283	10530796	B2	2020-01-07	Patterson et al.	Entire Document

	284	10540624	B2	2020-01-21	Hui et al.	Entire Document
	285	10579691	B2	2020-03-03	Levine et al.	Entire Document
	286	10601854	B2	2020-03-24	Lokamathe et al.	Entire Document
	287	10606454	B2	2020-03-31	Pani	Entire Document
	288	10609059	B2	2020-03-31	Apostolopoulos	Entire Document
	289	10628578	B2	2020-04-21	Eksten et al.	Entire Document
	290	10643144	B2	2020-05-05	Bowers et al.	Entire Document
	291	10645086	B1	2020-05-05	Hadler	Entire Document
	292	10645100	B1	2020-05-05	Wang et al.	Entire Document
	293	10673880	B1	2020-06-02	Pratt et al.	Entire Document
	294	10715534	B2	2020-07-14	Sander et al.	Entire Document

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	295	10740358	B2	2020-08-11	Chan et al.	Entire Document
	296	10764321	B2	2020-09-01	Bower, III et al.	Entire Document
	297	10776847	B1	2020-09-15	Comar et al.	Entire Document
	298	10789367	B2	2020-09-29	Joseph Durairaj et al.	Entire Document
	299	10791131	B2	2020-09-29	Nor et al.	Entire Document
	300	10817530	B2	2020-10-27	Siebel et al.	Entire Document

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS Remove

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1					

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2] ¡	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	1	2930026	CA	A1	2015-05-14	Theimer et al.	Entire Document	☐
	2	2014159150	WO	A1	2014-10-02	Jubran et al.	Entire Document	☐
	3	2015089463	WO	A1	2015-06-18	Gupta et al.	Entire Document	☐
	4	2020079685	WO	A1	2020-04-23	Atsmon et al.	Entire Document	☐

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]
	1	BOUKHTOUTA, et al, "Graph-theoretic characterization of cyber-threat infrastructures", Digital Investigation, 2015, vol 14, pg. S3-S15, USA	☐
	2	CUI, et al, "Non-intrusive process-based monitoring system to mitigate and prevent VM vulnerability explorations", Collaboratecom, 2013, Austin, USA	☐
	3	EKELHART, et al, "Integrating attacker behavior in IT security analysis: a discrete-event simulation approach", ResearchGate, 2015	☐
	4	FISK, VARGHESE, "Agile and Scalable Analysis of Network Events"	☐
	5	GEDIK et al, "Elastic Scaling for Data Stream Processing", IEEE Transactions on Parallel and Distributed Systems, June 2014, vol. 25, no. 6, pg. 1447-1463	☐

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	6	JAJODIA, et al, "Advanced Cyber Attack Modeling, Analysis, and Visualization", AFRL Final Technical Report, March 2010, USA	☐
	7	JPCERT-CC, "Detecting Lateral Movement through Tracking Event Logs (Version 2)", 2017, pg. 1-16, Japan	☐
	8	JUNGLES et al, "Mitigating Pass-the-Hash (PtH) Attacks and Other Credential Theft Techniques", TwC Next, 2012	☐
	9	KBAR, "Wireless Network Token-Based Fast Authentication", 17th International Conference on Telecommunications, 2010, pg. 227-233	☐
	10	KIESLING et al, "Selecting security control portfolios: a multi-objective simulation-optimization approach", EURO Journal on Decision Processes, April 2016	☐
	11	KOTENKO, "A Cyber Attack Modeling and Impact Assessment framework", Conference Paper, January 2013	☐
	12	KUMAR, et al, "DFuse: A Framework for Distributed Data Fusion", Georgia Institute of Technology	☐
	13	LU, et al, "Sybil Attack Detection through Global Topology Pattern Visualization", 2011	☐
	14	PASQUALETTI et al, "Attack Detection and Identification in Cyber-Physical Systems", IEEE Transactions on Automatic Control, vol. 58, no. 11, pg. 2715-2729	☐
	15	Patapanchala, "Exploring Security Metrics for Electric Grid Infrastructure Leveraging Attack Graphs", Oregon State Thesis, 2016	☐
	16	SHANDILYA et al, "Use of Attack Graphs in Security Systems", Journal of Computer Networks and Communications, vol. 2014	☐

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	17	YANG et al, "Attack Projection", Advances in Information Security 62, 2014, pg. 239-261	☐
	18	ZARGAR et al, "XABA: A Zero-Knowledge Anomaly-Based Behavioral Analysis Method to Detect Insider Threats", 2016	☐

If you wish to add additional non-patent literature document citation information please click the Add button Add

EXAMINER SIGNATURE

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

CERTIFICATION STATEMENT

Please see 37 CFR 1.97(e) to make the appropriate selection(s):

☐ That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

☐ See attached certification statement.

☒ A certification statement is not submitted herewith.

FEE

☐ The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

COPIES

☐ An identification of an earlier application pursuant to 37 CFR 1.98 (d) (1) is attached.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/BRIAN R. GALVIN/	Date (YYYY-MM-DD)	2024-07-19
Name/Print	Brian R. Galvin	Registration Number	50966

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these record s.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

PTO/SB/08a (01-22)
Approved for use through 05/31/2024. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT
(Not for submission under 37 CFR 1.99)

Application Number	
Filing Date	2024-07-21
First Named Inventor	Jason Crabtree
Art Unit	
Examiner Name	
Attorney Docket Number	15305FTNK

U.S.PATENTS Remove

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	10861028	B2	2020-12-08	Silberman et al.	Entire Document
	2	10862916	B2	2020-12-08	Hittel et al.	Entire Document
	3	10871951	B2	2020-12-22	Ding et al.	Entire Document
	4	10911470	B2	2021-02-02	Muddu et al.	Entire Document
	5	10944772	B2	2021-03-09	Mulchandani et al.	Entire Document
	6	10958667	B1	2021-03-23	Maida et al.	Entire Document
	7	10965711	B2	2021-03-30	Schiappa et al.	Entire Document
	8	10977551	B2	2021-04-13	Van Seijen et al.	Entire Document

	9	10985997	B2	2021-04-20	Duggal et al.	Entire Document
	10	10992698	B2	2021-04-27	Patel et al.	Entire Document
	11	11030520	B2	2021-06-08	Mankovskii et al.	Entire Document
	12	11032307	B2	2021-06-08	Tsironis	Entire Document
	13	11113667	B1	2021-09-07	Jiang et al.	Entire Document
	14	11138514	B2	2021-10-05	Hu et al.	Entire Document
	15	11194900	B2	2021-12-07	Loman et al.	Entire Document
	16	11256791	B2	2022-02-22	Douglas et al.	Entire Document
	17	11334831	B2	2022-05-17	Abu El Ata et al.	Entire Document
	18	11392875	B2	2022-07-19	Carstens et al.	Entire Document
	19	11477641	B2	2022-10-18	Damlaj et al.	Entire Document

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	20	11539663	B2	2022-12-27	Chasman et al.	Entire Document
	21	11574206	B2	2023-02-07	Butler, Jr. et al.	Entire Document
	22	11736299	B2	2023-08-22	Cerna, Jr.	Entire Document

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS Remove

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	20040255167	A1	2004-12-16	Knight, James Michael	Entire Document
	2	20050071223	A1	2005-03-31	Jain, Vivek et al.	Entire Document
	3	20050165822	A1	2005-07-28	Yeung, Simon D. et al.	Entire Document
	4	20050198099	A1	2005-09-08	Motsinger, David Lee et al.	Entire Document
	5	20050289072	A1	2005-12-29	Sabharwal, Vinay	Entire Document
	6	20060149575	A1	2006-07-06	Varadarajan et al.	Entire Document

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	7	20070055558	A1	2007-03-08	Shanahan et al.	Entire Document
	8	20070136821	A1	2007-06-14	Hershaft et al.	Entire Document
	9	20070150744	A1	2007-06-28	Cheng et al.	Entire Document
	10	20070168370	A1	2007-07-19	Hardy	Entire Document
	11	20070276714	A1	2007-11-29	Beringer	Entire Document
	12	20080021866	A1	2008-01-24	HINTON et al.	Entire Document
	13	20080270203	A1	2008-10-30	Holmes et al.	Entire Document
	14	20090012760	A1	2009-01-08	SCHUNEMANN	Entire Document
	15	20090094372	A1	2009-04-09	Nyang et al.	Entire Document
	16	20090199002	A1	2009-08-06	Erickson	Entire Document
	17	20090319247	A1	2009-12-24	Ratcliffe, III et al.	Entire Document

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	18	20100083240	A1	2010-04-01	Siman	Entire Document
	19	20100115276	A1	2010-05-06	Betouin et al.	Entire Document
	20	20100275183	A1	2010-10-28	Panicker et al.	Entire Document
	21	20100299651	A1	2010-11-25	FAINEKOS et al.	Entire Document
	22	20100325685	A1	2010-12-23	Sanbower	Entire Document
	23	20110087888	A1	2011-04-14	Rennie	Entire Document
	24	20110225287	A1	2011-09-15	DALAL et al.	Entire Document
	25	20110307467	A1	2011-12-15	Severance	Entire Document
	26	20120215575	A1	2012-08-23	Deb et al.	Entire Document
	27	20120296845	A1	2012-11-22	Andrews et al.	Entire Document
	28	20130046751	A1	2013-02-21	Tsiatsis et al.	Entire Document

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	29	20130117831	A1	2013-05-09	Hook et al.	Entire Document
	30	20130132149	A1	2013-05-23	Wei et al.	Entire Document
	31	20130159219	A1	2013-06-20	Pantel et al.	Entire Document
	32	20140082729	A1	2014-03-20	Shim et al.	Entire Document
	33	20140149186	A1	2014-05-29	Flaxer et al.	Entire Document
	34	20140244612	A1	2014-08-28	Bhasin et al.	Entire Document
	35	20140279762	A1	2014-09-18	Xaypanya et al.	Entire Document
	36	20140324521	A1	2014-10-30	Mun	Entire Document
	37	20150081363	A1	2015-03-19	Taylor et al.	Entire Document
	38	20150128258	A1	2015-05-07	Novozhenets	Entire Document
	39	20150149979	A1	2015-05-28	Talby et al.	Entire Document

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	40	20150161738	A1	2015-06-11	Stempora	Entire Document
	41	20150170053	A1	2015-06-18	Miao	Entire Document
	42	20150172311	A1	2015-06-18	Freedman et al.	Entire Document
	43	20150242509	A1	2015-08-27	Pall et al.	Entire Document
	44	20150281225	A1	2015-10-01	Schoen et al.	Entire Document
	45	20150295775	A1	2015-10-15	Dickey	Entire Document
	46	20150317745	A1	2015-11-05	Collins et al.	Entire Document
	47	20150347414	A1	2015-12-03	Xiao et al.	Entire Document
	48	20150379111	A1	2015-12-31	Hwang	Entire Document
	49	20160004858	A1	2016-01-07	Chen et al.	Entire Document
	50	20160006629	A1	2016-01-07	Ianakiev et al.	Entire Document

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	51	20160012235	A1	2016-01-14	Lee et al.	Entire Document
	52	20160057159	A1	2016-02-25	Yin et al.	Entire Document
	53	20160088000	A1	2016-03-24	Siva Kumar et al.	Entire Document
	54	20160099960	A1	2016-04-07	Gerritz et al.	Entire Document
	55	20160119365	A1	2016-04-28	Barel	Entire Document
	56	20160140519	A1	2016-05-19	Trepca et al.	Entire Document
	57	20160180240	A1	2016-06-23	MAJUMDAR et al.	Entire Document
	58	20160219066	A1	2016-07-28	Vasseur et al.	Entire Document
	59	20160275123	A1	2016-09-22	LIN et al.	Entire Document
	60	20160330233	A1	2016-11-10	Hart	Entire Document
	61	20160364307	A1	2016-12-15	Garg et al.	Entire Document

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	62	20160371363	A1	2016-12-22	MURO et al.	Entire Document
	63	20170010589	A1	2017-01-12	de Anda Fast	Entire Document
	64	20170023509	A1	2017-01-26	KIM et al.	Entire Document
	65	20170090893	A1	2017-03-30	Aditya et al.	Entire Document
	66	20170207926	A1	2017-07-20	GIL et al.	Entire Document
	67	20170241791	A1	2017-08-24	Madigan et al.	Entire Document
	68	20180268264	A1	2018-09-20	Marwah et al.	Entire Document
	69	20180336250	A1	2018-11-22	LLAVES et al.	Entire Document
	70	20190188797	A1	2019-06-20	Przechocki et al.	Entire Document
	71	20200004752	A1	2020-01-02	MAJUMDAR et al.	Entire Document
	72	20200177618	A1	2020-06-04	Hassanzadeh et al.	Entire Document

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	73	20200304534	A1	2020-09-24	RAKESH et al.	Entire Document
	74	20200356664	A1	2020-11-12	Maor	Entire Document
	75	20200364346	A1	2020-11-19	Gourisetti et al.	Entire Document
	76	20200396246	A1	2020-12-17	Zoldi et al.	Entire Document
	77	20210075822	A1	2021-03-11	CHUNG et al.	Entire Document

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2]	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1							☐

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]
	1		☐

If you wish to add additional non-patent literature document citation information please click the Add button Add

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

EXAMINER SIGNATURE			
Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

CERTIFICATION STATEMENT

Please see 37 CFR 1.97(e) to make the appropriate selection(s):

☐ That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

☐ See attached certification statement.

☒ A certification statement is not submitted herewith.

FEE

☐ The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

COPIES

☐ An identification of an earlier application pursuant to 37 CFR 1.98 (d) (1) is attached.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/BRIAN R. GALVIN/	Date (YYYY-MM-DD)	2024-07-19
Name/Print	Brian R. Galvin	Registration Number	50966

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these record s.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
23 April 2020 (23.04.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/079685 A1

(51) International Patent Classification:
G05D 1/00 (2006.01) *G06T 17/05* (2011.01)
G06T 7/579 (2017.01)

(21) International Application Number:
PCT/IL2019/051119

(22) International Filing Date:
15 October 2019 (15.10.2019)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
62/746,607 17 October 2018 (17.10.2018) US

(71) Applicant: **COGNATA LTD.** [EVIL]; 10 HaRav Yaakov Berman Street, 7621316 Rehovot (IL).

(72) Inventors: **ATSMON, Dan;** 10 HaRav Yaakov Berman Street, 7621316 Rehovot (IL). **ASA, Eran;** 20 HaRav Yeshayahu Meshorer Street, 4931925 Petach-Tikva (IL). **SPIEGEL, Ehud;** 62 Hafetz Haim Street, 4935076 Petach-Tikva (IL).

(74) Agent: **EHRLICH, Gal** et al.; G.E. Ehrlich (1995) LTD., 11 Menachem Begin Road, 5268104 Ramat-Gan (IL).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA,

(54) **Title:** SYSTEM AND METHOD FOR GENERATING REALISTIC SIMULATION DATA FOR TRAINING AN AUTONOMOUS DRIVER

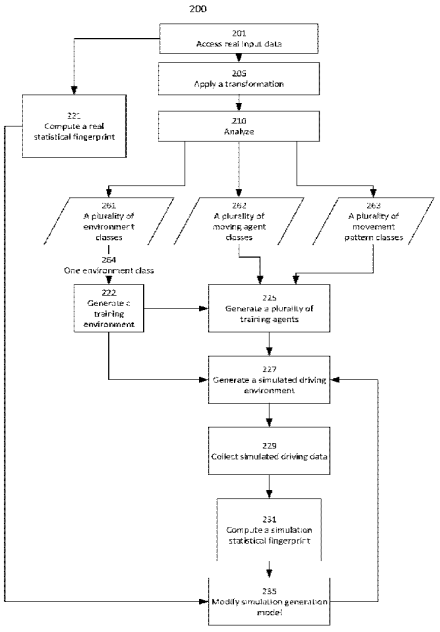


FIG. 2

(57) **Abstract:** A method for training a model for generating simulation data for training an autonomous driving agent, comprising: analyzing real data, collected from a driving environment, to identify a plurality of environment classes, a plurality of moving agent classes, and a plurality of movement pattern classes; generating a training environment, according to one environment class; and in at least one training iteration: generating, by a simulation generation model, a simulated driving environment according to the training environment and according to a plurality of generated training agents, each associated with one of the plurality of agent classes and one of the plurality of movement pattern classes; collecting simulated driving data from the simulated environment; and modifying at least one model parameter of the simulation generation model to minimize a difference between a simulation statistical fingerprint, computed using the simulated driving data, and a real statistical fingerprint, computed using the real data.

WO 2020/079685 A1

[Continued on next page]

SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *of inventorship (Rule 4.17(iv))*

Published:

— *with international search report (Art. 21(3))*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

SYSTEM AND METHOD FOR GENERATING REALISTIC SIMULATION DATA FOR TRAINING AN AUTONOMOUS DRIVER

RELATED APPLICATION/S

This application claims the benefit of priority of U.S. Provisional Patent Application No. 62/746,607 filed on 17 October 2018, the contents of which are incorporated herein by reference in their entirety.

BACKGROUND

The present invention, in some embodiments thereof, relates to creating a simulated model of a geographical area, and, more specifically, but not exclusively, to creating a simulated model of a geographical area, optionally including vehicle traffic to generate simulation sensory data for training an autonomous driving system.

When generating simulated data for training an autonomous driving system there is a need to generate data that exhibits realistic behavior and a realistic look (for example for visual sensors).

For a simulated environment that exhibits realistic behavior for training autonomous driving there is a need to simulate agents representing other moving objects. Each of these moving objects is a vehicle or a pedestrian with a driver following some driving pattern that governs the moving object's behavior in the simulation (for example: a reckless driver tends to keep shorter distances from a vehicle in front than a sedate driver does). Training an autonomous driving system may comprise providing the autonomous driving system simulation sensory data comprising the agents representing other moving objects and analyzing the autonomous driving system's actions in response to the simulation sensory data.

In some systems for training an autonomous driving system the simulation sensory data comprises a plurality of images. A video is a sequence of images, that is a plurality of images arranged in a sequence. A photo-realistic synthetic image is an image that looks as though it were photographed by a camera. A rendering engine can generate a synthetic image according to semantic descriptions of the required image. However, many images generated by a rendering engine do not appear photo-realistic. Some characteristics that cause an image to appear non-realistic are: color saturation in some areas of the image; gradients (or lack thereof) in the color of the sky, a road, or any other surface in the image; and lighting and shading. There is a need to generate photo-realistic looking synthetic environments for a variety of applications. Training a machine learning model of an autonomous driving system is one example where there is a need for a photo-realistic synthetic environment. Other examples are video games, computer generated

animation movies, and computer enhanced movies. A possible way to generate realistic looking images is to use an image refining model to refine synthetic images generated by a simulation generation model (referred to above as a rendering engine).

SUMMARY

It is an object of the present invention to provide a system and a method for generating simulation data and more specifically, but not exclusively, generating simulation data for the purpose of training an autonomous driving system.

The foregoing and other objects are achieved by the features of the independent claims. Further implementation forms are apparent from the dependent claims, the description and the figures.

According to a first aspect of the invention, a method for training a model for generating simulation data for training an autonomous driving agent comprises: analyzing real input data, collected from a driving environment, to identify a plurality of environment classes, a plurality of moving agent classes, and a plurality of movement pattern classes; generating a training environment, according to one environment class of the plurality of environment classes; and in at least one of a plurality of training iterations: generating, by a simulation generation model, a simulated driving environment according to the training environment and according to a plurality of generated training agents, each associated with one of the plurality of moving agent classes and one of the plurality of movement pattern classes; collecting simulated driving data from the simulated driving environment; and modifying at least one model parameter of the simulation generation model to minimize a difference between a simulation statistical fingerprint, computed using the simulated driving data, and a real statistical fingerprint, computed using the real input data.

According to a second aspect of the invention, a system for training a model for generating simulation data for training an autonomous driving agent comprises at least one hardware processor adapted to: analyzing real input data, collected from a driving environment, to identify a plurality of environment classes, a plurality of moving agent classes, and a plurality of movement pattern classes; generating a training environment, according to one environment class of the plurality of environment classes; in each of a plurality of training iterations: generating, by a simulation generation model, a simulated driving environment according to a plurality of generated training agents, each associated with one of the plurality of moving agent classes and one of the plurality of movement pattern classes; collecting simulated driving data from the simulated driving environment; and modifying a plurality of model parameters of the simulation

generation model to minimize a difference between a simulation statistical fingerprint, computed using the simulated driving data, and a real statistical fingerprint, computed using the real input data.

According to a third aspect of the invention, a method for creating data for training an autonomous driving agent comprises: accessing a simulation generation model trained by: analyzing real input data, collected from a driving environment, to identify a plurality of environment classes, a plurality of moving agent classes, and a plurality of movement pattern classes; generating a training environment, according to one environment class of the plurality of environment classes; in each of a plurality of training iterations: generating, by the simulation generation model, a simulated training driving environment according to the training environment and according to a plurality of generated training agents, each associated with one of the plurality of moving agent classes and one of the plurality of movement pattern classes; collecting simulated driving data from the simulated training driving environment; and modifying a plurality of model parameters of the simulation generation model to minimize a difference between a simulation statistical fingerprint, computed using the simulated driving data, and a real statistical fingerprint, computed using the real input data; receiving a new environment class of the plurality of environment classes; and instructing at least one other hardware processor to: generate a simulated environment, according to the new environment class; generate a plurality of simulated agents, each associated with one of another plurality of moving agent classes and another plurality of movement pattern classes; and generate a simulated driving environment by the simulation generation model according to the simulation environment and the plurality of simulated agents.

According to a fourth aspect of the invention, A method for training an image refining model for generating simulation data for training an autonomous driving agent, comprising: for each of a plurality of image refining models, training the image refining model to generate a refined image in response to input comprising a synthetic image; computing a plurality of model scores by computing for each of the plurality of image refining models a model score indicative of a quality of realism of the respective refined image generated by the respective image refining model in response to input comprising the synthetic image, and indicative of a quality of content preservation of the respective refined image with reference to the synthetic image; and selecting an image refining model of the plurality of image refining models having a preferred model score of the plurality of model scores.

According to a fifth aspect of the invention, a system for training an image refining model for generating simulation data for training an autonomous driving agent comprises at least one hardware processor adapted to: for each of a plurality of image refining models, training the image

refining model to generate a refined image in response to input comprising a synthetic image; computing a plurality of model scores by computing for each of the plurality of image refining models a model score indicative of a quality of realism of the respective refined image generated by the respective image refining model in response to input comprising the synthetic image, and indicative of a quality of content preservation of the respective refined image with reference to the synthetic image; and selecting an image refining model of the plurality of image refining models having a preferred model score of the plurality of model scores.

According to a sixth aspect of the invention, a method for generating simulation data for training an autonomous driving agent comprises: accessing an image refining model trained by: for each of a plurality of image refining models, training the image refining model to generate a refined image in response to input comprising a synthetic image; computing a plurality of model scores by computing for each of the plurality of image refining models a model score indicative of a quality of realism of the respective refined image generated by the respective image refining model in response to input comprising the synthetic image, and indicative of a quality of content preservation of the respective refined image with reference to the synthetic image; and selecting a computer generated model of the plurality of computer generated models having a preferred model score of the plurality of model scores; and instructing at least one hardware processor to generate by the image refining model at least one refined output image in response to at least one input synthetic image, received via an input interface from an environment generation engine.

With reference to the first and second aspects, in a first possible implementation of the first and second aspects of the present invention the real input data comprises environment data and agent data; the environment data comprises a plurality of environment values of a plurality of environment attributes describing the driving environment; and the agent data comprises for each of a plurality of agents operating in the driving environment a plurality of agent values of a plurality of agent attributes, each plurality of agent values describing an appearance of the respective agent and a behavior in the driving environment of the respective agent. Describing the driving environment using a plurality of environment values of a plurality of environment attributes and describing an appearance of an agent and a behavior of the agent in the driving environment using a plurality of agent values of a plurality of agent attributes facilitates generating one or more simulated agents realistically representing one or more agents detected in the real input data, increasing accuracy of an output of a model trained using the one or more simulated agents, thus increasing accuracy of an autonomous driving agent trained using the output of the model trained using the one or more simulated agents. Optionally, at least one of the plurality of environment values is selected from a group of environment values consisting of: a time of day

value, a day of week value, a month of year value, a temperature value, an air clarity metric value, a precipitation value, a precipitation type value, a wind velocity value, a wind direction value, an amount of light value, an artificial light indication value, a crowd density metric value, an incline value, a curve direction value, a curve radius value, and a road horizontal angle value. Optionally, at least one of the plurality of agent values is selected from a group of agent values consisting of: a height value, a width value, a longitudinal velocity value, a lateral velocity value, a longitudinal acceleration value, a lateral acceleration value, a longitudinal distance value, a lateral distance value, a longitudinal direction value, a lateral direction value, a traffic rule violation indication value, an opposite lane indication value, a shoulder indication value, a collision indication value, an amount of passes of other agents, a transition time value, a transition delay time value, an outbound transition angle value, and an inbound transition angle value. Optionally, each of the plurality of environment classes comprises at least some of the plurality of environment values. Optionally, each of the plurality of moving agent classes describes a moving agent selected from a group of possible moving agents consisting of: a car, a truck, a motorized vehicle, a train, a boat, an air-born vehicle, a waterborne vessel, a motorized scooter, a scooter, a bicycle, and a pedestrian; and each of the plurality of moving agent classes comprises a plurality of agent probabilities, each agent probability indicative of a likelihood of a moving agent associated with the respective moving agent class to demonstrate an agent behavior described by an identified plurality of agent values of the plurality of agent values when the driving environment is described by a first environment class of the plurality of environment classes. Using a plurality of agent probabilities facilitates simulating realistic behavior of one or more simulated agents, increasing accuracy of an output of a model trained using the one or more simulated agents, thus increasing accuracy of an autonomous driving agent trained using the output of the model trained using the one or more simulated agents. Optionally, each of the plurality of movement pattern classes describes an actor selected from a group of possible actors consisting of: a pedestrian, a bicycle rider, a vehicle driver; and each of the plurality of movement pattern classes comprises a plurality of actor probabilities, each actor probability indicative of a likelihood of an actor associated with the respective movement pattern class to demonstrate an actor behavior described by an identified movement pattern plurality of agent values of the plurality of agent values when the driving environment is described by a second environment class of the plurality of environment classes. Using a plurality of movement pattern probabilities facilitates simulating realistic behavior of one or more simulated agents, increasing accuracy of an output of a model trained using the one or more simulated agents, thus increasing accuracy of an autonomous driving agent trained using the output of the model trained using the one or more simulated agents.

With reference to the first and second aspects, in a second possible implementation of the first and second aspects of the present invention analyzing the real input data comprises applying a transformation to the plurality of agent values. Applying a transformation to the real input data allows aligning one or more signals captured by one or more sensors, thus increasing accuracy of the plurality of environment classes, the plurality of moving agent classes, and the plurality of movement classes, and therefore increasing accuracy of the simulated driving environment generated according to the plurality of moving agent classes, the plurality of movement classes, and one of the plurality of environment classes. Increasing accuracy of the simulated driving environment increases accuracy of an output of a model trained using the simulated driving environment and thus increases accuracy of an autonomous driving agent trained using the output of the model.

With reference to the first and second aspects, in a third possible implementation of the first and second aspects of the present invention the real statistical fingerprint is computed using the plurality of environment values and the plurality of agent values. Optionally, the simulated driving data comprises simulated environment data and simulated agent data. Optionally, the simulated environment data comprises a plurality of simulated environment values of the plurality of environment attributes describing the simulated driving environment; the simulated agent data comprises for each of the plurality of generated training agents a plurality of simulated agent values of the plurality of agent attributes, each plurality of simulated agent values describing an appearance of the respective generated training agent and a behavior in the simulated driving environment of the respective generated training agent; and the simulation statistical fingerprint is computed using the plurality of simulation environment attribute values and the plurality of simulation agent values. Computing a real statistical fingerprint using the plurality of environment values and the plurality of agent values and computing a simulation statistical fingerprint using the plurality of simulation environment attribute values and the plurality of simulation agent values increases accuracy of a comparison between a simulated driving environment and a real driving environment, thus reduces an amount of time required to train a model for generating simulation data, reducing cost of development of the model. In addition, using such simulation statistical fingerprint and real statistical fingerprint increases accuracy of an output of a model trained using the simulated driving environment.

With reference to the first and second aspects, in a fourth possible implementation of the first and second aspects of the present invention at least some of the plurality of environment classes, the plurality of moving agent classes and the plurality of movement pattern classes are identified using at least one machine learning model to analyze the real input data. Optionally, at

least some of the plurality of environment classes, the plurality of moving agent classes and the plurality of movement pattern classes are identified by applying at least one statistical analysis method to the real input data. Optionally, the real input data further comprises at least one label indicative of one or more environmental conditions when at least some of the real input data is collected from the driving environment, and analyzing the real input data comprises using at least some of the at least one label.

With reference to the first and second aspects, in a fifth possible implementation of the first and second aspects of the present invention generating the simulated driving environment according to the plurality of generated training agents comprises: associating each of the plurality of generated training agents a moving agent class of the plurality of moving agent classes and a movement pattern class of the plurality of movement classes; and generating for each of the plurality of generated training agents simulated behavior data in accordance with: 1) the plurality of agent probabilities of the respective moving agent class according to the one environment class, and 2) the plurality of actor probabilities of the respective movement pattern class according to the one environment class. Optionally, generating the simulated driving environment further comprises computing a probability score for each of the plurality of moving agent classes and each of the plurality of movement pattern classes. Optionally, a moving agent class of the plurality of moving agent classes is associated with a generated training agent according to the respective probability score; and a movement pattern class of the plurality of movement pattern classes is associated with a generated training agent according to the respective probability score. Optionally, generating the simulated driving environment further comprises: identifying in the plurality of moving agent classes at least one less probable moving agent classes according to the respective probability score; and associating at least one of the generated training agents with one of the at least one less probable moving agent classes according to an agent probability higher than an initial agent probability in accordance with the respective probability score. Generating at least one training agent associated with a less probable moving agent class allows ensuring an autonomous driving agent trained with the simulation environment encounters behavior described by the less probable moving agent class more frequently than realistic driving conditions dictate, thus increasing accuracy of an output of the autonomous driving agent. Optionally, generating the simulated driving environment further comprises: identifying in the plurality of movement pattern classes at least one less probable movement pattern classes according to the respective probability score; and associating at least one of the generated training agents with one of the at least one less probable movement pattern classes according to an actor probability higher than an initial actor probability in accordance with the respective probability score. Generating at least one

training agent associated with a less probable movement pattern class allows ensuring an autonomous driving agent trained with the simulation environment encounters behavior described by the less probable movement pattern class more frequently than realistic driving conditions dictate, thus increasing accuracy of an output of the autonomous driving agent.

With reference to the first and second aspects, in a sixth possible implementation of the first and second aspects of the present invention training the model further comprises generating another training environment, according to another environment class of the plurality of environment classes; and in each of a plurality of additional training iterations: generating, by another simulation generation model, another simulated driving environment according to the other training environment and according to a plurality of other generated training agents, each associated with one of the plurality of moving agent classes and one of the plurality of movement pattern classes; collecting other simulated driving data from the other simulated driving environment; and modifying a plurality of other model parameters of the other simulation generation model to minimize another difference between another simulation statistical fingerprint, computed using the other simulated driving data, and the real statistical fingerprint, computed using the real input data. Using the plurality of environment classes, the plurality of moving agent classes and the plurality of movement pattern classes identified in the real input data to generate more than one training environment reduces cost of development of the model.

With reference to the first and second aspects, in a seventh possible implementation of the first and second aspects of the present invention the real input data comprises real data collected by at least one sensor selected from a group of sensors comprising: a camera, an electromagnetic radiation sensor, a radar, a Light Detection and Ranging (LIDAR) sensor, a microphone, a thermometer, an accelerometer, and a video camera.

With reference to the first and second aspects, in an eighth possible implementation of the first and second aspects of the present invention the at least one hardware processor is further adapted to output the simulation generation model to a non-volatile digital storage connected to the at least one hardware processor. Optionally, the at least one hardware processor is further adapted to receiving the real input data via at least one digital communication network interface connected to the at least one hardware processor. Optionally, the at least one hardware processor is further adapted to output the simulation generation model via the at least one digital communication network interface. Outputting the simulation generation model via the at least one digital communication network interface facilitates using the simulation generation model in more than one simulation generation systems, reducing cost of implementation of a simulation generation system using the simulation generation model.

With reference to the fourth and fifth aspects, in a first possible implementation of the fourth and fifth aspects of the present invention the model score is computed by: producing a plurality of refined output images, each generated by the respective image refining model in response to input comprising one of a plurality of test images; computing a plurality of image self-distance scores, each indicative of a quality of content preservation in one of the plurality of refined output images with reference to the respective test image; computing a plurality of realism scores, each indicative of a quality of realism of one of the plurality of refined output images; computing a plurality of image quality scores each indicative of an image quality of one of the plurality of refined output images; computing a task-oriented score by: computing a training score of a first classifier model, trained using the plurality of refined output images, indicative of a success rate of the first classifier model classifying a plurality of real images; and computing a testing score of a second classifier model, trained using the plurality of real images, indicative of a success rate of the second classifier model classifying the plurality of refined output images; and computing the model score using the plurality of image self-distance scores, the plurality of feature scores, the plurality of image quality scores and the task-oriented score. Optionally, computing the model score comprises computing at least one term selected from a group of terms consisting of: an average of the plurality of image self-distance scores, a standard deviation of the plurality of image self-distance scores, a sum of the plurality of image self-distance scores, an average of the plurality of realism scores, a standard deviation of the plurality of realism scores, a sum of the plurality of realism scores, an average of the plurality of image quality scores, a standard deviation of the plurality of image quality scores, and a sum of the plurality of image quality scores. Optionally, computing the plurality of realism scores comprises computing a plurality of feature scores, each indicative of a distance between: 1) a plurality of refined features in one of the plurality of refined output images, and 2) a plurality of features extracted from a target real image equivalent to the respective test image. Optionally, at least one of the plurality of feature scores is a Frechet Inception Distance (FID). Optionally, computing the plurality of realism scores comprises computing a plurality of reconstructions scores, each indicative of a difference between a refined image of the plurality of refined output images and a reconstructed image, reconstructed from a corrupted refined image by a reconstruction model trained to reconstruct real images. Computing the model score using the plurality of image self-distance scores, the plurality of feature scores, the plurality of image quality scores and the task-oriented score allows the model score to reflect one or more metrics of model quality, and in addition allows adjusting how the one or more metrics are reflected in the model score, increasing accuracy of the model score with respect to an identified goal, thus increasing accuracy of an output of the image refining model.

With reference to the fourth and fifth aspects, or the first implementation of the fourth and fifth aspects, in a second possible implementation of the fourth and fifth aspects of the present invention each of the plurality of image-quality scores is normalized in a range from 0 to 1. Optionally, each of the plurality of realism scores is normalized in a range from 0 to 1. Optionally, each of the plurality of image self-distance scores is normalized in a range from 0 to 1. Normalizing one or more of the scores used to compute the model score facilitates applying one or weights when computing the model score, increasing accuracy of the model score.

With reference to the fourth and fifth aspects, or the first implementation of the fourth and fifth aspects, in a third possible implementation of the fourth and fifth aspects of the present invention the first classifier is a first semantic segmentation classifier; and the training score is a first F1 score indicative of a first percentage of pixels of each of the plurality of refined output images classified correctly by the first classifier. Optionally, the second classifier is a second semantic segmentation classifier; and the testing score is a second F1 score indicative of a second percentage of pixels of each of the plurality of real images classified correctly by the second classifier.

Other systems, methods, features, and advantages of the present disclosure will be or become apparent to one with skill in the art upon examination of the following drawings and detailed description. It is intended that all such additional systems, methods, features, and advantages be included within this description, be within the scope of the present disclosure, and be protected by the accompanying claims.

Unless otherwise defined, all technical and/or scientific terms used herein have the same meaning as commonly understood by one of ordinary skill in the art to which the invention pertains. Although methods and materials similar or equivalent to those described herein can be used in the practice or testing of embodiments of the invention, exemplary methods and/or materials are described below. In case of conflict, the patent specification, including definitions, will control. In addition, the materials, methods, and examples are illustrative only and are not intended to be necessarily limiting.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

Some embodiments of the invention are herein described, by way of example only, with reference to the accompanying drawings. With specific reference now to the drawings in detail, it is stressed that the particulars shown are by way of example and for purposes of illustrative discussion of embodiments of the invention. In this regard, the description taken with the drawings makes apparent to those skilled in the art how embodiments of the invention may be practiced.

In the drawings:

FIG. 1 is a schematic block diagram of an exemplary system for training a model for generating simulation data, according to some embodiments of the present invention;

FIG. 2 is a flowchart schematically representing an optional flow of operations for training a model for generating simulation data, according to some embodiments of the present invention;

FIG. 3 is a flowchart schematically representing an optional flow of operations for generating agents, according to some embodiments of the present invention;

FIG. 4 is flowchart schematically representing another optional flow of operations for training a model for generating simulation data, according to some embodiments of the present invention;

FIG. 5 is a flowchart schematically representing an optional flow of operations for generating simulation data, according to some embodiments of the present invention;

FIG. 6 is a schematic block diagram of an exemplary system for training an image refining model, according to some embodiments of the present invention;

FIG. 7 is a flowchart schematically representing an optional flow of operations for training an image refining model, according to some embodiments of the present invention;

FIG. 8 is a flowchart schematically representing an optional flow of operations for computing a model score of an image refining model, according to some embodiments of the present invention; and

FIG. 9 is a flowchart schematically representing another optional flow of operations for generating simulation data, according to some embodiments of the present invention.

DETAILED DESCRIPTION

The present invention, in some embodiments thereof, relates to creating a simulated model of a geographical area, and, more specifically, but not exclusively, to creating a simulated model of a geographical area, optionally including vehicle traffic to generate simulation sensory data for training an autonomous driving system.

For brevity, henceforth the term "agent" is used to mean "simulated agent" and the terms are used interchangeably. In addition, for brevity the term "autonomous driver" is used to mean "autonomous driving system" and the terms are used interchangeably.

In the field of generating simulation environments for training and testing it is common to use random values to simulate real world situations. Thus, some systems for generating simulation data for training an autonomous driving system generate agents that simulate random movement patterns of a moving object. One possible way to generate a random movement pattern for a

simulated agent is by generating random movement characteristics of the simulated agent. Some examples of an agent's movement characteristics are: a forward velocity, a forward acceleration, a lateral velocity, and a lateral acceleration. Some movement characteristics of the agent pertain to distances, for example a distance from a lane boundary, and a distance from another moving object moving in front of the moving object represented by the agent. Some movement characteristics of the agent pertain to passing other moving objects, for example a distance from another object in front of the moving object, a distance from another object on a side of the moving object, an angle of a transition out of a lane, and an angle of a transition in to an original lane. Some movement characteristics of the agent pertain to lane transitions, for example an angle of a lane transition, a direction of a lane transition, and a distance from another object when transitioning between two or more lanes. Some movement characteristics of the agent pertain to pedestrian movement, for example an amount of time between a pedestrian stopping and the pedestrian transitioning from a sidewalk to a street. Some movement characteristics of the agent pertain to collisions, for example an angle of a collision, a location of impact on a vehicle, and location of impact on a pedestrian. Some movement characteristics of the agent pertain to traffic rules, for example a vehicle or pedestrian moving through a red light, a vehicle moving in a lane in a direction opposite to the lane's direction, and a vehicle moving on a shoulder of a road. Some movement characteristics are indicative of where a pedestrian crosses a road, for example at a designated crosswalk, at a street comer, and in the middle of a straight stretch of road.

Another possible way to generate a random movement pattern for a simulated agent is by randomly generating the simulated agent's behavior at every point of the simulation. However, there is a need to produce agents that simulate realistic driving patterns and not just random driving patterns.

Instead of randomly generating driving characteristics of a simulated agent, or randomly generating a simulated agent' s behavior at every point of a simulation, the present invention, in some embodiments thereof, proposes learning realistic moving object characteristics and realistic movement patterns from real traffic data collected from real traffic environments. In such embodiments the present invention further proposes generating one or more simulated agents according to the learned moving object characteristics and realistic movement patterns, such that each simulated agent embodies a realistic type of moving object (that is, vehicle or pedestrian) and a realistic type of movement pattern in a realistic traffic environment. Optionally, a simulation generation model generates a simulated driving environment using the one or more simulated agents. The simulated driving environment may be used to train an autonomous driving system. Additionally, or alternatively, the simulated driving environment may be used to test an

autonomous driving system. Generating the simulated agents according to the learned moving object characteristics and realistic movement patterns increases accuracy of the simulation driving environment generated using the simulated agents and adds valuable simulated cases, and thus increases accuracy of an output of a system trained using the simulation driving environment, for example an autonomous driving system.

In computing, a statistical fingerprint of identified data is a statistical value computed using the identified data that uniquely identifies the identified data for practical comparison purposes. The present invention additionally proposes in some embodiments thereof classifying the realism of the simulated driving environment using a difference between a simulation statistical fingerprint computed using simulated driving data collected from a simulated driving, and a real statistical fingerprint computed using the real traffic data. An understanding of a world may be based on statistics collected about the world. In simulating agents representing moving objects, some statistics are of the nature of "how probable is it that a vehicle of type X driven by a driver of type Y will do Z in driving environment W", for example "how probable is it that a vehicle of type X driven by a teenager will cut a lane at angle a at night". Thus, according to some embodiments of the present invention, the real traffic data is analyzed to identify a plurality of environment classes, a plurality of moving agent classes and a plurality of movement pattern classes. Each of the environment classes describes a driving environment, and optionally comprises one or more environment values of a plurality of environment attributes describing the driving environment. Some examples of an environment attribute are a time of day, a time of week, and a month of year. Some environment attributes pertain to weather conditions, for example a temperature, an amount of fog, an air clarity metric, for example an amount of fog or an amount of smog, an amount of precipitation, a wind velocity and a wind direction. Some environment attributes pertain to environment conditions, for example amount of daylight, amount of clouds, type of clouds, artificial light, and an amount of vehicles per an identified amount of time. Some environment attributes pertain to geography, for example an incline, a direction of a curve, a radius of a curve a horizontal angle of a road, a body of water, and a geographical environment type such as a building site, a forest, an urban area, a sub-urban area, a military area, and an open field. Each of the plurality of moving agent classes describes a moving agent simulating a moving object, and optionally comprises one or more agent probabilities, each indicative of a likelihood of a moving agent associated with the respective moving agent class to demonstrate an identified agent behavior in an identified driving environment. The identified agent behavior is optionally described by an identified plurality of agent values of the plurality of agent attributes. The driving environment is optionally described by an identified environment class of the plurality of

environment classes. Optionally, each of the plurality of moving agent classes comprises one or more agent values each of one of a plurality of agent attributes. Some examples of an agent attribute are height, width, and weight. Optionally, some of plurality of agent attributes are agent movement characteristics, for example as described above. Each of the plurality of movement pattern classes describes an actor. Some examples of an actor are a pedestrian, a bicycle driver, and a vehicle driver. Optionally each of the plurality of movement pattern classes comprises a plurality of actor probabilities, each indicative of a likelihood of an actor associated with the respective movement pattern class to demonstrate an identified actor behavior in another identified driving environment. The identified actor behavior is optionally described by an identified movement pattern plurality of agent values of the plurality of agent attributes. Optionally, the other identified driving environment is described by another identified environment class of the plurality of environment classes.

Optionally, the plurality of environment classes, the plurality of moving agent classes and the plurality of movement classes are used to generate the simulated driving environment. Optionally, the driving environment is generated according to one of the plurality of environment classes. Optionally, the simulated driving environment is generated according to one or more simulated agents, each associated with one of the plurality of moving agent classes and one of the plurality of movement pattern classes. Optionally the simulated driving environment is generated according to the respective plurality of agent probabilities and the respective plurality of actor probabilities of each of the one or more simulated agents.

In some embodiments the present invention proposes training the simulation generation model using the difference between the simulation statistical fingerprint and the real statistical fingerprint. Optionally the real statistical fingerprint is computed using the plurality of environment values and the plurality of agent values. Optionally, the simulated driving data comprises one or more simulated environment values of the plurality of environment attributes describing the simulated driving environment. Optionally, the simulated driving data comprises for each of the plurality of generated agents a plurality of simulated agent values of the plurality of agent attributes. Optionally, the simulation statistical fingerprint is computed using the plurality of simulated environment values and the plurality of simulated agent values. Optionally, in at least one of a plurality of training iterations at least one model parameter of the simulation generation model is modified to minimize a difference between the simulation statistical fingerprint and the real statistical fingerprint. Using the difference between the simulation statistical fingerprint and the real statistical fingerprint to train the simulation generation model increases the accuracy of an output of the simulation generation model, thus increasing accuracy of a system trained using the

output of the simulation generation model, for example an autonomous driving system. Computing a statistical fingerprint according to a plurality of environment values and a plurality of agent values increases accuracy of the statistical fingerprint as an encapsulation of an environment's characteristics, thus increasing accuracy of an output of a simulation generation model trained using one or more statistical fingerprints each computed according to a plurality of environment values and a plurality of agent values.

In addition, the present invention proposes training more than one simulation generation model, each trained according to one of the plurality of environment classes, and training an autonomous driving system using output generated by the more than one simulation generation model. Training the autonomous driving system using more than one simulated driving environment, each generated by one of the more than one simulation generation models increases the accuracy of an output of the autonomous driving system, allowing the autonomous driving system to drive correctly in more than one environment.

To generate the simulated driving environment according to the one or more simulated agents, the present invention further proposes, in some embodiments thereof, computing for each of the plurality of moving agent classes and each of the plurality of movement pattern classes a probability score, each indicative of a probability that the simulated driving environment, characterized by the environment class, comprises a moving agent associated with the respective moving agent class and additionally or alternatively with the respective movement pattern class. For example, during school hours (such as some hours of the morning), a movement pattern characteristic of an adolescent may be less likely than during the afternoon and early evening. In another example, during rush hours there may be a higher probability for a car to exist in a driving environment than in the middle of the night. In some embodiments of the present invention, when simulating an amount of agents, assignment of an agent to a moving agent class is according to a probability the moving agent class appears in the simulated driving environment according to the environment class. Optionally, assignment of an agent to a movement pattern class is according to a probability the movement pattern class appears in the simulated driving environment according to the environment class.

Training an autonomous driver may require special attention to edge cases - reckless drivers cutting lanes, inattentive pedestrians stepping off the sidewalk and crossing in the middle of the road instead of at a cross walk, particularly slow drivers blocking traffic etc. To train an autonomous driver it may be desirable to assign an agent to a less probable moving agent class or movement pattern class, in order to expose the autonomous driver to such relatively rare conditions. Thus, according to some embodiments of the present invention an agent is assigned a

movement pattern class at a higher probability than the movement pattern class's probability. Optionally the agent is assigned a moving agent class at a higher probability than the moving agent class's probability. For example, when the probability of a very slow pedestrian (for example an older pedestrian) in rainy conditions is low, for example 1%, the probability of generating an agent associated with a very slow pedestrian may be increased to higher value, for example 25%, to ensure the autonomous driver encounters such behavior more frequently than realistic conditions dictate, thus increasing accuracy of an output of the autonomous driver in rare driving conditions.

As described above, an image refining model (a refiner) may be used to refine synthetic images generated by a simulation generation model, for example a simulated driving environment, to increase realism of the synthetic images.

Some methods for training an image refining model comprise using various metrics to evaluate an output of the refiner and attempt to improve a value of the refiner evaluation metric. Output of one trained refiner may have significantly different quality than output of another trained refiner. It may be that training an identified refiner does not converge to a desired equilibrium. Investing in improving such a refiner could consume processing resources and time without yielding any benefit. Identifying such a refiner a-priori is currently not possible.

Instead of improving one image refining model, the present invention, in some embodiments thereof, additionally proposes training a plurality of image refining models, computing for each a model score indicative of how good the output images of the respective model are, and using an image refining model with a preferred model score to generate realistic simulation data for training an autonomous driver.

There is a need to train an image refining model to produce an output image that is similar to an input synthetic image in its content and similar to a target real environment in its appearance, or style. Existing methods for training an image refining model cannot achieve best similarity in contents and best similarity to the target real environment simultaneously. The present invention, in some embodiments thereof, further proposes evaluating output of an image refining model using a model score reflecting one or more qualities of the output images: how realistic the images are and how well they preserve the contents of the input synthetic images. In addition, the score optionally reflects how good the images are according to one or more image quality metrics (blur, noise, distortion, artifacts etc.). In addition, the model score optionally reflects a task oriented score of the image refining model, indicative of how well images produced by the image refiner model can pass as real images with a classifier model trained to classify digital images. Optionally, computing the task oriented score comprises training a classifier using a set of refined output images generated by the image refining model, and optionally computing a score according to the

classifier's success at classifying as real a set of real target images (the higher the better). Successfully classifying a set of real target images by a classifier trained using a set of refined output images generated by the image refining model is indicative of an accuracy of an output of the image refining model. Optionally, computing the task oriented score comprises training another classifier using the set of real target images and optionally computing another score according to the other classifier's success at classifying the set of refined output images as real (the higher the better). Successfully classifying a set of refined output images generated by the image refining model by a classifier trained using a set of real target images is indicative of an accuracy of an output of the image refining model. Computing a model score using the task oriented score increases accuracy of the model score, and thus increases accuracy of realistic simulation data generated using an image refining model selected according to the preferred model score.

As there is a tradeoff between one or more qualities of the output images, the model score is optionally computed using a combination of the one or more qualities of the output images, reflecting an identified tradeoff between the one or more qualities, thus increasing the accuracy of the realistic simulation data generated using the image refining model selected according to the preferred model score.

Before explaining at least one embodiment of the invention in detail, it is to be understood that the invention is not necessarily limited in its application to the details of construction and the arrangement of the components and/or methods set forth in the following description and/or illustrated in the drawings and/or the Examples. The invention is capable of other embodiments or of being practiced or carried out in various ways.

The present invention may be a system, a method, and/or a computer program product. The computer program product may include a computer readable storage medium (or media) having computer readable program instructions thereon for causing a processor to carry out aspects of the present invention.

The computer readable storage medium can be a tangible device that can retain and store instructions for use by an instruction execution device. The computer readable storage medium may be, for example, but is not limited to, an electronic storage device, a magnetic storage device, an optical storage device, an electromagnetic storage device, a semiconductor storage device, or any suitable combination of the foregoing.

Computer readable program instructions described herein can be downloaded to respective computing/processing devices from a computer readable storage medium or to an external

computer or external storage device via a network, for example, the Internet, a local area network, a wide area network and/or a wireless network.

The computer readable program instructions may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider). In some embodiments, electronic circuitry including, for example, programmable logic circuitry, field-programmable gate arrays (FPGA), or programmable logic arrays (PLA) may execute the computer readable program instructions by utilizing state information of the computer readable program instructions to personalize the electronic circuitry, in order to perform aspects of the present invention.

Aspects of the present invention are described herein with reference to flowchart illustrations and/or block diagrams of methods, apparatus (systems), and computer program products according to embodiments of the invention. It will be understood that each block of the flowchart illustrations and/or block diagrams, and combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer readable program instructions.

The flowchart and block diagrams in the Figures illustrate the architecture, functionality, and operation of possible implementations of systems, methods, and computer program products according to various embodiments of the present invention. In this regard, each block in the flowchart or block diagrams may represent a module, segment, or portion of instructions, which comprises one or more executable instructions for implementing the specified logical function(s). In some alternative implementations, the functions noted in the block may occur out of the order noted in the figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that each block of the block diagrams and/or flowchart illustration, and combinations of blocks in the block diagrams and/or flowchart illustration, can be implemented by special purpose hardware-based systems that perform the specified functions or acts or carry out combinations of special purpose hardware and computer instructions.

For brevity, the term processing unit refers to one or more processors of a system executing code. Some examples of a processor of a system are a virtual machine, a server, a desktop computer

and a hardware controller. Henceforth, the term hardware processor is used interchangeably with the term processing unit, meaning one or more processors of a system executing a code.

In addition, for brevity the term "network interface" is used to mean at least one digital communication network interface.

Henceforth the term simulation generation model is used to mean a model for generating simulation data, and the terms are used interchangeably.

Reference is now made to FIG. 1, showing a schematic block diagram of an exemplary system 100 for training a model for generating simulation data, according to some embodiments of the present invention. In such embodiments, processing unit 101 is connected to at least one non-volatile digital storage 111. Some examples of a non-volatile digital storage are a hard disk drive, a solid state memory component, and a network connected storage. Optionally, processing unit 101 outputs a trained simulation generation model to at least one non-volatile digital storage 111. Optionally, processing unit 101 retrieves real input data, collected from a driving environment, from at least one non-volatile digital storage 111. Optionally, processing unit 101 is connected to network interface 110. Optionally, processing unit 101 receives the real input data from a database or a network connected source via network interface 110. Optionally, processing unit 101 outputs the trained simulation generation model via network interface 110, for example by sending the trained simulation generation model to another processing unit. Optionally, network interface 110 is connected to a local area network, for example and Ethernet network or a wireless network. Optionally, network interface 110 is connected to a wide area network, for example the Internet.

To train a model for generating simulation data, system 100 implements in some embodiments of the present invention the following optional method.

Reference is now made also to FIG. 2, showing a flowchart schematically representing an optional flow of operations 200 for training a model for generating simulation data, according to some embodiments of the present invention. In such embodiments, in 201 processing unit 101 accesses real input data, collected from a driving environment. Optionally, processing unit 101 retrieves the real input data from at least one non-volatile digital storage 111. Optionally, processing unit 101 receives the real input data via network interface 110. Optionally, the real input data comprises real data collected by one or more sensors. Some examples of a sensor are a camera, an electromagnetic radiation sensor, a radar, a Light Detection and Ranging (LIDAR) sensor, a microphone, a thermometer, an accelerometer and a video camera. Optionally, the real data is collected by one or more sensors mounted on one or more vehicles, optionally while the vehicle traverses one or more driving environments. Optionally, the real input data comprises

environment data, describing the one or more driving environments. Optionally, the environment data comprises a plurality of environment values of a plurality of environment attributes describing the driving environment. Some examples of an environment value are a time of day value, a time of week value, and a month of year value. Some other examples of an environment value are a temperature value, an amount of fog, an air clarity metric value, for example an amount of fog or an amount of smog, a precipitation value, for example an amount of precipitation or a precipitation rate value, a precipitation type value, a wind velocity value, a wind direction value, an amount of light value, an artificial light indication value, a characteristic of artificial light such as a color value or an intensity value, a crowd density metric value, an incline value, a curve direction value, a curve radius value, and a road horizontal angle value. Optionally, the real input data comprises one or more labels, indicative of one or more environmental conditions when capturing at least some of the data from the driving environment, for example a geographical location of the driving environment or a time of day of the driving environment. Optionally, the real input data comprises agent data, describing one or more agents operating in at least one of the one or more driving environments. Optionally, the agent data describes, for each of the one or more agents, one or more structural features of a vehicle or a pedestrian, and additionally or alternatively one or more behavior characteristics of a vehicle of a pedestrian. Some examples of a vehicle are a car, a truck, and a bicycle. The vehicle may be a motorized vehicle other than a car and a truck. The vehicle may be an air-bome vehicle, for example an airplane or a drone. The vehicle may be a waterborne vessel, for example a boat, a raft, and a personal watercraft such as a water scooter. A pedestrian may be a human pedestrian. A pedestrian may be an animal, such as a dog or a horse. Optionally, the agent data comprises for each of the one or more agents a plurality of agent attributes describing an appearance of the respective agent and a behavior in the driving environment of the respective agent. Some examples of an agent value are a height value and a width value. Some other examples of an agent value are: a longitudinal velocity value, a lateral velocity value, a longitudinal acceleration value, a lateral acceleration value, a longitudinal distance value, a lateral distance value, a longitudinal direction value, a lateral direction value, a traffic rule violation indication value, an opposite lane indication value, a shoulder indication value, a collision indication value, an amount of passes of other agents, a transition time value, a transition delay time value, an outbound transition angle value, and an inbound transition angle value.

In 205, processing unit 101 optionally applies to the real input data a transformation according to a lane, to produce transformed input data. Optionally, the transformation according to a lane provides a perspective view relative to the lane. Optionally, the transformation according

to a lane provides a birds-eye view of the lane. Optionally, the transformed input data comprises the one or more labels of the real input data.

In 210, processing unit 101 optionally analyzes the transformed input data to identify plurality of environment classes 261, each describing a driving environment, plurality of moving agent classes 262, each describing a moving agent simulating a moving object, and plurality of movement pattern classes 263, each describing an actor. Optionally, each of plurality of environment classes 261 comprises at least some of the plurality of environment values, optionally transformed in 205 by the transformation. Optionally, an environment class of the plurality of environment classes describes environmental noise of an identified geographical location, for example San Francisco, Denver or New Delhi. Some examples of environmental noise are an air clarity metric value, an amount of vegetation, a type of vegetation, an amount of buildings, a density of buildings, a type of buildings, and a light characteristic value. Some examples of a moving agent are a car, a truck, a motorized vehicle, a train, a boat, an air-bom vehicle, a waterborne vessel, a motorized scooter, a scooter, a bicycle and a pedestrian. Some examples of an actor are a pedestrian, a bicycle rider and a vehicle driver. Optionally, each of the plurality of moving agent classes comprises a plurality of agent probabilities. Optionally, each agent probability is indicative of a likelihood of a moving agent associated with the respective moving agent class to demonstrate an agent behavior described by an identified plurality of agent values of the plurality of agent values when a driving environment is described by a first environment class of the plurality of environment classes. For example, an agent class may have a likelihood of a velocity value exceeding 100 miles per hour lower when the first environment class describes a driving environment having fog, than when the first environment class describes a driving environment having no fog. Optionally, each of the plurality of movement pattern classes comprises a plurality of actor probabilities. Optionally, each actor probability is indicative of a likelihood of an actor associated with the respective movement pattern class to demonstrate an actor behavior described by an identified movement pattern plurality of agent values of the plurality of agent values when a driving environment is described by a second environment class of the plurality of environment classes. For example, a movement pattern class may have a likelihood of not stopping before stepping from a curb to the road lower when an actor associated with the second agent class has a height value exceeding 165 centimeters (possibly an adult) than when the actor has a height value less than 130 centimeters (possibly a child). Optionally, one or more of plurality of environment classes 261 and additionally or alternatively one or more of plurality of moving agent classes 262 and further additionally and alternatively one or more of

plurality of movement pattern classes 263 are identified according to at least some of the real input data collected by more than one sensor, optionally using one or more sensor fusion methods.

Optionally, in 210 processing unit 101 analyzes the transformed input data using at least one machine learning model, and optionally identifies at least some of plurality of environment classes 261, plurality of moving agent classes 262 and plurality of movement pattern classes 263 using the at least one machine learning model. Optionally, in 210 processing unit 101 applies at least one statistical analysis method to the transformed input data to identify the at least some of plurality of environment classes 261, plurality of moving agent classes 262 and plurality of movement pattern classes 263 using the at least one machine learning model.

Optionally, in 210 processing unit 101 further uses the one or more labels when analyzing the transformed input data. For example, when some of the real input data is labeled as an urban area, analyzing the transformed data optionally increases an emphasis on identifying one or more road features in the some of the real input data. In another example, when some other of the real input is labeled as an open field, analyzing the transformed data optionally decreases an emphasis on identifying one or more other road features in the some other of the real input data.

Optionally, in 210 processing unit 101 analyzes the real input data instead of the transformed input data to identify plurality of environment classes 261, plurality of moving agent classes 262 and plurality of movement pattern classes 263.

In 221, processing unit 101 optionally computes a real statistical fingerprint using the real input data, such that the real statistical fingerprint is indicative of one or more characteristics of the one or more driving environments for practical comparison purposes. Optionally, the real statistical fingerprint is computed using the plurality of environment values and the plurality of agent values.

In 222, processing unit 101 optionally generates a training environment according to one environment class 264 of plurality of environment classes 261. The model may be created according to one of the identified environment classes. Optionally, environment class 264 is an intersection of two or more other environment classes of plurality of environment classes 261. For example, plurality of environment classes 261 may comprise an environment class of "daytime", an environment class of "San Francisco" and an environment class of "San Francisco at daytime". In this example. environment class 264 may be "San Francisco at daytime".

In 225, processing unit 101 optionally generates a plurality of generated training agents, each associated with one of plurality of moving agent classes 262 and one of plurality of movement pattern classes 263. Optionally, in 227 processing unit 101 uses a simulation generation model to

generate a simulated driving environment according to the training environment generated in 222 and the plurality of generated training agents generated in 225.

Reference is now made also to FIG. 3, showing a flowchart schematically representing an optional flow of operations 300 for generating a plurality of agents, according to some embodiments of the present invention. Processing unit 101 optionally selects an amount of the plurality of generated training agents according to an environment value of the training environment indicative of an amount of vehicles, and additionally or alternatively an amount of pedestrians, in an identified amount of time in the training environment.

In 301, processing unit 101 optionally associates each of the plurality of generated training agents with a moving agent class of the plurality of moving agent classes and a movement pattern class of the plurality of movement classes. In 320, processing unit 101 optionally generates for each of the plurality of generated training agents simulated behavior data. Optionally, the simulated behavior data is in accordance with the plurality of agent probabilities of the respective moving agent class according to environment class 261. Optionally, the simulated behavior data is in accordance with the plurality of actor probabilities of the respective movement pattern class according to environment class 261.

Optionally, in 310 processing unit 101 computes for each of the plurality of moving agent classes and each of the plurality of movement pattern classes a probability score. Optionally, in 301 processing unit 101 associates a generated training agent of the plurality of generated training agents with a moving agent class of the plurality of moving agent classes according to the respective probability score of the moving agent class, optionally according to an initial agent probability in accordance with the respective probability score. Optionally, in 301 processing unit 101 associates a generated training agent of the plurality of generated training agents with a movement pattern class of the plurality of movement pattern classes according to the respective probability score of the movement pattern class, optionally according to an initial actor probability in accordance with the respective probability score. Optionally, in 311, processing unit 101 identifies one or more less probable moving agent classes of the plurality of moving agent classes according to the respective probability scores. Optionally, in 312 processing unit 101 associates one or more of the plurality of generated training agents with one of the one or more less probable moving agent classes according to an agent probability higher than the initial agent probability. Optionally, in 316, processing unit 101 identifies one or more less probable movement pattern classes of the plurality of movement pattern classes according to the respective probability scores. Optionally, in 317 processing unit 101 associates one or more of the plurality of generated training

agents with one of the one or more less probable movement pattern classes according to an actor probability higher than the initial actor probability.

Reference is now made again to FIG. 2. Processing unit 101 optionally collects in 229 simulated driving data from the simulated driving environment. In 231 processing unit 101 optionally computes a simulation statistical fingerprint using the simulated driving data. Optionally, the simulated driving data comprises simulated environment data describing the simulated driving environment. Optionally, the simulated environment data comprises a plurality of simulation environment values of the plurality of environment attributes describing the simulated driving environment. Optionally, the simulated driving data comprises simulation agent data, describing the plurality of generated training agents. Optionally, the generated training agent data comprises for each of the one or more generated training agents a plurality of agent attributes describing an appearance of the respective generated training agent and a behavior in the simulated driving environment of the respective generated training agent. Optionally, processing unit 101 computes the simulation statistical fingerprint in 231 using the plurality of simulation environment attribute values and the plurality of simulation agent values.

In 235, processing unit 101 optionally modifies one or more model parameters of the simulation generation model to minimize a difference between the real statistical fingerprint computed in 221 and the simulation statistical fingerprint computed in 231.

Optionally, processing unit 101 repeats 227, 229, 231 and 235 in one or more of a plurality of training iterations.

According to some embodiments of the present invention, system 100 may train more than one simulation model, for more than one environment.

Reference is now made also to FIG. 4, showing a flowchart schematically representing another optional flow of operations 400 for training another model for generating simulation data, according to some embodiments of the present invention. In such embodiment, processing unit 101 generates in 422 another training environment according to other environment class 401 of plurality of environment classes 261. Optionally, other environment class 401 is an intersection of other two or more other environment classes of plurality of environment classes 261. In 425, processing unit 101 optionally generates another plurality of simulated training agents, each associated with one of the plurality of moving agent classes and one of the plurality of movement pattern classes, and in 427 processing unit 101 optionally generates by another simulation generation model another simulated driving environment according to the plurality of other generated training agents and the other training environment. Processing unit 101 optionally collects in 429 other simulated driving data from the other simulated driving environment. In 431

processing unit 101 optionally computes another simulation statistical fingerprint using the other simulated driving data. In 435, processing unit 101 optionally modifies one or more other model parameters of the other simulation generation model to minimize another difference between the real statistical fingerprint computed in 221 and the other simulation statistical fingerprint computed in 431.

Optionally, processing unit 101 repeats 427, 429, 431 and 435 in one or more of a plurality of additional training iterations.

According to some embodiments of the present invention the simulation generation model trained by system 100 using method 200 is used to generate simulation data, for example for training an autonomous driving agent. In such embodiments, a system comprising a first hardware processor implements the following optional method.

Reference is now made also to FIG. 5, showing a flowchart schematically representing an optional flow of operations 500 for generating simulation data, according to some embodiments of the present invention. In such embodiments, in 501 the first hardware processor accesses a simulation generation model generated by system 100 using method 200. Optionally, in 510 the first hardware processor receives a new environment class of the plurality of environment classes and optionally instructs a second hardware processor to generate the simulated driving environment. Optionally, the first hardware processor is the second hardware processor. Optionally, in 521 the second hardware processor generates a simulated environment, according to the new environment class and in 522 the second hardware processor optionally generates a plurality of simulated agents, each associated with one of another plurality of moving agent classes and one of another plurality of movement pattern classes. Optionally, the other plurality of moving agent classes is the plurality of moving agent classes. Optionally, the other plurality of movement pattern classes is the plurality of movement agent classes. In 523, the second hardware processor optionally generates a simulated driving environment by the simulation generation model according to the simulation environment and the plurality of simulated agents.

To enhance realism of generated simulation data, for example a simulated driving environment, optionally for the purpose of training an autonomous driving agent, in some embodiments of the present invention an image refining model is trained.

Reference is now made also to FIG. 6, showing a schematic block diagram of an exemplary system 600 for training an image refining model, according to some embodiments of the present invention. In such embodiments, processor 601 is connected to at least one non-volatile digital storage 611. Optionally, processing unit 601 outputs a trained image refining model to at least one non-volatile digital storage 611. Optionally, processing unit 101 retrieves from at least one non-

volatile digital storage 611 input data. Optionally the input data comprises a plurality of test images. Optionally the plurality of test images comprises one or more synthetic images. Optionally, processing unit 601 is connected to network interface 610. Optionally, processing unit 601 receives the input from a database or a network connected source via network interface 610. Optionally, processing unit 601 outputs the trained image refining model via network interface 610, for example by sending the trained image refining model to another processing unit. Optionally, network interface 610 is connected to a local area network, for example and Ethernet network or a wireless network. Optionally, network interface 610 is connected to a wide area network, for example the Internet.

To train an image refining model, in some embodiments of the present invention system 600 implements the following optional method.

Reference is now made also to FIG. 7, showing a flowchart schematically representing an optional flow of operations 700 for training an image refining model, according to some embodiments of the present invention. In such embodiments, processing unit 601 trains in 701 each of a plurality of image refining models. Optionally, each of the plurality of image refining models is trained to generate a refined image in response to input comprising a synthetic image. Optionally, one or more of the plurality of image refining models use a Generative Adversarial Network (GAN). In 710, processing unit 601 optionally computes a plurality of model scores by computing for each of the plurality of image refining models a model score. Optionally, each model score of the plurality of model scores is indicative of a quality of realism of the respective refined image generated by the respective image refining model in response to input comprising the synthetic image. Optionally, each model score is additionally or alternatively indicative of a quality of content preservation of the respective refined image with reference to the synthetic image.

Reference is now made also to FIG. 8, showing a flowchart schematically representing an optional flow of operations 800 for computing a model score of an image refining model, according to some embodiments of the present invention. In such embodiments, processing unit 601 produces in 801 a plurality of refined output images, each generated of the respective image refining model in response to input comprising one of a plurality of test images. In 820, processing unit 601 optionally computes the model score, using one or more of: a plurality of image self-distance scores, a plurality of realism scores, a plurality of image quality scores, and a task-oriented score.

An image self-distance score of a refiner measures how well a refined image generated by the refiner in response to an input image preserves content of the input image. In 811, processing unit

601 optionally computes a plurality of image self-distance scores, each indicative of a quality of content preservation in one of the plurality of refined output images with reference to the respective test image of the plurality of test images. Optionally each of the plurality of image self-distance scores is normalized in a range from 0 to 1.

A realism score of a refiner measures how realistic a refined image generated by the refiner in response to an input image appears. In 812, processing unit optionally computes a plurality of realism scores, each indicative of a quality of realism of one of the plurality of refined output images. Optionally each of the plurality of realism scores is normalized in a range from 0 to 1. Optionally, computing the plurality of realism scores comprises computing a plurality of feature scores, each indicative of a distance between a plurality of refined features and a plurality of features extracted from a target real image, where the plurality of refined features are in one of the plurality of refined output images generated in response to a test image of the plurality of test images and the target real image is equivalent to the test image. Optionally, one or more of the plurality of feature scores is a Frechet Inception Distance (FID). A corrupted image is an image that had noise introduced in to it - random noise that corrupts some of the pixels of the image. Some corruption patterns include blurring, and creating artefacts. Optionally, computing the plurality of realism scores comprises computing a plurality of reconstructions scores. Optionally, each of the reconstruction scores is indicative of a difference between a refined image of the plurality of refined output images and a reconstructed image. Optionally, the reconstructed image is reconstructed from a corrupted refined image by a reconstruction model trained to reconstruct real images.

Optionally, to compute a reconstruction score processing unit 601 accesses an image reconstruction model. Optionally, the image reconstruction is trained using real images (photos) to learn a distribution of characteristics of a real environment. The distribution of characteristics is reflected in the reconstruction model's weight values. Optionally, the image reconstruction model is additionally or alternatively trained using a set of corrupted real images (generated by corrupting each of a set of real images) and in a plurality of reconstruction training iterations the image reconstruction model is trained to reconstruct each of the real images. A distance function between an original real image and the respective reconstructed image is optionally used to measure success of reconstruction. Optionally, the image reconstruction model is a de-noising auto-encoder model.

Optionally, for each of the plurality of test images, processing unit 601 corrupts the respective refined output image, and uses the image reconstruction model to output a respective reconstructed image in response to input comprising the respective corrupted refined image.

Processing unit 601 optionally computes a reconstruction score indicative of an amount of errors in the reconstructed image compared to the refined image. A small amount of errors is indicative of a refined output image fitting a distribution that the refiner learned from real data, thus indicating a greater degree of realism of the respective refined output image.

An image quality score of an image assesses one or more quality metrics of the image including, but not limited to, noise, blurriness, distortion artifacts. Optionally, processing unit 601 computes in 813 a plurality of image quality scores, each indicative of an image quality of one of the plurality of refined output images. Optionally each of the plurality of image-quality scores is normalized in a range from 0 to 1.

A task-oriented score of a refiner is indicative of a quality of an output of the refiner with respect to an identified task, for example training a classifier. In 814, processing unit 601 optionally computes a task oriented score. Optionally, computing the task oriented score comprises computing a training score of a first classifier model, trained using the plurality of refined output images, where the task oriented score is indicative of a success rate of the first classifier classifying a plurality of real images. Optionally, computing the task oriented score comprises computing a testing score of a second classifier model, trained using the plurality of real images, indicative of a success rate of the second classifier model classifying the plurality of refined output images. Optionally, processing unit 601 computes the task-oriented score using the training score and the testing score. The first classifier, and additionally or alternatively the second classifier, may be a semantic segmentation classifier trained to provide a semantic classification of each pixel of an input image. Optionally, the training score is a first F1 score indicative of a first percentage of pixels of each of the plurality of refined output images classified correctly by the first classifier model. Optionally, the testing score is a second F1 score indicative of a second percentage of pixels of each of the plurality of real images classified correctly by the second classifier model.

In 820, processing unit 601 optionally computes the model score using a combination of one or more of the plurality of image self-distance scores, the plurality of feature scores, the plurality of image quality scores and the task-oriented score. Optionally, processing unit 601 computes a weighted combination of the image self-distance scores, the realism scores, the image quality scores and the task oriented score. The weighted combination may reflect a tradeoff between one or more qualities of a set of refined output images generated by the refiner in response to the plurality of test images. Computing the model score optionally comprises computing one or more of a sum, an average and a standard deviation of one or more of the plurality of image self-distance scores, the plurality of realism scores and the plurality of image quality scores.

Reference is now made again to FIG. 8. In 720, processing unit 601 optionally selects an image refining model of the plurality of image refining models having a preferred model score of the plurality of model scores. Optionally, processing unit 601 outputs the selected image refining model, optionally via network interface 610 and alternatively or additionally optionally by storing the image refining model to at least one non-volatile digital storage 611.

Optionally, processing unit 601 provides the selected image refining model to at least one other processing unit for the purpose of performing an image refinement task, for example for the purpose of generating a simulated driving environment.

Reference is now made to FIG. 9, showing a flowchart schematically representing another optional flow of operations 900 for generating simulation data, according to some embodiments of the present invention. In such embodiments, in 901 a simulation generation processing unit accesses an image refining model trained by system 600 using method 700. Optionally, in 910 the simulation generation processing unit instructs another simulation generation processing unit to generate by the image refining model at least one refined output image in response to at least one input synthetic image, received via an input interface from an environment generation engine. Optionally, the other simulation generation processing unit is the other simulation generation processing unit. Optionally, the simulation generation processing unit receives the one or more synthetic input images from the environment generation engine via a network interface connected to the simulation generation processing unit. Optionally, the simulation generation processing unit executes the environment generation engine.

The descriptions of the various embodiments of the present invention have been presented for purposes of illustration, but are not intended to be exhaustive or limited to the embodiments disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art without departing from the scope and spirit of the described embodiments. The terminology used herein was chosen to best explain the principles of the embodiments, the practical application or technical improvement over technologies found in the marketplace, or to enable others of ordinary skill in the art to understand the embodiments disclosed herein.

It is expected that during the life of a patent maturing from this application many relevant autonomous drivers will be developed and the scope of the term "autonomous driver" is intended to include all such new technologies a priori. In addition, it is expected that during the life of a patent maturing from this application many relevant image self-distance scores, realism scores and task oriented scores will be developed and the scope of the terms image self-distance scores, realism scores and task oriented scores is intended to include all such new technologies a priori.

As used herein the term "about" refers to ± 10 %.

The terms "comprises", "comprising", "includes", "including", “having” and their conjugates mean "including but not limited to". This term encompasses the terms "consisting of" and "consisting essentially of".

The phrase "consisting essentially of" means that the composition or method may include additional ingredients and/or steps, but only if the additional ingredients and/or steps do not materially alter the basic and novel characteristics of the claimed composition or method.

As used herein, the singular form "a", "an" and "the" include plural references unless the context clearly dictates otherwise. For example, the term "a compound" or "at least one compound" may include a plurality of compounds, including mixtures thereof.

The word “exemplary” is used herein to mean “serving as an example, instance or illustration”. Any embodiment described as “exemplary” is not necessarily to be construed as preferred or advantageous over other embodiments and/or to exclude the incorporation of features from other embodiments.

The word “optionally” is used herein to mean “is provided in some embodiments and not provided in other embodiments”. Any particular embodiment of the invention may include a plurality of “optional” features unless such features conflict.

Throughout this application, various embodiments of this invention may be presented in a range format. It should be understood that the description in range format is merely for convenience and brevity and should not be construed as an inflexible limitation on the scope of the invention. Accordingly, the description of a range should be considered to have specifically disclosed all the possible subranges as well as individual numerical values within that range. For example, description of a range such as from 1 to 6 should be considered to have specifically disclosed subranges such as from 1 to 3, from 1 to 4, from 1 to 5, from 2 to 4, from 2 to 6, from 3 to 6 etc., as well as individual numbers within that range, for example, 1, 2, 3, 4, 5, and 6. This applies regardless of the breadth of the range.

Whenever a numerical range is indicated herein, it is meant to include any cited numeral (fractional or integral) within the indicated range. The phrases “ranging/ranges between” a first indicate number and a second indicate number and “ranging/ranges from” a first indicate number “to” a second indicate number are used herein interchangeably and are meant to include the first and second indicated numbers and all the fractional and integral numerals therebetween.

It is appreciated that certain features of the invention, which are, for clarity, described in the context of separate embodiments, may also be provided in combination in a single embodiment. Conversely, various features of the invention, which are, for brevity, described in the context of a single embodiment, may also be provided separately or in any suitable

subcombination or as suitable in any other described embodiment of the invention. Certain features described in the context of various embodiments are not to be considered essential features of those embodiments, unless the embodiment is inoperative without those elements.

All publications, patents and patent applications mentioned in this specification are herein incorporated in their entirety by reference into the specification, to the same extent as if each individual publication, patent or patent application was specifically and individually indicated to be incorporated herein by reference. In addition, citation or identification of any reference in this application shall not be construed as an admission that such reference is available as prior art to the present invention. To the extent that section headings are used, they should not be construed as necessarily limiting.

In addition, any priority document(s) of this application is/are hereby incorporated herein by reference in its/their entirety.

WHAT IS CLAIMED IS:

1. A method for training a model for generating simulation data for training an autonomous driving agent, comprising:

analyzing real input data, collected from a driving environment, to identify a plurality of environment classes, a plurality of moving agent classes, and a plurality of movement pattern classes;

generating a training environment, according to one environment class of the plurality of environment classes; and

in at least one of a plurality of training iterations:

generating, by a simulation generation model, a simulated driving environment according to the training environment and according to a plurality of generated training agents, each associated with one of the plurality of moving agent classes and one of the plurality of movement pattern classes;

collecting simulated driving data from the simulated driving environment; and

modifying at least one model parameter of the simulation generation model to minimize a difference between a simulation statistical fingerprint, computed using the simulated driving data, and a real statistical fingerprint, computed using the real input data.

2. The method of claim 1, wherein the real input data comprises environment data and agent data;

wherein the environment data comprises a plurality of environment values of a plurality of environment attributes describing the driving environment; and

wherein the agent data comprises for each of a plurality of agents operating in the driving environment a plurality of agent values of a plurality of agent attributes, each plurality of agent values describing an appearance of the respective agent and a behavior in the driving environment of the respective agent.

3. The method of claim 2, wherein at least one of the plurality of environment values is selected from a group of environment values consisting of: a time of day value, a day of week value, a month of year value, a temperature value, an air clarity metric value, a precipitation value, a precipitation type value, a wind velocity value, a wind direction value, an amount of light value, an artificial light indication value, a crowd density metric value, an incline value, a curve direction value, a curve radius value, and a road horizontal angle value.

4. The method of claim 2, wherein at least one of the plurality of agent values is selected from a group of agent values consisting of: a height value, a width value, a longitudinal velocity value, a lateral velocity value, a longitudinal acceleration value, a lateral acceleration value, a longitudinal distance value, a lateral distance value, a longitudinal direction value, a lateral direction value, a traffic rule violation indication value, an opposite lane indication value, a shoulder indication value, a collision indication value, an amount of passes of other agents, a transition time value, a transition delay time value, an outbound transition angle value, and an inbound transition angle value.

5. The method of claim 2, wherein each of the plurality of environment classes comprises at least some of the plurality of environment values.

6. The method of claim 2, wherein each of the plurality of moving agent classes describes a moving agent selected from a group of possible moving agents consisting of: a car, a truck, a motorized vehicle, a train, a boat, an air-born vehicle, a waterborne vessel, a motorized scooter, a scooter, a bicycle, and a pedestrian; and

wherein each of the plurality of moving agent classes comprises a plurality of agent probabilities, each agent probability indicative of a likelihood of a moving agent associated with the respective moving agent class to demonstrate an agent behavior described by an identified plurality of agent values of the plurality of agent values when the driving environment is described by a first environment class of the plurality of environment classes.

7. The method of claim 2, wherein each of the plurality of movement pattern classes describes an actor selected from a group of possible actors consisting of: a pedestrian, a bicycle rider, a vehicle driver; and

wherein each of the plurality of movement pattern classes comprises a plurality of actor probabilities, each actor probability indicative of a likelihood of an actor associated with the respective movement pattern class to demonstrate an actor behavior described by an identified movement pattern plurality of agent values of the plurality of agent values when the driving environment is described by a second environment class of the plurality of environment classes.

8. The method of claim 2, wherein analyzing the real input data comprises applying a transformation to the plurality of agent values.

9. The method of claim 2 wherein the real statistical fingerprint is computed using the plurality of environment values and the plurality of agent values.

10. The method of claim 2 wherein the simulated driving data comprises simulated environment data and simulated agent data;

wherein the simulated environment data comprises a plurality of simulated environment values of the plurality of environment attributes describing the simulated driving environment;

wherein the simulated agent data comprises for each of the plurality of generated training agents a plurality of simulated agent values of the plurality of agent attributes, each plurality of simulated agent values describing an appearance of the respective generated training agent and a behavior in the simulated driving environment of the respective generated training agent; and

wherein the simulation statistical fingerprint is computed using the plurality of simulation environment attribute values and the plurality of simulation agent values.

11. The method of claim 1, wherein at least some of the plurality of environment classes, the plurality of moving agent classes and the plurality of movement pattern classes are identified using at least one machine learning model to analyze the real input data.

12. The method of claim 1, wherein at least some of the plurality of environment classes, the plurality of moving agent classes and the plurality of movement pattern classes are identified by applying at least one statistical analysis method to the real input data.

13. The method of claim 1, wherein the real input data further comprises at least one label indicative of one or more environmental conditions when at least some of the real input data is collected from the driving environment; and

wherein analyzing the real input data comprises using at least some of the at least one label.

14. The method of claim 1, wherein generating the simulated driving environment according to the plurality of generated training agents comprises:

associating each of the plurality of generated training agents a moving agent class of the plurality of moving agent classes and a movement pattern class of the plurality of movement classes; and

generating for each of the plurality of generated training agents simulated behavior data in accordance with: 1) the plurality of agent probabilities of the respective moving agent class

according to the one environment class, and 2) the plurality of actor probabilities of the respective movement pattern class according to the one environment class.

15. The method of claim 14, further comprising computing a probability score for each of the plurality of moving agent classes and each of the plurality of movement pattern classes;

wherein a moving agent class of the plurality of moving agent classes is associated with a generated training agent according to the respective probability score; and

wherein a movement pattern class of the plurality of movement pattern classes is associated with a generated training agent according to the respective probability score.

16. The method of claim 15, further comprising:

identifying in the plurality of moving agent classes at least one less probable moving agent classes according to the respective probability score; and

associating at least one of the generated training agents with one of the at least one less probable moving agent classes according to an agent probability higher than an initial agent probability in accordance with the respective probability score.

17. The method of claim 15, further comprising:

identifying in the plurality of movement pattern classes at least one less probable movement pattern classes according to the respective probability score; and

associating at least one of the generated training agents with one of the at least one less probable movement pattern classes according to an actor probability higher than an initial actor probability in accordance with the respective probability score.

18. The method of claim 1, further comprising:

generating another training environment, according to another environment class of the plurality of environment classes; and

in each of a plurality of additional training iterations:

generating, by another simulation generation model, another simulated driving environment according to the other training environment and according to a plurality of other generated training agents, each associated with one of the plurality of moving agent classes and one of the plurality of movement pattern classes;

collecting other simulated driving data from the other simulated driving environment; and

modifying a plurality of other model parameters of the other simulation generation model to minimize another difference between another simulation statistical fingerprint, computed using the other simulated driving data, and the real statistical fingerprint, computed using the real input data.

19. A system for training a model for generating simulation data for training an autonomous driving agent, comprising at least one hardware processor adapted to:

analyzing real input data, collected from a driving environment, to identify a plurality of environment classes, a plurality of moving agent classes, and a plurality of movement pattern classes;

generating a training environment, according to one environment class of the plurality of environment classes;

in each of a plurality of training iterations:

generating, by a simulation generation model, a simulated driving environment according to a plurality of generated training agents, each associated with one of the plurality of moving agent classes and one of the plurality of movement pattern classes;

collecting simulated driving data from the simulated driving environment; and

modifying a plurality of model parameters of the simulation generation model to minimize a difference between a simulation statistical fingerprint, computed using the simulated driving data, and a real statistical fingerprint, computed using the real input data.

20. The system of claim 19, wherein the real input data comprises real data collected by at least one sensor selected from a group of sensors comprising: a camera, an electromagnetic radiation sensor, a radar, a Light Detection and Ranging (LIDAR) sensor, a microphone, a thermometer, an accelerometer, and a video camera.

21. The system of claim 19, wherein the at least one hardware processor is further adapted to output the simulation generation model to a non-volatile digital storage connected to the at least one hardware processor.

22. The system of claim 19, wherein the at least one hardware processor is further adapted to receiving the real input data via at least one digital communication network interface connected to the at least one hardware processor.

23. The system of claim 22, wherein the at least one hardware processor is further adapted to output the simulation generation model via the at least one digital communication network interface.

24. A method for creating data for training an autonomous driving agent, comprising:

accessing a simulation generation model trained by:

analyzing real input data, collected from a driving environment, to identify a plurality of environment classes, a plurality of moving agent classes, and a plurality of movement pattern classes;

generating a training environment, according to one environment class of the plurality of environment classes;

in each of a plurality of training iterations:

generating, by the simulation generation model, a simulated training driving environment according to the training environment and according to a plurality of generated training agents, each associated with one of the plurality of moving agent classes and one of the plurality of movement pattern classes;

collecting simulated driving data from the simulated training driving environment; and

modifying a plurality of model parameters of the simulation generation model to minimize a difference between a simulation statistical fingerprint, computed using the simulated driving data, and a real statistical fingerprint, computed using the real input data;

receiving a new environment class of the plurality of environment classes; and

instructing at least one other hardware processor to:

generate a simulated environment, according to the new environment class;

generate a plurality of simulated agents, each associated with one of another plurality of moving agent classes and another plurality of movement pattern classes; and

generate a simulated driving environment by the simulation generation model according to the simulation environment and the plurality of simulated agents.

25. A method for training an image refining model for generating simulation data for training an autonomous driving agent, comprising:

for each of a plurality of image refining models, training the image refining model to generate a refined image in response to input comprising a synthetic image;

computing a plurality of model scores by computing for each of the plurality of image refining models a model score indicative of a quality of realism of the respective refined image generated by the respective image refining model in response to input comprising the synthetic image, and indicative of a quality of content preservation of the respective refined image with reference to the synthetic image; and

selecting an image refining model of the plurality of image refining models having a preferred model score of the plurality of model scores.

26. The method of claim 25, wherein the model score is computed by:

producing a plurality of refined output images, each generated by the respective image refining model in response to input comprising one of a plurality of test images;

computing a plurality of image self-distance scores, each indicative of a quality of content preservation in one of the plurality of refined output images with reference to the respective test image;

computing a plurality of realism scores, each indicative of a quality of realism of one of the plurality of refined output images;

computing a plurality of image quality scores each indicative of an image quality of one of the plurality of refined output images;

computing a task-oriented score by:

computing a training score of a first classifier model, trained using the plurality of refined output images, indicative of a success rate of the first classifier model classifying a plurality of real images; and

computing a testing score of a second classifier model, trained using the plurality of real images, indicative of a success rate of the second classifier model classifying the plurality of refined output images; and

computing the model score using the plurality of image self-distance scores, the plurality of feature scores, the plurality of image quality scores and the task-oriented score.

27. The method of claim 26, wherein computing the model score comprises computing at least one term selected from a group of terms consisting of: an average of the plurality of image self-distance scores, a standard deviation of the plurality of image self-distance scores, a sum of the plurality of image self-distance scores, an average of the plurality of realism scores, a standard deviation of the plurality of realism scores, a sum of the plurality of realism scores, an average of

the plurality of image quality scores, a standard deviation of the plurality of image quality scores, and a sum of the plurality of image quality scores.

28. The method of claim 26, wherein computing the plurality of realism scores comprises computing a plurality of feature scores, each indicative of a distance between: 1) a plurality of refined features in one of the plurality of refined output images, and 2) a plurality of features extracted from a target real image equivalent to the respective test image.

29. The method of claim 28, wherein at least one of the plurality of feature scores is a Frechet Inception Distance (FID).

30. The method of claim 26, wherein computing the plurality of realism scores comprises computing a plurality of reconstructions scores, each indicative of a difference between a refined image of the plurality of refined output images and a reconstructed image, reconstructed from a corrupted refined image by a reconstruction model trained to reconstruct real images.

31. The method of claim 26, wherein each of the plurality of image-quality scores is normalized in a range from 0 to 1.

32. The method of claim 26, wherein each of the plurality of realism scores is normalized in a range from 0 to 1.

33. The method of claim 26, wherein each of the plurality of image self-distance scores is normalized in a range from 0 to 1.

34. The method of claim 26, wherein the first classifier is a first semantic segmentation classifier; and

wherein the training score is a first F1 score indicative of a first percentage of pixels of each of the plurality of refined output images classified correctly by the first classifier.

35. The method of claim 26, wherein the second classifier is a second semantic segmentation classifier; and

wherein the testing score is a second F1 score indicative of a second percentage of pixels of each of the plurality of real images classified correctly by the second classifier.

36. A system for training an image refining model for generating simulation data for training an autonomous driving agent, comprising at least one hardware processor adapted to:

for each of a plurality of image refining models, training the image refining model to generate a refined image in response to input comprising a synthetic image;

computing a plurality of model scores by computing for each of the plurality of image refining models a model score indicative of a quality of realism of the respective refined image generated by the respective image refining model in response to input comprising the synthetic image, and indicative of a quality of content preservation of the respective refined image with reference to the synthetic image; and

selecting an image refining model of the plurality of image refining models having a preferred model score of the plurality of model scores.

37. A method for generating simulation data for training an autonomous driving agent, comprising:

accessing an image refining model trained by:

for each of a plurality of image refining models, training the image refining model to generate a refined image in response to input comprising a synthetic image;

computing a plurality of model scores by computing for each of the plurality of image refining models a model score indicative of a quality of realism of the respective refined image generated by the respective image refining model in response to input comprising the synthetic image, and indicative of a quality of content preservation of the respective refined image with reference to the synthetic image; and

selecting a computer generated model of the plurality of computer generated models having a preferred model score of the plurality of model scores; and

instructing at least one hardware processor to generate by the image refining model at least one refined output image in response to at least one input synthetic image, received via an input interface from an environment generation engine.

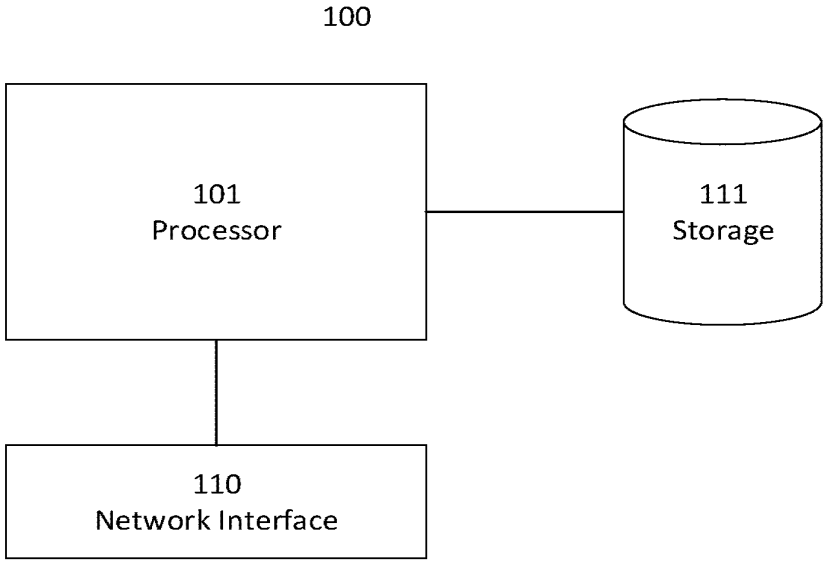


FIG. 1

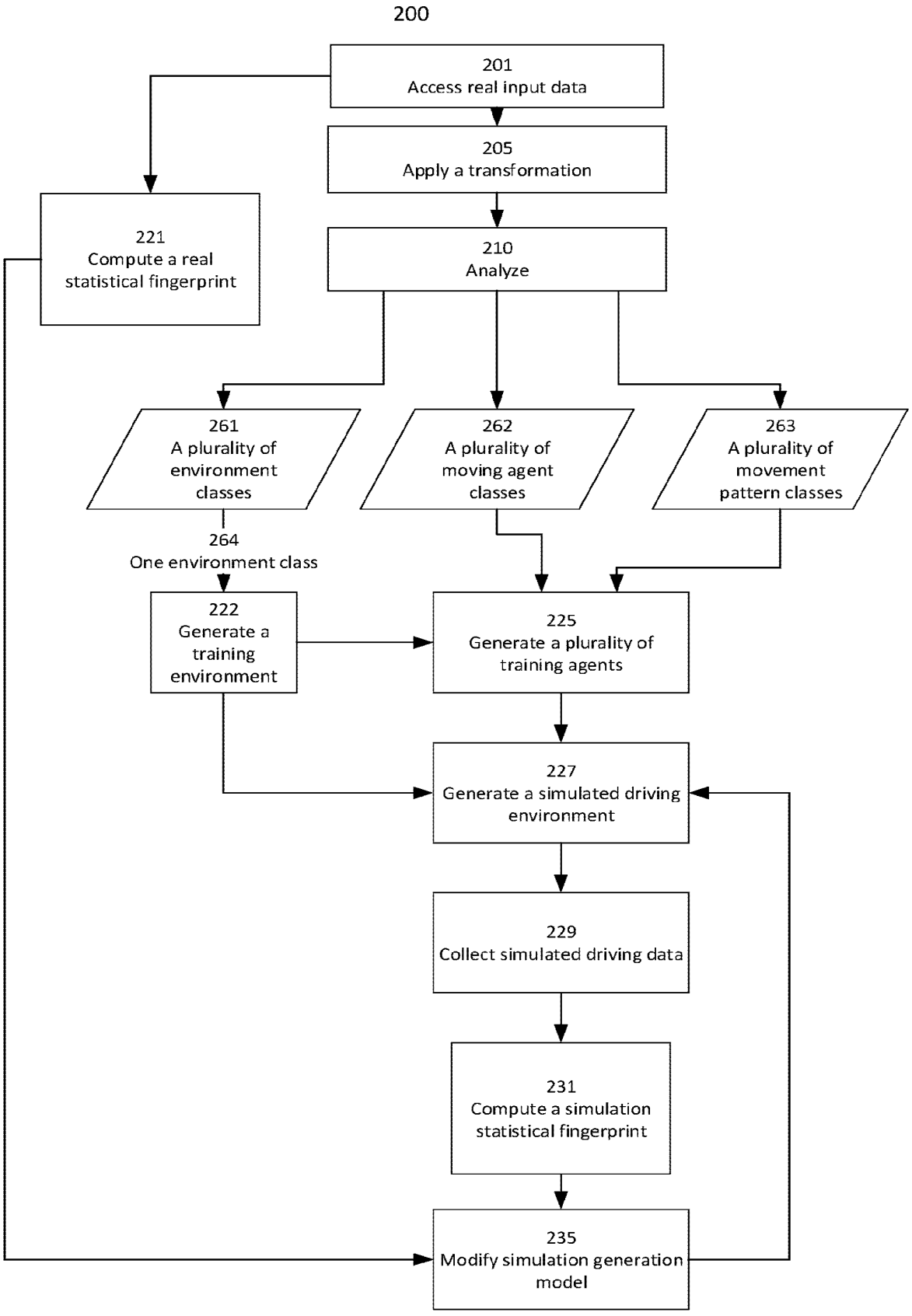

200
201
Access real input data
205
Apply a transformation
221
Compute a real statistical fingerprint
210
Analyze
261
A plurality of environment classes
262
A plurality of moving agent classes
263
A plurality of movement pattern classes
264
One environment class
222
Generate a training environment
225
Generate a plurality of training agents
227
Generate a simulated driving environment
229
Collect simulated driving data
231
Compute a simulation statistical fingerprint
235
Modify simulation generation model

FIG. 2

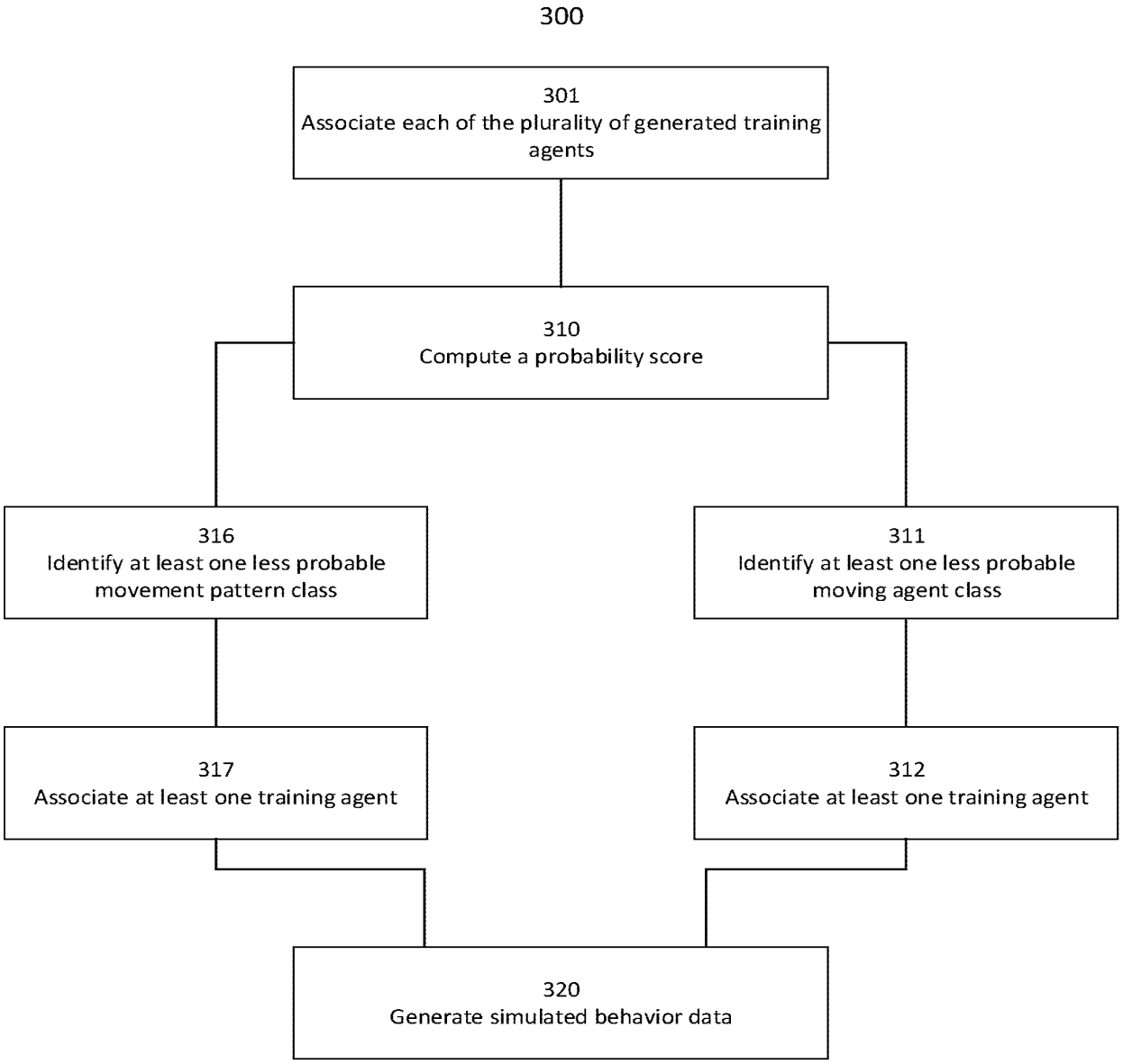

300
301
Associate each of the plurality of generated training agents
310
Compute a probability score
316
Identify at least one less probable movement pattern class
311
Identify at least one less probable moving agent class
317
Associate at least one training agent
312
Associate at least one training agent
320
Generate simulated behavior data

FIG. 3

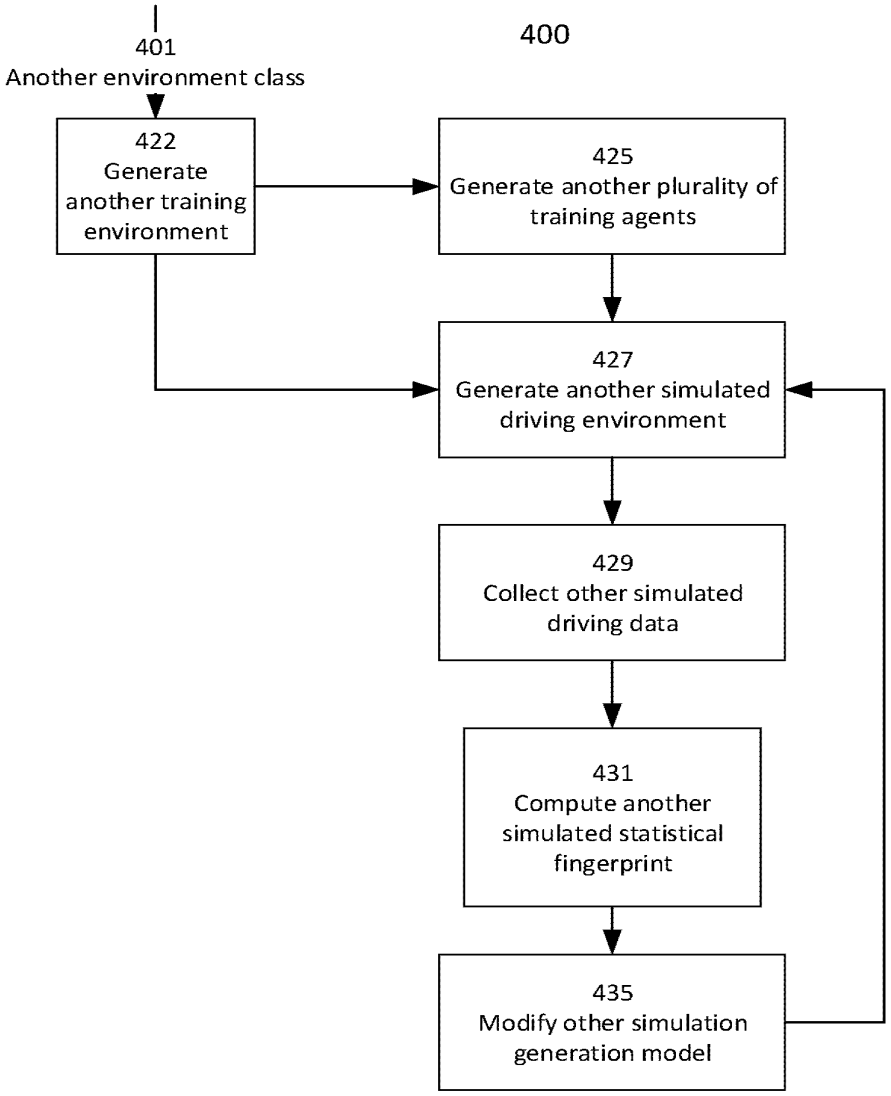

400
401
Another environment class
422
Generate another training environment
425
Generate another plurality of training agents
427
Generate another simulated driving environment
429
Collect other simulated driving data
431
Compute another simulated statistical fingerprint
435
Modify other simulation generation model

FIG. 4

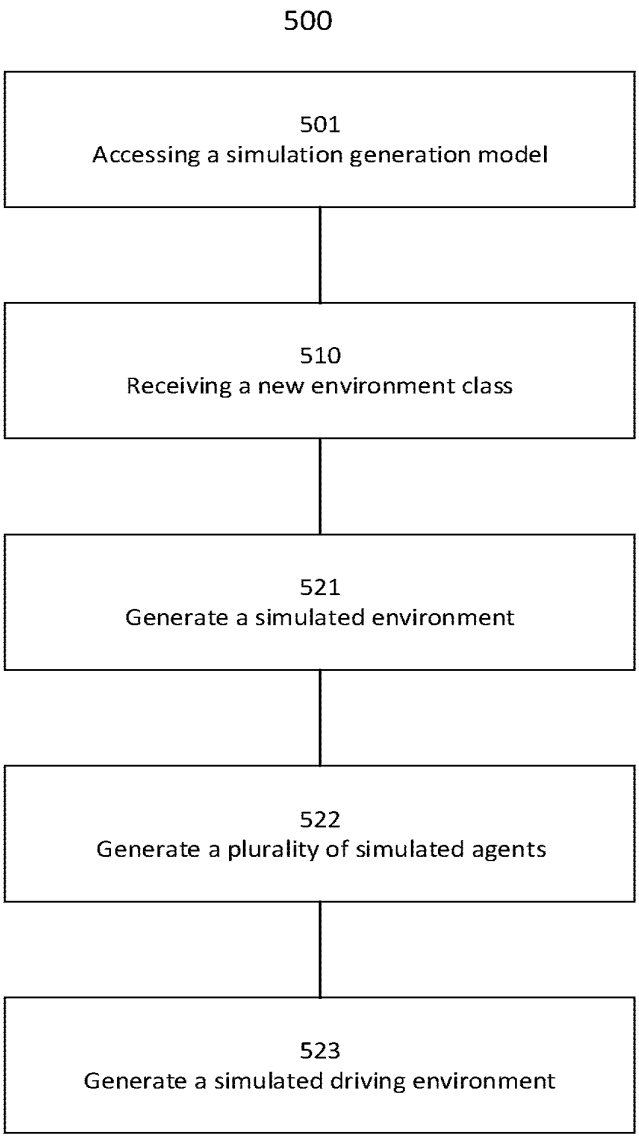

500
501
Accessing a simulation generation model
510
Receiving a new environment class
521
Generate a simulated environment
522
Generate a plurality of simulated agents
523
Generate a simulated driving environment

FIG. 5

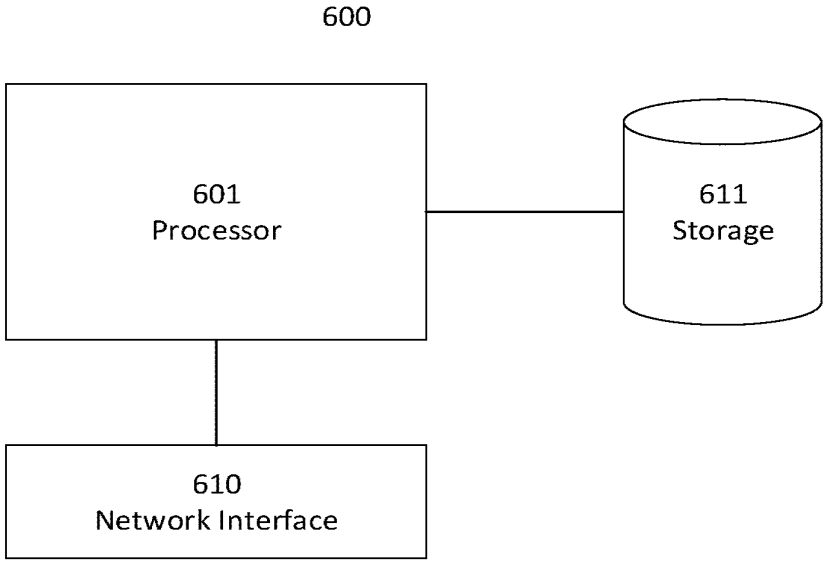


FIG. 6

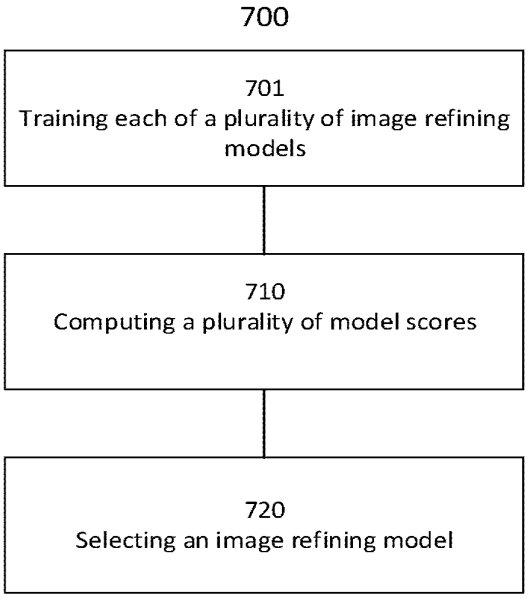

700
701
Training each of a plurality of image refining models
710
Computing a plurality of model scores
720
Selecting an image refining model

FIG. 7

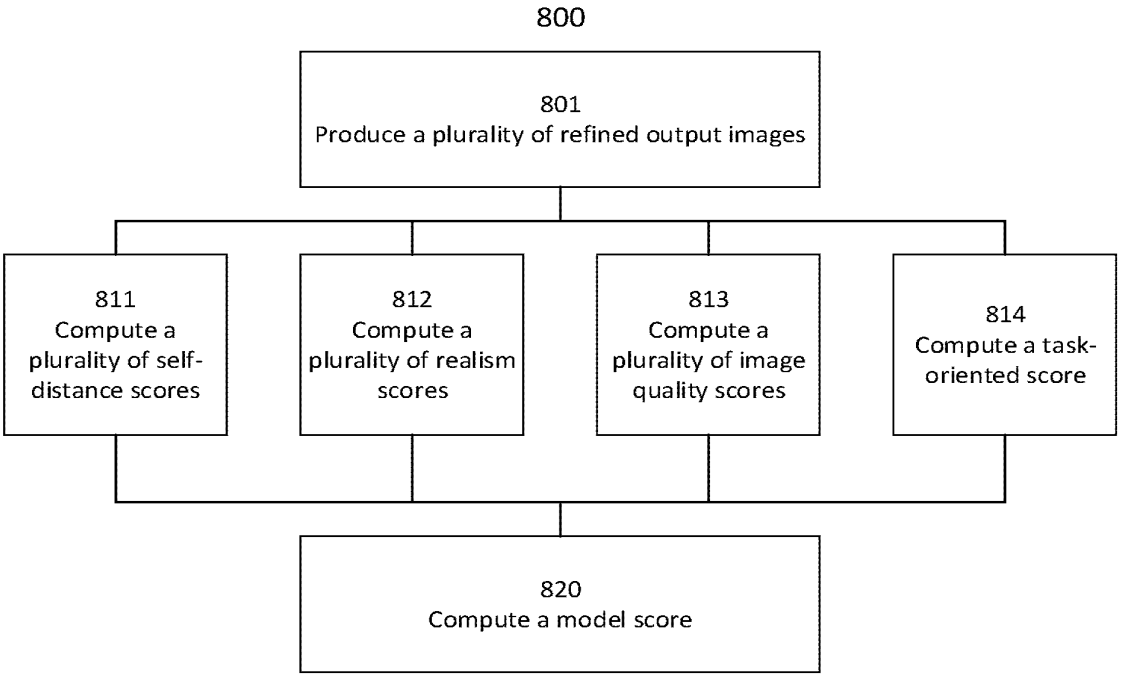

800
801
Produce a plurality of refined output images
811
Compute a plurality of self-distance scores
812
Compute a plurality of realism scores
813
Compute a plurality of image quality scores
814
Compute a task-oriented score
820
Compute a model score

FIG. 8

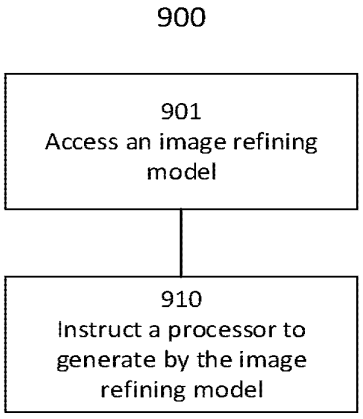

900
901
Access an image refining model
910
Instruct a processor to generate by the image refining model

FIG. 9

INTERNATIONAL SEARCH REPORT

international application No.
PCT/IL 2019/051193

A. CLASSIFICATION OF SUBJECT MATTER

IPC - G05D 1/00, 1/02; G06T 7/579, 17/05 (2020.01)

CPC - G06F 30/20; G05D 1/00, 1/02, 1/0088, 1/0221; G06T 7/579, 17/05, 19/006

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
See Search History document

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
See Search History document

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
See Search History document

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2018/002910 A1 (COGNATA LTD) 04 January 2018; page 10, lines 9-15, page 11, lines 23-25, page 19, lines 12-21	1-14, 18-24 --- 15-17, 25-37
A	WO 2017/079229 A1 (ZOOX, INC) 11 May 2017; entire document	1-37
A	US 2013/0184928 A1 (KERKHOF, B et al.) 18 July 2013; entire document	1-37
A	US 2016/0210775 A1 (FORD GLOBAL TECHNOLOGIES, LLC) 21 July 2016; entire document	1-37

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"D" document cited by the applicant in the international application
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
02 March 2020 (02.03.2020)	12 MAR 2020
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-8300	Authorized officer Shane Thomas Telephone No. PCT Helpdesk: 571-272-4300

Form PCT/ISA/210 (second sheet) (July 2019)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IL19/51119

Box No. II Observations where certain claims were found unsearchable (Continuation of item 2 of first sheet)

This international search report has not been established in respect of certain claims under Article 17(2)(a) for the following reasons:

1. ☐ Claims Nos.:
because they relate to subject matter not required to be searched by this Authority, namely:

2. ☐ Claims Nos.:
because they relate to parts of the international application that do not comply with the prescribed requirements to such an extent that no meaningful international search can be carried out, specifically:

3. ☐ Claims Nos.:
because they are dependent claims and are not drafted in accordance with the second and third sentences of Rule 6.4(a).

Box No. III Observations where unity of invention is lacking (Continuation of item 3 of first sheet)

This International Searching Authority found multiple inventions in this international application, as follows:

See Extra Sheet

1. ☒ As all required additional search fees were timely paid by the applicant, this international search report covers all searchable claims.

2. ☐ As all searchable claims could be searched without effort justifying additional fees, this Authority did not invite payment of additional fees.

3. ☐ As only some of the required additional search fees were timely paid by the applicant, this international search report covers only those claims for which fees were paid, specifically claims Nos.:

4. ☐ No required additional search fees were timely paid by the applicant. Consequently, this international search report is restricted to the invention first mentioned in the claims; it is covered by claims Nos.:

Remark on Protest

☐ The additional search fees were accompanied by the applicant's protest and, where applicable, the payment of a protest fee.

☐ The additional search fees were accompanied by the applicant's protest but the applicable protest fee was not paid within the time limit specified in the invitation.

☒ No protest accompanied the payment of additional search fees.

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IL19/51119

-***-Continued from Box No. III Observations where unity of invention is lacking-***-

This application contains the following inventions or groups of inventions which are not so linked as to form a single general inventive concept under PCT Rule 13.1. In order for all inventions to be examined, the appropriate additional examination fee must be paid.

Group I: Claims 1-24 are directed towardsa method and system for training a model, and a method for creating data, the methods and system comprising: generating a training environment.
Group II: Claims 25-37 are directed towards a method for training an image refining model for generating simulation data for training an autonomous driving agent.

The inventions listed as Groups I-II do not relate to a single general inventive concept under PCT Rule 13.1 because, under PCT Rule 13.2, they lack the same or corresponding special technical features for the following reasons:

The special technical features of Group I are at least a method for training a model for generating simulation data for training an autonomous driving agent, comprising: analyzing real input data, collected from a driving environment, to identify a plurality of environment classes, a plurality of moving agent classes, and a plurality of movement pattern classes; generating a training environment, according to one environment class of the plurality of environment classes, which are not present in Group II.

The special technical features of Group II are at least a method for training an image refining model for generating simulation data for training an autonomous driving agent: for each of a plurality of image refining models, training the image refining model to generate a refined image in response to input comprising a synthetic image, which are not present in Group I.

The common technical features shared by Groups I-II are training a model for generating simulation data for training an autonomous driving agent comprising at least one hardware processor.

However, these common features are previously disclosed by WO 2018/002910 A1 to COGNATA LTD. (hereinafter 'COGNATA'). COGNATA discloses training a model for generating simulation data for training an autonomous driving agent comprising at least one hardware processor (training a model for generating simulation data for training an autonomous driving system comprising a processor; claims 1 and 18 of COGNATA).

Since the common technical features are previously disclosed by COGNATA, these common features are not special and so Groups I-II lack unity

PTO/AIA/14 (01-22)
Approved for use through 05/31/2024. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	15305FTNK
	Application Number	
Title of Invention	RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH	

The application data sheet is part of the provisional or nonprovisional application for which it is being submitted. The following form contains the bibliographic data arranged in a format specified by the United States Patent and Trademark Office as outlined in 37 CFR 1.76.
This document may be completed electronically and submitted to the Office in electronic format using the Electronic Filing System (EFS) or the document may be printed and included in a paper filed application.

Secrecy Order 37 CFR 5.2:

☐ Portions or all of the application associated with this Application Data Sheet may fall under a Secrecy Order pursuant to 37 CFR 5.2 (Paper filers only. Applications that fall under Secrecy Order may not be filed electronically.)

Inventor Information:

Inventor 1 Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Jason		Crabtree	

Residence Information (Select One) ◉ US Residency ○ Non US Residency ○ Active US Military Service

City	Vienna	State/Province	VA	Country of Residence	US

Mailing Address of Inventor:

Address 1	10208 Westford Drive		
Address 2			
City	Vienna	State/Province	VA
Postal Code	22182	Country	US

Inventor 2 Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Andrew		Sellers	

Residence Information (Select One) ◉ US Residency ○ Non US Residency ○ Active US Military Service

City	Monument	State/Province	CO	Country of Residence	US

Mailing Address of Inventor:

Address 1	16258 Windsor Creek Dr.		
Address 2			
City	Monument	State/Province	CO
Postal Code	80132	Country	US

All Inventors Must Be Listed - Additional Inventor Information blocks may be generated within this form by selecting the **Add** button. Add

Correspondence Information:

Enter either Customer Number or complete the Correspondence Information section below.
For further information see 37 CFR 1.33(a).

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	15305FTNK
	Application Number	
Title of Invention	RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH	

☐ An Address is being provided for the correspondence Information of this application.			
Customer Number	82318		
Email Address	uspto@galvinpatentlaw.com	Add Email	Remove Email

Application Information:

Title of the Invention	RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH		
Attorney Docket Number	15305FTNK	**Small Entity Status Claimed**	☐
Application Type	Nonprovisional		
Subject Matter	Utility		
Total Number of Drawing Sheets (if any)	16	**Suggested Figure for Publication (if any)**	1

Filing By Reference:

Only complete this section when filing an application by reference under 35 U.S.C. 111(c) and 37 CFR 1.57(a). Do not complete this section if application papers including a specification and any drawings are being filed. Any domestic benefit or foreign priority information must be provided in the appropriate section(s) below (i.e., "Domestic Benefit/National Stage Information" and "Foreign Priority Information").

For the purposes of a filing date under 37 CFR 1.53(b), the description and any drawings of the present application are replaced by this reference to the previously filed application, subject to conditions and requirements of 37 CFR 1.57(a).

Application number of the previously filed application	Filing date (YYYY-MM-DD)	Intellectual Property Authority or Country

Publication Information:

☐ Request Early Publication (Fee required at time of Request 37 CFR 1.219)

☐ **Request Not to Publish.** I hereby request that the attached application not be published under 35 U.S.C. 122(b) and certify that the invention disclosed in the attached application **has not and will not** be the subject of an application filed in another country, or under a multilateral international agreement, that requires publication at eighteen months after filing.

Representative Information:

Representative information should be provided for all practitioners having a power of attorney in the application. Providing this information in the Application Data Sheet does not constitute a power of attorney in the application (see 37 CFR 1.32).
Either enter Customer Number or complete the Representative Name section below. If both sections are completed the customer Number will be used for the Representative Information during processing.

Please Select One:	◉ Customer Number	○ US Patent Practitioner	○ Limited Recognition (37 CFR 11.9)
Customer Number	82318		

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	15305FTNK
	Application Number	
Title of Invention	RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH	

Domestic Benefit/National Stage Information:

This section allows for the applicant to either claim benefit under 35 U.S.C. 119(e), 120, 121, 365(c), or 386(c) or indicate National Stage entry from a PCT application. Providing benefit claim information in the Application Data Sheet constitutes the specific reference required by 35 U.S.C. 119(e) or 120, and 37 CFR 1.78.
When referring to the current application, please leave the "Application Number" field blank.

Prior Application Status	Pending		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
	Continuation of	18581375	2024-02-20
Prior Application Status	Pending		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
18581375	Continuation of	17189161	2021-03-01
Prior Application Status	Patented		Remove

Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
17189161	Continuation in part of	17061195	2020-10-01	11570214	2023-01-31
Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
17061195	Continuation in part of	17035029	2020-09-28	11546380	2023-01-03
Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
17035029	Continuation in part of	17008276	2020-08-31	11323484	2022-05-03
Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
17008276	Continuation in part of	17000504	2020-08-24	11477245	2022-10-18
Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
17000504	Continuation in part of	16855724	2020-04-22	11218510	2022-01-04
Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
16855724	Continuation in part of	16836717	2020-03-31	10917428	2021-02-09

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	15305FTNK
	Application Number	
Title of Invention	RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH	

Prior Application Status	Patented			Remove	
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
16836717	Continuation in part of	15887496	2018-02-02	10783241	2020-09-22
Prior Application Status	Patented			Remove	
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
15887496	Continuation in part of	15823285	2017-11-27	10740096	2020-08-11
Prior Application Status	Patented			Remove	
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
15823285	Continuation in part of	15788718	2017-10-19	10861014	2020-12-08
Prior Application Status	Abandoned			Remove	
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)		
15788718	Continuation in part of	15788002	2017-10-19		
Prior Application Status	Patented			Remove	
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
15788002	Continuation in part of	15787601	2017-10-18	10860660	2020-12-08
Prior Application Status	Expired			Remove	
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)		
15787601	Claims benefit of provisional	62568312	2017-10-04		
Prior Application Status	Abandoned			Remove	
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)		
15787601	Continuation in part of	15616427	2017-06-07		
Prior Application Status	Abandoned			Remove	
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)		
15616427	Continuation in part of	14925974	2015-10-28		
Prior Application Status	Expired			Remove	
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)		
15788002	Claims benefit of provisional	62568305	2017-10-04		

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	15305FTNK
	Application Number	
Title of Invention	RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH	

Prior Application Status	Expired		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
15788718	Claims benefit of provisional	62568307	2017-10-04

Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
15887496	Continuation in part of	15818733	2017-11-20	10673887	2020-06-02

Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
15818733	Continuation in part of	15725274	2017-10-04	10609079	2020-03-31

Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
15725274	Continuation in part of	15655113	2017-07-20	10735456	2020-08-04

Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
15655113	Continuation in part of	15616427	2017-06-07

Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
15655113	Continuation in part of	15237625	2016-08-15	10248910	2019-04-02

Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
15237625	Continuation in part of	15206195	2016-07-08

Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
15206195	Continuation in part of	15186453	2016-06-18

Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
15186453	Continuation in part of	15166158	2016-05-26

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	15305FTNK
	Application Number	
Title of Invention	RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH	

Prior Application Status	Patented			Remove	
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
15166158	Continuation in part of	15141752	2016-04-28	10860962	2020-12-08
Prior Application Status	Patented			Remove	
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
15141752	Continuation in part of	15091563	2016-04-05	10204147	2019-02-12
Prior Application Status	Patented			Remove	
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
15141752	Continuation in part of	14986536	2015-12-31	10210255	2019-02-19
Prior Application Status	Abandoned			Remove	
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)		
15141752	Continuation in part of	14925974	2015-10-28		
Prior Application Status	Patented			Remove	
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
16855724	Continuation in part of	16777270	2020-01-30	11025674	2021-06-01
Prior Application Status	Patented			Remove	
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
16777270	Continuation in part of	16720383	2019-12-19	10944795	2021-03-09
Prior Application Status	Patented			Remove	
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
16720383	Continuation of	15823363	2017-11-27	10560483	2020-02-11
Prior Application Status	Patented			Remove	
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
15823363	Continuation in part of	15725274	2017-10-04	10609079	2020-03-31
Prior Application Status	Patented			Remove	
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
17000504	Continuation in part of	16412340	2019-05-14	11539663	2022-12-27
Prior Application Status	Abandoned			Remove	
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)		
16412340	Continuation in part of	16267893	2019-02-05		

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	15305FTNK
	Application Number	
Title of Invention	RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH	

Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
16267893	Continuation in part of	16248133	2019-01-15

Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
16248133	Continuation in part of	15849901	2017-12-21	11023284	2021-06-01

Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
15849901	Continuation in part of	15835436	2017-12-07	10572828	2020-02-25

Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
15835436	Continuation in part of	15790457	2017-10-23	10884999	2021-01-05

Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
15790457	Continuation in part of	15790327	2017-10-23	10860951	2020-12-08

Prior Application Status	Expired		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
15790327	Claims benefit of provisional	62568291	2017-10-04

Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
15790327	Continuation in part of	15616427	2017-06-07

Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
15790327	Continuation in part of	15141752	2016-04-28	10860962	2020-12-08

Prior Application Status	Expired		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
15790457	Claims benefit of provisional	62568298	2017-10-04

Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
15849901	Continuation in part of	15835312	2017-12-07	11055451	2021-07-06

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	15305FTNK
	Application Number	
Title of Invention	RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH	

Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
15835312	Continuation in part of	15186453	2016-06-18
Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
16248133	Continuation in part of	15813097	2017-11-14
Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
15813097	Continuation in part of	15616427	2017-06-07
Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
16248133	Continuation in part of	15806697	2017-11-08
Prior Application Status	Patented		Remove

Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
15806697	Continuation in part of	15376657	2016-12-13	10402906	2019-09-03
Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
15376657	Continuation in part of	15237625	2016-08-15	10248910	2019-04-02
Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
15806697	Continuation in part of	15343209	2016-11-04	11087403	2021-08-10
Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
15343209	Continuation in part of	15237625	2016-08-15	10248910	2019-04-02
Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
15343209	Continuation in part of	15229476	2016-08-05	10454791	2019-10-22

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	15305FTNK
	Application Number	
Title of Invention	RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH	

Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
15229476	Continuation in part of	15206195	2016-07-08
Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
16248133	Continuation in part of	15673368	2017-08-09
Prior Application Status	Patented		Remove

Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
15673368	Continuation in part of	15376657	2016-12-13	10402906	2019-09-03

Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
17061195	Continuation in part of	15879801	2018-01-25
Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
15879801	Continuation in part of	15379899	2016-12-15
Prior Application Status	Patented		Remove

Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
15379899	Continuation in part of	15376657	2016-12-13	10402906	2019-09-03

Prior Application Status	Patented		Remove

Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
17189161	Continuation in part of	16709598	2019-12-10	11507858	2022-11-22

Prior Application Status	Abandoned		Remove
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)
16709598	Continuation in part of	14925974	2015-10-28

Additional Domestic Benefit/National Stage Data may be generated within this form by selecting the **Add** button. Add

Foreign Priority Information:

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	15305FTNK
	Application Number	
Title of Invention	RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH	

This section allows for the applicant to claim priority to a foreign application. Providing this information in the application data sheet constitutes the claim for priority as required by 35 U.S.C. 119(b) and 37 CFR 1.55. When priority is claimed to a foreign application that is eligible for retrieval under the priority document exchange program (PDX)[i] the information will be used by the Office to automatically attempt retrieval pursuant to 37 CFR 1.55(i)(1) and (2). Under the PDX program, applicant bears the ultimate responsibility for ensuring that a copy of the foreign application is received by the Office from the participating foreign intellectual property office, or a certified copy of the foreign priority application is filed, within the time period specified in 37 CFR 1.55(g)(1).

Remove

Application Number	Country[i]	Filing Date (YYYY-MM-DD)	Access Code[i] (if applicable)

Additional Foreign Priority Data may be generated within this form by selecting the **Add** button. Add

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications

☐ This application (1) claims priority to or the benefit of an application filed before March 16, 2013 and (2) also contains, or contained at any time, a claim to a claimed invention that has an effective filing date on or after March 16, 2013.
NOTE: By providing this statement under 37 CFR 1.55 or 1.78, this application, with a filing date on or after March 16, 2013, will be examined under the first inventor to file provisions of the AIA.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	15305FTNK
	Application Number	
Title of Invention	RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH	

Authorization or Opt-Out of Authorization to Permit Access:

When this Application Data Sheet is properly signed and filed with the application, applicant has provided written authority to permit a participating foreign intellectual property (IP) office access to the instant application-as-filed (see paragraph A in subsection 1 below) and the European Patent Office (EPO) access to any search results from the instant application (see paragraph B in subsection 1 below).

Should applicant choose not to provide an authorization identified in subsection 1 below, applicant **must opt-out** of the authorization by checking the corresponding box A or B or both in subsection 2 below.

NOTE: This section of the Application Data Sheet is **ONLY** reviewed and processed with the **INITIAL** filing of an application. After the initial filing of an application, an Application Data Sheet cannot be used to provide or rescind authorization for access by a foreign IP office(s). Instead, Form PTO/SB/39 or PTO/SB/69 must be used as appropriate.

1. Authorization to Permit Access by a Foreign Intellectual Property Office(s)

A. Priority Document Exchange (PDX) - Unless box A in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the European Patent Office (EPO), the Japan Patent Office (JPO), the Korean Intellectual Property Office (KIPO), the State Intellectual Property Office of the People's Republic of China (SIPO), the World Intellectual Property Organization (WIPO), and any other foreign intellectual property office participating with the USPTO in a bilateral or multilateral priority document exchange agreement in which a foreign application claiming priority to the instant patent application is filed, access to: (1) the instant patent application-as-filed and its related bibliographic data, (2) any foreign or domestic application to which priority or benefit is claimed by the instant application and its related bibliographic data, and (3) the date of filing of this Authorization. See 37 CFR 1.14(h)(1).

B. Search Results from U.S. Application to EPO - Unless box B in subsection 2 (opt-out of authorization) is checked, the undersigned hereby **grants the USPTO authority** to provide the EPO access to the bibliographic data and search results from the instant patent application when a European patent application claiming priority to the instant patent application is filed. See 37 CFR 1.14(h)(2).

The applicant is reminded that the EPO's Rule 141(1) EPC (European Patent Convention) requires applicants to submit a copy of search results from the instant application without delay in a European patent application that claims priority to the instant application.

2. Opt-Out of Authorizations to Permit Access by a Foreign Intellectual Property Office(s)

☐ A. Applicant **DOES NOT** authorize the USPTO to permit a participating foreign IP office access to the instant application-as-filed. If this box is checked, the USPTO will not be providing a participating foreign IP office with any documents and information identified in subsection 1A above.

☐ B. Applicant **DOES NOT** authorize the USPTO to transmit to the EPO any search results from the instant patent application. If this box is checked, the USPTO will not be providing the EPO with search results from the instant application.

NOTE: Once the application has published or is otherwise publicly available, the USPTO may provide access to the application in accordance with 37 CFR 1.14.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	15305FTNK
	Application Number	
Title of Invention	RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH	

Applicant Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

Applicant 1 Remove

If the applicant is the inventor (or the remaining joint inventor or inventors under 37 CFR 1.45), this section should not be completed. The information to be provided in this section is the name and address of the legal representative who is the applicant under 37 CFR 1.43; or the name and address of the assignee, person to whom the inventor is under an obligation to assign the invention, or person who otherwise shows sufficient proprietary interest in the matter who is the applicant under 37 CFR 1.46. If the applicant is an applicant under 37 CFR 1.46 (assignee, person to whom the inventor is obligated to assign, or person who otherwise shows sufficient proprietary interest) together with one or more joint inventors, then the joint inventor or inventors who are also the applicant should be identified in this section.

Clear

◉ Assignee	○ Legal Representative under 35 U.S.C. 117	○ Joint Inventor
○ Person to whom the inventor is obligated to assign.	○ Person who shows sufficient proprietary interest	

If applicant is the legal representative, indicate the authority to file the patent application, the inventor is:

Name of the Deceased or Legally Incapacitated Inventor:

If the Applicant is an Organization check here. ☒

Organization Name	QOMPLX LLC		
Mailing Address Information For Applicant:			
Address 1	1900 Reston Metro Plaza		
Address 2	Ste. 600		
City	Reston	**State/Province**	VA
Country	US	Postal Code	20190
Phone Number		Fax Number	
Email Address			

Additional Applicant Data may be generated within this form by selecting the Add button. Add

Assignee Information including Non-Applicant Assignee Information:

Providing assignment information in this section does not substitute for compliance with any requirement of part 3 of Title 37 of CFR to have an assignment recorded by the Office.

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	15305FTNK
	Application Number	
Title of Invention	RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH	

Assignee 1

Complete this section if assignee information, including non-applicant assignee information, is desired to be included on the patent application publication. An assignee-applicant identified in the "Applicant Information" section will appear on the patent application publication as an applicant. For an assignee-applicant, complete this section only if identification as an assignee is also desired on the patent application publication.

Remove

If the Assignee or Non-Applicant Assignee is an Organization check here. ☐

Prefix	**Given Name**	Middle Name	**Family Name**	Suffix

Mailing Address Information For Assignee including Non-Applicant Assignee:

Address 1			
Address 2			
City		**State/Province**	
Country		Postal Code	
Phone Number		Fax Number	
Email Address			

Additional Assignee or Non-Applicant Assignee Data may be generated within this form by selecting the Add button. Add

Signature:

Remove

NOTE: This Application Data Sheet must be signed in accordance with 37 CFR 1.33(b). **However, if this Application Data Sheet is submitted with the INITIAL filing of the application and either box A or B is not checked in subsection 2 of the "Authorization or Opt-Out of Authorization to Permit Access" section, then this form must also be signed in accordance with 37 CFR 1.14(c).**

This Application Data Sheet **must** be signed by a patent practitioner if one or more of the applicants is a **juristic entity** (e.g., corporation or association). If the applicant is two or more joint inventors, this form must be signed by a patent practitioner, **all** joint inventors who are the applicant, or one or more joint inventor-applicants who have been given power of attorney (e.g., see USPTO Form PTO/AIA/81) on behalf of **all** joint inventor-applicants.

See 37 CFR 1.4(d) for the manner of making signatures and certifications.

Signature	/BRIAN R. GALVIN/			Date (YYYY-MM-DD)	2024-07-19
First Name	Brian	Last Name	Galvin	Registration Number	50966

Additional Signature may be generated within this form by selecting the Add button. Add

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	15305FTNK
	Application Number	
Title of Invention	RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH	

This collection of information is required by 37 CFR 1.76. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 23 minutes to complete, including gathering, preparing, and submitting the completed application data sheet form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1 The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these records.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3 A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent CooperationTreaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

PTO/AIA/01 (08-12)
Approved for use through 01/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office: U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

DECLARATION (37 CFR 1.63) FOR UTILITY OR DESIGN APPLICATION USING AN APPLICATION DATA SHEET (37 CFR 1.76)

Title of Invention	RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH

As the below named inventor, I hereby declare that:

This declaration is directed to:

- [■] The attached application, or
- [ ] United States application or PCT international application number _____ filed on _____

The above-identified application was made or authorized to be made by me.

I believe that I am the original inventor or an original joint inventor of a claimed invention in the application.

I hereby acknowledge that any willful false statement made in this declaration is punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both.

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

LEGAL NAME OF INVENTOR

Inventor: Jason Crabtree Date (Optional): 10 / 28 / 2015

Signature: Jason Crabtree

Note: An application data sheet (PTO/SB/14 or equivalent), including naming the entire inventive entity, must accompany this form or must have been previously filed. Use an additional PTO/AIA/01 form for each additional inventor.

This collection of information is required by 35 U.S.C. 115 and 37 CFR 1.63. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 1 minute to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

PTO/SB/08a (01-22)
Approved for use through 05/31/2024. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	
Filing Date	2024-07-21
First Named Inventor	Jason Crabtree
Art Unit	
Examiner Name	
Attorney Docket Number	15305FTNK

U.S.PATENTS Remove

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1					

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS Remove

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1					

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2]	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1							☐

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	1	Excerpts of raw documentation source located in the Apache Logging Flume GitHub "apache/logging-flume" repository tagged as Apache Flume Release 1.6.0, dated May 5, 2015, full release available at https://github.com/apache/logging-flume/tree/release-1.6.0, 200 pages	☐
	2	Excerpts of raw documentation source located in the Apache Airflow GitHub "apache/airflow" repository tagged as Apache Airflow Release v1.5.1, dated September 4, 2015, full release available at https://github.com/apache/airflow/tree/1.5.1, 75 pages	☐
	3	Excerpts of raw documentation source located in the Apache Apex core GitHub "apache/apex-core" repository tagged as Apache Apex Core Release v3.1.1, dated October 8, 2015, full release available at https://github.com/apache/apex-core/tree/v3.1.1, 158 pages	☐
	4	Excerpts of raw documentation source located in the Apache Beam GitHub "apache/beam" repository tagged as Apache Beam Release v1.2.0, dated October 5, 2015, full release available at https://github.com/apache/beam/tree/v1.2.0, 31 pages	☐
	5	Excerpts of raw documentation source located in the Apache Flink GitHub "apache/flink" repository tagged as Apache Flink Release 0.9.7, dated August 27, 2015, full release available at https://github.com/apache/flink/tree/release-0.9.1, 712 pages	☐
	6	Excerpts of raw documentation source located in the Apache Hadoop GiHub "apache/hadoop" repository tagged as Apache Hadoop Release 2.6.1, dated September 23, 2015, full release available at https://github.com/apache/hadoop/tree/release-2.6.1, 1640 pages	☐
	7	Excerpts of raw documentation source located in the Apache Ignite GitHub "apache/ignite" repository tagged as Apache Ignite Release 1.4.1, dated September 25, 2015, full release available at https://github.com/apache/ignite/tree/1.4.1, 103 pages	☐
	8	Excerpts of raw documentation source located in the Apache Kafka GitHub "apache/kafka" repository tagged as Apache Kafka Release 0.8.2.2, dated September 2, 2015, full release available at https://github.com/apache/kafka/tree/0.8.2.2, 13 pages	☐
	9	Excerpts of raw documentation source located in the Apache NiFi GitHub "apache/nifi" repository tagged as Apache NiFi Release nifi-0.3.0-RC1, dated September 14, 2015, full release available at https://github.com/apache/nifi/tree/nifi-0.3.0-RC1, 286 pages	☐
	10	Excerpts of raw documentation source located in the Apache Samza GitHub "apache/samza" repository tagged as Apache Samza Release 0.9.1-rc1, date June 23, 2015, full release available at https://github.com/apache/samza/tree/release-0.9.1-rc1, 221 pages	☐
	11	Excerpts of raw documentation source located in the Apache Spark GitHub "apache/spark" repository tagged as Apache Spark Release v.15.1-rc1, dated September 23, 2015, full release available at https://github.com/apache/spark/tree/v1.5.1, 2122 pages	☐

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	12	Excerpts of raw documentation source located in the Apache Storm GitHub "apache/storm" repository tagged as Apache Storm Release v0.10.0-beta1, dated June 19, 2015, full release available at https://github.com/apache/storm/tree/v0.10.0-beta1, 589 pages	☐

If you wish to add additional non-patent literature document citation information please click the Add button Add

EXAMINER SIGNATURE

Examiner Signature		Date Considered	

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

CERTIFICATION STATEMENT

Please see 37 CFR 1.97(e) to make the appropriate selection(s):

☐ That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

☐ See attached certification statement.

☒ A certification statement is not submitted herewith.

FEE

☐ The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

COPIES

☐ An identification of an earlier application pursuant to 37 CFR 1.98 (d) (1) is attached.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/BRIAN R. GALVIN/	Date (YYYY-MM-DD)	2024-07-19
Name/Print	Brian R. Galvin	Registration Number	50966

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these record s.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
18 June 2015 (18.06.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/089463 A1

(51) International Patent Classification:
H04L 12/28 (2006.01)

(21) International Application Number:
PCT/US2014/070131

(22) International Filing Date:
12 December 2014 (12.12.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/916,061 13 December 2013 (13.12.2013) US
14/567,684 11 December 2014 (11.12.2014) US

(71) Applicant: **BLOOMREACH INC.** [US/US]; 82 Pioneer Way, Mountain View, CA 94041 (US).

(72) Inventors: **GUPTA, Prateek**; 82 Pioneer Way, Mountain View, CA 94041 (US). **PATNALA, Praveen, K.**; 82 Pioneer Way, Mountain View, CA 94041 (US). **AGGARWAL, Amit**; 82 Pioneer Way, Mountain View, CA 94041 (US). **GARG, Ashutosh**; 82 Pioneer Way, Mountain View, CA 94041 (US). **BRAR, Gurashish, Singh**; 82 Pioneer Way, Mountain View, CA 94041 (US). **SHARMA, Kannan, Nitin**; 82 Pioneer Way, Mountain View, CA 94041 (US). **SARKAR, Manojit**; 82 Pioneer Way, Mountain View, CA 94041 (US). **WANG, Shao-chuan**; 82 Pioneer Way, Mountain View, CA 94041 (US).

(74) Agent: **SCHALLOP, Michael, J.**; Van Pelt, YI & James LLP, 10050 N. Foothill Blvd., Suite 200, Cupertino, CA 95014 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) Title: DISTRIBUTED AND FAST DATA STORAGE LAYER FOR LARGE SCALE WEB DATA SERVICES

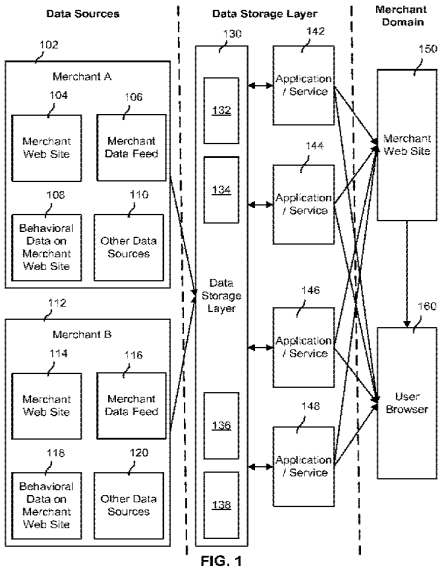


FIG. 1

(57) Abstract: Techniques for providing a distributed and fast data storage layer for large scale web data services are disclosed. In some embodiments, a distributed data storage layer for web data services includes storing web related data for a plurality of merchants in the distributed data storage layer; and copying on demand a data store and a search index for web related data for a first merchant to perform a job for updating the web related data for the first merchant.

WO 2015/089463 A1

Published:

— *with international search report (Art. 21(3))*

DISTRIBUTED AND FAST DATA STORAGE LAYER FOR LARGE SCALE WEB DATA SERVICES

CROSS REFERENCE TO OTHER APPLICATIONS

[0001] This application claims priority to U.S. Provisional Patent Application No. 61/916,061, entitled DISTRIBUTED AND FAST DATA STORAGE LAYER FOR LARGE SCALE WEB DATA SERVICES filed December 13, 2013, which is incorporated herein by reference for all purposes.

BACKGROUND OF THE INVENTION

[0002] Web services can be used to provide communications between electronic/computing devices over a network, such as the Internet. A web site is an example of a type of web service. A web site is typically a set of related web pages that can be served from a web domain. A web site can be hosted on a web server. A publicly accessible web site can generally be accessed via a network, such as the Internet. The publicly accessible collection of web sites is generally referred to as the World Wide Web.

[0003] Internet-based web services can be delivered through web sites on the World Wide Web (WWW). Web pages are often formatted using HyperText Markup Language (HTML), eXtensible HTML (XHTML), or using another language that can be processed by a web browser that is typically executed on a user's client device, such as a computer, tablet, phablet, smart phone, smart watch, smart television, or other (client) device. A web site can be hosted on a web server that is typically accessible via a network, such as the Internet, through a web address, which is generally known as a Uniform Resource Indicator (URI) or a Uniform Resource Locator (URL).

[0004] Search engines can be used to facilitate searching of web services, such as to identify relevant web sites for particular content and/or services on the World Wide Web. Search engines (e.g., web-based search engines provided by various vendors, including, for example, Google, Microsoft Bing, and Yahoo) provide for searches of online information that includes searchable content (e.g., digitally stored electronic data), such as searchable content available via the World Wide Web. As input, a search engine typically receives a search query (e.g., query input including one or more terms, such as keywords, by a user of

the search engine). The search engine performs the search for the search query and outputs results that are typically presented in a ranked list, often referred to as search results or hits (e.g., links or Uniform Resource Locators (URLs) for one or more web pages and/or web sites). The search results can include web pages, images, audio, video, database results, directory results, information, and other types of data.

[0005] Search engines typically provide paid search results (e.g., the first set of results in the main listing and/or results often presented in a separate listing on the right side of the output screen). For example, advertisers may pay for placement in such paid search results based on keywords (e.g., keywords in search queries). Search engines also typically provide organic search results, also referred to as natural search results. Organic search results are based on various algorithms employed by different search engines that attempt to provide relevant search results based on a received search query.

[0006] For improved Internet marketing, search engine optimization (SEO) has developed as a form of industry/technical consulting (often referred to as search engine optimizers) provided to web site operators (e.g., vendors of products/services with web sites and/or e-commerce vendors of products/services) for improving the volume or quality of traffic to a web site from a search engine via organic search results (e.g., to improve the web site's web presence as a paid service engagement or pursuant to a marketing campaign). Generally, the higher a web site appears in the organic search results list, the more users it will receive from the search engine. SEO can target different kinds of searches, including image searches, local searches, and industry specific, vertical search engines to improve the web site's web presence. For example, SEO often considers how search engines work and what people search for to recommend web site related changes to optimize a web site (e.g., which primarily involves editing its content and HyperText Markup Language (HTML) coding to both increase its relevance to specific keywords and to remove barriers to the indexing activities of search engines).

BRIEF DESCRIPTION OF THE DRAWINGS

[0007] Various embodiments of the invention are disclosed in the following detailed description and the accompanying drawings.

[0008] **Figure 1** is a functional block diagram illustrating an overview of an architecture for providing a distributed and fast data storage layer for large scale web data services in accordance with some embodiments.

[0009] **Figure 2** is a functional block diagram illustrating an architecture for ingesting data from merchant data sources at the data storage layer in accordance with some embodiments.

[0010] **Figure 3** is a functional block diagram illustrating a distributed and fast data storage layer for large scale web data services including an example application that utilizes the data storage layer in accordance with some embodiments.

[0011] **Figure 4** is a functional block diagram illustrating a topology of a distributed system for implementing the data storage layer in accordance with some embodiments.

[0012] **Figure 5** is a functional block diagram illustrating a data storage layer in communication with an application in accordance with some embodiments.

[0013] **Figure 6** is a functional block diagram illustrating a data storage layer in communication with a web search related application in accordance with some embodiments.

[0014] **Figure 7** is a flow diagram illustrating the process for providing a distributed and fast data storage layer for large scale web data services in accordance with some embodiments.

[0015] **Figure 8** is another flow diagram illustrating the process for providing a distributed and fast data storage layer for large scale web data services in accordance with some embodiments.

[0016] **Figure 9** is another flow diagram illustrating the process for providing a distributed and fast data storage layer for large scale web data services in accordance with some embodiments.

DETAILED DESCRIPTION

[0017] The invention can be implemented in numerous ways, including as a process; an apparatus; a system; a composition of matter; a computer program product embodied on a computer readable storage medium; and/or a processor, such as a processor configured to

execute instructions stored on and/or provided by a memory coupled to the processor. In this specification, these implementations, or any other form that the invention may take, may be referred to as techniques. In general, the order of the steps of disclosed processes may be altered within the scope of the invention. Unless stated otherwise, a component such as a processor or a memory described as being configured to perform a task may be implemented as a general component that is temporarily configured to perform the task at a given time or a specific component that is manufactured to perform the task. As used herein, the term 'processor' refers to one or more devices, circuits, and/or processing cores configured to process data, such as computer program instructions.

[0018] A detailed description of one or more embodiments of the invention is provided below along with accompanying figures that illustrate the principles of the invention. The invention is described in connection with such embodiments, but the invention is not limited to any embodiment. The scope of the invention is limited only by the claims and the invention encompasses numerous alternatives, modifications and equivalents. Numerous specific details are set forth in the following description in order to provide a thorough understanding of the invention. These details are provided for the purpose of example and the invention may be practiced according to the claims without some or all of these specific details. For the purpose of clarity, technical material that is known in the technical fields related to the invention has not been described in detail so that the invention is not unnecessarily obscured.

[0019] Many vendors provide big data applications that can use content from a variety of data sources, such as a merchant domain, web-wide data, and/or data from other sources. This generally requires storing information so that it is easily accessible by several different applications and possibly several different customers. However, such presents particular technical challenges to provide such data in a scalable, reliable, and fast response time manner.

[0020] What is needed are new techniques for providing a data storage layer that can support large scale data applications that can use content from a variety of data sources, such as a merchant domain, web-wide data, and/or data from other sources (e.g., to optimize customer acquisition and experiences). Thus, there is a need for a distributed and fast data storage layer for large scale web data services.

[0021] Accordingly, techniques for providing a distributed and fast data storage layer for large scale web data services are disclosed. In some embodiments, a distributed and fast data storage layer for large scale web data services is disclosed for implementing techniques for storing, retrieving, and searching information about web pages, such as further described below. In some embodiments, a system for a distributed data storage layer for web data services includes the distributed data storage layer for storing web related data for a plurality of merchants; and a copy on demand dynamic cluster including a data store and a search index for storing web related data for a first merchant to perform a job for updating the web related data for the first merchant using a processor.

[0022] For example, the copy on demand dynamic cluster can include the data store and the search index for storing the web related data for the first merchant corresponds to a snapshot of the data store and the index stored in the dynamic cluster of the distributed data storage layer. The web related data for the plurality of merchants can be stored in a back-end cluster of the distributed data storage layer. In some cases, the web related data can include crawled web site data, feed data, and user behavior data.

[0023] In some embodiments, the system for a distributed data storage layer for web data services further includes an application that is in communication with the distributed data storage layer and a web site associated with the first merchant (e.g., the application can also be in communication with a user browser).

[0024] In some embodiments, the system for a distributed data storage layer for web data services further includes a merge product information engine for merging crawled web site data and feed data for each of the plurality of merchants (e.g., the merge product information engine can maintain data consistency for the crawled web site data and feed data for each of the plurality of merchants).

[0025] In some embodiments, the system for a distributed data storage layer for web data services further includes a product attribute computation engine for reading crawl related product attribute data and feed related product data stored in a data store of the distributed data storage layer and for computing crawl related product attribute data and feed related product data for each of the plurality of merchants, in which the merged crawl related product attribute data and feed related product data for each of the plurality of merchants is stored in the data store of the distributed data storage layer.

[0026] In some embodiments, the system for a distributed data storage layer for web data services further includes a back-end job manager for delegating jobs for processing by a back-end cluster or a copy on demand dynamic cluster.

[0027] In some embodiments, the system for a distributed data storage layer for web data services further includes a resource aware adaptive capacity/throttling engine for determining resource requirements for performing back-end job processing based on one or more rules.

[0028] In some embodiments, the system for a distributed data storage layer for web data services further includes a resource manager for optimizing a cluster size for a back-end cluster and/or the copy on demand dynamic cluster for performing back-end job processing.

[0029] In some embodiments, the system for a distributed data storage layer for web data services further includes a front-end cluster for storing a subset of the web related data for each of the plurality of merchants in the distributed data storage layer.

[0030] In some embodiments, a method for a distributed data storage layer for web data services includes storing web related data for a plurality of merchants in the distributed data storage layer; and copying on demand a data store and a search index for web related data for a first merchant to perform a job for updating the web related data for the first merchant.

[0031] For example, various techniques are disclosed for storing, retrieving, and searching information about web data (e.g., web pages) for a merchant's web site in accordance with some embodiments. A merchant as used herein generally refers to a third party entity that uses a web site (e.g., on the World Wide Web (WWW)) to engage with customers and/or to buy and sell products and/or services. A data feed as used herein generally refers to a data object (e.g., a file) that includes a list of products/services that use groupings of attributes that define or identify each one of a merchant's products/services in a unique way. Response time for a service as used herein generally refers to a time interval between a time that a request is received and a response is sent. Referrer URL generally refers to the URL of the web page that preceded the page in question.

[0032] In some embodiments, a distributed data storage layer is disclosed that provides a new and improved way of storing and searching web documents for multiple third party merchant domains from multiple data sources including World Wide Web information.

For example, the distributed data storage layer disclosed herein can provide a consistent (e.g., facilitating data consistency for maintained merchant data using various data consistency processing techniques, such as further described herein), accurate, and real-time (e.g., near real-time, such as further described herein) view of third party merchant content that can be derived from multiple data sources, such as web pages on a web site, data from a merchant data feed, and/or data from user behavior on a merchant's web site. In an example implementation, the distributed data storage layer can also provide fast and guaranteed performance in terms of "response time." In this example implementation, the data is also available in a consistent and fast manner across geographic locations, such as US East, US West, EU West, and/or other geographic locations.

[0033] These and various other techniques for a distributed and fast data storage layer for large scale web data services are further described below.

[0034] **Figure 1** is a functional block diagram illustrating an overview of an architecture for providing a distributed and fast data storage layer for large scale web data services in accordance with some embodiments. Figure 1 provides an exemplary architecture that can be implemented using various computing hardware and performing various processes as will be further described below.

[0035] Referring to Figure 1, data sources include various merchant data, such as merchant data for a Merchant A shown as 102 and a Merchant B shown as 112. Merchant A data can include a merchant web site 104, a merchant data feed 106, behavioral data on the merchant web site 108, and various other data sources 110 (e.g., various other web data sources including search logs, social media data sources, and/or other data sources). Similarly, Merchant B data can include a merchant web site 114, a merchant data feed 116, behavioral data on the merchant web site 118, and various other data sources 120.

[0036] As also shown in Figure 1, a data storage layer includes a data storage layer 130 and a plurality of applications/services. For example, the data storage layer is in communication with each of the merchant data sources, including data sources for Merchant A and data sources for Merchant B, to receive merchant web site data, merchant data feeds, behavior data on the merchant web site, and/or other data sources for each of the merchants. Data storage layer 130 is also in communication with an application/service 142, an application/service 144, an application/service 146, and an application/service 148. Example applications/services are further described below.

[0037] A merchant domain includes a merchant web site 150 and a user browser 160 as also shown in Figure 1. For example, each of the applications/services can be in communication with merchant web site 150 and user browser 160 as shown. Also, merchant web site 150 is in communication with user browser 160 as also shown.

[0038] In some embodiments, the above-described data sources (e.g., data sources for Merchant A shown as 102 and data sources for Merchant B shown as 112) can be processed using various techniques as described below. For example, a web site content fetching process can be performed (e.g., implemented using a web site content fetching engine) to fetch web pages from a merchant web site (e.g., to periodically fetch web pages from merchant web site 104 and merchant web site 114). The content (e.g., HTML content, dynamic content such as Java Script content, and/or other content) from the fetched web pages from the merchant web site can then be parsed to extract attributes for a product, such as a title, a heading, a description, a price, a color(s), and/or a size(s). As another example, a merchant data feed retrieving process can be performed (e.g., implemented using a merchant data feed retrieving engine) to retrieve merchant feeds and parse product attributes, such as title, heading, description, and/or price. As yet another example, a behavior data collection process can be performed (e.g., implemented using a behavioral data collection engine) using pixel data on a merchant's web site. Using this approach, when a user visits a web page of a given merchant's web site, then dynamic content (e.g., Java Script code or other dynamic code/content) on the web page generates a pixel request to the data storage layer that includes, for example, a machine IP (e.g., the IP address associated with the user's device), a user ID (optional), a referrer URL, a web page URL, and/or other information (optional). In this example, raw pixel data can then be provided from each of the merchants (e.g., shown as behavioral data on merchant web site 108 and behavioral data on merchant web site 118). Each of the data sources is described in more detail below in accordance with various embodiments.

[0039] As similarly described above, a continuous process can be performed to fetch pages from a merchant website (e.g., implemented using a web site content parsing engine, which can be implemented as a fetching engine (fetcher) shown as 132 and a parsing engine shown as 134). The fetcher (132) (e.g., in some implementations, a web site content parsing engine can include a fetching engine to fetch web pages and a parsing engine to parse the fetched web pages) receives as input a list of URLs on a merchant web site and fetches the content of the web page (e.g., HTML and/or other content on each of the fetched web pages). For example, the HTML content of a fetched web page can then be parsed (e.g., by parsing

engine 134) using parsing rules configured for merchant web pages (e.g., manually configured for a particular merchant's web pages or using a default configuration for parsing typical merchant web pages). The parser (134) can parse each page to extract attributes, such as title, heading, description, and price. In some cases, the information on the web page may not be fresh or accurate. For example, the product may go "out of stock," but the web page may display the product to be in-stock. The user may try to purchase the product on the web page but will receive an error saying that the product is "out of stock." Solutions provided by the data storage layer for addressing these and other technical problems that can arise with web related services are further described below with respect to various embodiments.

[0040] As similarly described above, a process can be performed to retrieve merchant data feeds for each of the merchants (e.g., implemented using a merchant data feed retrieving engine shown as 136). In an example implementation, the merchant can either provide a location to fetch the merchant data feed or upload the merchant data feed to a location specified for the data storage layer provider. For example, the merchant data feed can be provided as a file (e.g., in a format that can be parsed, such as in an XML or other file format that provides a (subset) view of, for example, a merchant's product/service catalog). In some embodiments, the merchant data feed can be provided on a daily or hourly basis (e.g., or at some other periodic/time interval and/or based on demand). For example, the merchant data feed can be parsed using parsing rules configured for a given merchant or a set of merchants (e.g., based on types/groups of merchants) to extract attributes, such as title, heading, and description. In some cases, the merchant data feed may not be complete, in which, for example, the merchant data feed may only contain a subset of products sold on the merchant web site (e.g., the merchant data may only contain a small subset of the products sold on the merchant web site). Solutions provided by the data storage layer for addressing technical problems that can arise with merging subsets of merchant feed data with other merchant data and other technical problems that can arise with web related services are further described below with respect to various embodiments.

[0041] In an example implementation, a process to combine product information from parsed merchant web pages and a retrieved merchant data feed (e.g., such as using various processes as described in detail above) is performed, such as using a product data merging engine shown as 138. It is important to note that for a given product, the information from parsed merchant web pages and a retrieved merchant data feed may be incomplete or inaccurate. For example, "out of stock" information may not be fresh or accurate on merchant web pages. The merchant data feed may contain missing information about

products, such as the description for a given product may be missing or incomplete. In some cases, the information from these data sources (e.g., the merchant web site and the merchant data feed) may not match. For example, the price for a given product may be updated in the merchant data feed on an hourly basis, but the web page may display outdated price information. In some implementations, the information from these data sources (e.g., the merchant web site and the merchant data feed) is combined into a single entry for each product using a combination of machine learning algorithms (e.g., association rule learning, support vector machines, Bayesian networks, clustering, decision trees, linear regression, and/or various other machine learning techniques and custom rules can be implemented to facilitate improved merging of such product information from these two different data sources for ensuring data consistency at the data storage layer for the merchant's product/service related data) and custom rules specified for the merchant, such as further described below. For example, in the case of mismatching price information, a rule may be configured to always prefer the price from the merchant data feed (if available) over the price retrieved from parsing the merchant web pages.

[0042] Another data source of information is behavioral user data on a merchant web site. In some cases, a user may visit a merchant web page. For example, Java Script code (e.g., or other executable code/content) on the web page can make a pixel request to the data storage layer (e.g., to a server/service of the data storage layer provider) that includes, for example, a machine IP (e.g., the IP address associated with the user's device), a user ID (optional), a referrer URL, a web page URL, and/or other information (optional). The list of web page URLs can be combined periodically (e.g., each day or at some other periodic interval) and provided as input to the above-described fetching and parsing process to extract information from web pages of a merchant's web site. In an example implementation, for each user, information can be stored to identify products visited and purchased by the user. Similarly, information can be stored to aggregate all user visits and products purchased on a merchant web site. In one embodiment, pixel data can be aggregated across different web sites/merchants as further described herein.

[0043] In one embodiment, the above-described processes implemented by the data storage layer are performed per each merchant. For example, each merchant's data is processed independently using the above-described processes for processing each merchant's respective data sources and for performing data merging from distinct data sources (e.g., merging product information from the merchant web site and the merchant data feed) for ensuring data consistency and accuracy of the merchant data stored at the data storage layer.

[0044] In some implementations, a product schema is used for storing information in a data store of the data storage layer about products available for sale on a merchant web site. As will be apparent to one of ordinary skill in the art, various forms of product schemas can be used for storing information in a data store about products available for sale on a merchant web site.

[0045] An example product schema per merchant is shown below.

```
-- Only supported types are:
--    varchar, boolean, int, bigint, float, double, timestamp, map

-- Product Model:
-- --------------
-- A product is an item that can be purchased or "converted".
--
-- row-key: mpid
--
create table br${merchant_id}_product (
  m_pid              varchar,                  --  PartitionKeyType: merchant product id

  -- dependent (NON PRIMARY-KEY) attributes
  -- ------------------------------------------------------------------------------------------------------------

  -- relation to other product groups
  pg_type            varchar,                  --  my group type
  pg_o_refs          map<varchar, varchar>,      --  other groups do I refer to? e.g. { 'sku' : ['sku123'] }
  pg_i_refs          map<varchar, varchar>,      --  other groups that refer to me? e.g. { 'suite': 'prod1234' }

  -- life-cycle attributes
  l_first_live_ts    timestamp,                --  when was it first seen as a 'live' product on the site
  l_last_live_ts     timestamp,                --  when was it last seen as a 'live' product on the site
  l_launch_ts        timestamp,                --  merchant supplied
  l_is_live          boolean,                  --  whether a product is 'live' or not
  is_blacklist       boolean,                  --  whether a product is 'blacklisted' or not

  -- display attributes
  d_title            varchar,                  --  display name, the more descriptive the better
  d_description      varchar,                  --  paragraph(s) describing its most relevant attributes
  d_brand            varchar,                  --  brand name of the manufacturer
  d_model_name       varchar,                  --  model_name for this product
  d_gender           varchar,                  --  target gender:  male, female, unisex
  d_age_group        varchar,                  --  target age: baby, kids, junior, adult, senior
  d_mature           boolean,                  --  mature content (non family-safe)

  d_conditions       map<varchar, varchar>,      --  condition (multi-valued): new, used, refurbished
  d_colors           map<varchar, varchar>,      --  colors (non-standard names)
  d_color_group      map<varchar, varchar>,      --  color group (standard names, no ordering)
  d_sizes            map<varchar, varchar>,      --  size (non-standard names)
  d_pattern          map<varchar, varchar>,      --  pattern (e.g. polka dot)
  d_material         map<varchar, varchar>,      --  material (e.g. nylon)
  d_flags            map<varchar, varchar>,      --  static display-flags from merchant, e.g. "more colors"
  d_urls             map<varchar, varchar>,      --  product urls
  d_th_img_urls      map<varchar, varchar>,      --  thumb image urls (for listing pages)
  d_la_img_urls      map<varchar, varchar>,      --  large image urls (for product pages, listing quick view)
  d_zo_img_urls      map<varchar, varchar>,      --  zoom image urls (for product page zoom)
  d_sw_img_urls      map<varchar, varchar>,      --  swatch image urls (swatches of fabric)
  d_keywords         map<varchar, varchar>,      --  for keyword searches
```

```
  d_stores          map<varchar, varchar>,      -- for store information

  -- collection attributes that hold multiple feed fields
  d_others          map<varchar, varchar>,      -- use sparingly, custom stuff: e.g. mobile NM merchant_api_json

  -- inventory & price attributes
  i_price           float,                      -- regular price
  i_price_rl        float,                      -- regular price range (low, high) for group = suite etc.
  i_price_rh        float,
  i_price_tags      map<varchar, varchar>,

  i_sale_price      float,                      -- sale price
  i_sale_price_rl   float,                      -- sale price range (low, high) for group = suite etc.
  i_sale_price_rh   float,
  i_sale_price_tags map<varchar, varchar>,
  -- i_sale_dates      map<timestamp, timestamp>,  -- [start_timestamp, end_timestamp)

  i_status          varchar,                    -- in stock, available for order, out of stock, pre order
  i_level           int,                        -- number of units available
  i_have_avail      boolean,                    -- is the item sellable?
  i_last_avail_ts   timestamp,                  -- when was it last available?
  i_buyable         boolean,                    -- is the item buyable?
  i_have_local_avail map<varchar, varchar>,     -- bnm_store.m_bnid => { latlon: ..., level: InventoryLevelType, ... }

  -- category attributes
  cat_goog          varchar,                    -- google taxonomy, e.g. Apparel & Accessories > Clothing > Jeans
  cat_nodes         map<varchar, varchar>,      -- leaf category nodes { category-id : {category-name: ... } }
  cat_crumbs        map<varchar, varchar>,      -- merge page data breadcrumbs, e.g. { 'Clearance > Dresses' : 'cat1 > cat2'
}

  -- blobs
  b_paa             blob,                       -- PAAnnotatedPage in compressed byte array form
  b_page            blob,                       -- PageData in compressed byte array form
  b_variant         blob,                       -- Variant for the product as served by mobile frontend apis

  -- canon urls
  c_canon_url       varchar,
  c_canon_key       varchar,
  c_url_flags       varchar,

  -- status flags
  s_deleted         int,                        -- 0 false, 1 true
  s_status_tags     list<varchar>,

  -- page_data:

  primary key (m_pid)
);
-------------------------------------------------------------------------------------------------------------------
-- Generic Key-Value Store:  stores miscellaneous items such as category trees, static assets, etc.
--
create table m${merchant_id}_mob_misc (
  m_rk              varchar,                    -- generic row key
  ts                timestamp,                  -- timestamp for versioning
  mi_ns             varchar,                    -- namespace of data
  mi_nk             varchar,                    -- namespace key

  cat_tree          varchar,

  nm_assets         varchar,
  nm_midday         varchar,
```

```
  scrape_cat        varchar,
  scrape_search     varchar,
  scrape_update     boolean,

  others            map<varchar, varchar>,

  primary key (m_rk, ts, mi_ns, mi_nk)
) with clustering order by (ts desc);
```

[0046] In some implementations, a non-product schema is used for storing information in a data store about non-product information on a merchant web site (e.g., collected by crawling the merchant web site, such as using the above-described fetching engine). As will be apparent to one of ordinary skill in the art, various forms of non-product schemas can be used for storing information in a data store about non-product information on a merchant web site.

[0047] An example schema for storing non-product information per merchant is shown below.

```
-- Web Page View:
-- --------------
-- A web-page is a materialized-view that is accessible via a URL. A web-page may be composed of
-- many entities (e.g., some of which may correspond to other models in the system, while others being represented as
-- a bag of words or not represented).
--
-- key: mck
--
create table v${merchant_id}_web_page (
  m_ck              varchar,                -- PartitionKeyType: merchant canon-key
  ts                timestamp,              -- versioning timestamp
  src               varchar,                -- ProjectSrc: source of the data
  ptype             varchar,                -- PageType: page-type

  -- Instead of duplicating the various column schemas from the various models that contributed to this web-page,
  -- that data can be viewed as uninterpreted key-value pairs and delegate to the model logic for recreating
  -- structured data.

  -- m_product stores Product proto, which includes PageData & PAAnnotation
  -- m_product_quality stores ProductQuality proto

  b_page            blob,               -- PageData in compressed byte array form
  c_canon_url       varchar,
  c_url_flags       varchar,
  s_deleted         int,                -- 0 false, 1 true
  s_status_tags     list<varchar>,      -- ArrayList serialized to json

  primary key (m_ck, ts, src, ptype)
) with clustering order by (ts desc);
```

[0048] In some implementations, a schema for personalization is used for storing information in a data store about users (e.g., for user personalization). As will be apparent to one of ordinary skill in the art, various forms of schemas for personalization can be used for storing information in a data store about users (e.g., for user personalization).

[0049] An example schema for personalization is shown below.

```
//------------------------------------------------------------------------
create table m{{accid}}_shared_items (
  id      varchar,                  // item id
  ts      timestamp,                  // timestamp - only for versioning, HEAD = '9999-01-01+0000'

  // current owner
  coid    varchar,                  // owner id
  copp    varchar,                  // owner passphrase
  cots    timestamp,                  // ownership timestamp

  // last owner (to avoid race during ownership transfer)
  loid    varchar,                  // owner id
  lots    timestamp,                  // ownership timestamp
  lopp    varchar,                  // owner passphrase

  title   varchar,                  // visible name of the item
  cts     timestamp,                  // creation timestamp
  inv     boolean,                  // is the item no longer valid? (unset = valid)

  // attributes
  attrs   map<varchar, varchar>,

  //
  // Children of this item (e.g. products belonging to a favorites list).
  // e.g. products are children of favorites:
  // i.e. k: p=prod1234  v: { price: 30, title: studded shoe, brand: prada }
  //
  // NOTE: collections do not support writetime, so if we need to know when updates happened,
  // another column or key inside the collection can be provided.
  //
  parts   map<varchar, varchar>,

  primary key (id, ts, coid, copp)
) with clustering order by (ts desc);

//------------------------------------------------------------------------
// objects that refer to a user.
//   user <--- item (via action)
//   user <--- user (via handle + item)
create table r{{accid}}_user_ref (
  id      varchar,
  refid   varchar,
  ts      timestamp,
  hl      varchar,

  attrs   map<varchar, varchar>,

  primary key (id, ts, refid)
) with clustering order by (ts desc, refid asc);
// These tables store user profile preferences.
create table m{{accid}}_user_profile_1 (
  id varchar,  // user id or cohort id ("feed" is special user)

  p_ns varchar,  // primary key namespace (reserved)
```

```
  p_nk varchar, // primary key namekey (reserved)

  last_update timestamp,
  user_profile_response blob, // UserProfileResponse protobuf as byte array
  num_pref int,
  total_visits int,

  primary key (id, p_ns, p_nk)
);

// This table store user facts
create table m{{accid}}_user_facts (
  id varchar,  // user id or cohort id ("feed" is special user)

  p_ns varchar,  // primary key namespace (reserved)
  p_nk varchar, // primary key namekey (reserved)

  last_update timestamp, // timestamp of latest user fact
  user_facts blob, // UserProfileResponse protobuf as byte array
  primary key (id, p_ns, p_nk)
);

// Table for storing product recommendations
create table m{{accid}}_product_recommendations (
  id varchar,  // product id

  p_ns varchar,  // primary key namespace (reserved)
  p_nk varchar, // primary key namekey (reserved)

  last_update timestamp, // timestamp of latest recommendation
  recommendations blob, // ProductRecommendations protobuf as byte array
  primary key (id, p_ns, p_nk)
);
```

[0050] In one embodiment, the above-described processes implemented by the data storage layer are performed across merchants, such as for merchants that sell overlapping products/services. For example, assuming that Merchant A and Merchant B both sell overlapping apparel or other products, behavioral data for both of these merchants can be processed to facilitate common behavior signals that may be relevant to products/services available from each of these merchants (e.g., overlapping products/services offered from each of the merchants can be automatically identified, such as if two different vendors both offer for sale certain Apple or Google consumer electronic/computing products, then behavior signals related to any of such overlapping products/services can be processed across these merchants to facilitate increased behavior data signals for such overlapping products/services). However, in some cases, merchants may desire to not allow for such cross-merchant data processing at the data storage layer, such as for user privacy or other data protection concerns or contractual restrictions (e.g., a given merchant may have user privacy policies and/or other contractual restrictions that prohibit certain cross-merchant data processing at the data storage layer, in which case any such data for that given merchant can

be processed using the above-described per merchant data processing techniques and not the across merchant data processing techniques at the data storage layer).

[0051] In some implementations, a combined product schema is used for storing information in a data store about products (e.g., for all or a group of merchants). As will be apparent to one of ordinary skill in the art, various forms of combined product schemas can be used for storing information in a data store about products for all or a group of merchants.

[0052] An example combined product schema (e.g., for all or a group of merchants, such as can be used to facilitate the across merchant data processing techniques) is shown below.

```
-- Only supported types are:
--    varchar, boolean, int, bigint, float, double, timestamp, map

-- Product Model:
-- --------------
-- A product is an item that can be purchased or "converted".
--
-- row-key: merchant_id + mpid
--
create table br_product (
  merchant_id       varchar,                -- PartitionKeyType: merchant id
  m_pid             varchar,                -- PartitionKeyType: merchant product id

  -- dependent (NON PRIMARY-KEY) attributes
  -- ------------------------------------------------------------------------------------------------------------

  -- relation to other product groups
  pg_type           varchar,                -- my group type
  pg_o_refs         map<varchar, varchar>,  -- other groups do I refer to? e.g. { 'sku' : ['sku123'] }
  pg_i_refs         map<varchar, varchar>,  -- other groups that refer to me? e.g. { 'suite': 'prod1234' }

  -- life-cycle attributes
  l_first_live_ts   timestamp,              -- when was it first seen as a 'live' product on the site
  l_last_live_ts    timestamp,              -- when was it last seen as a 'live' product on the site
  l_launch_ts       timestamp,              -- merchant supplied
  l_is_live         boolean,                -- whether a product is 'live' or not
  is_blacklist      boolean,                -- whether a product is 'blacklisted' or not

  -- display attributes
  d_title           varchar,                -- display name, the more descriptive the better
  d_description     varchar,                -- paragraph(s) describing its most relevant attributes
  d_brand           varchar,                -- brand name of the manufacturer
  d_model_name      varchar,                -- model_name for this product
  d_gender          varchar,                -- target gender: male, female, unisex
  d_age_group       varchar,                -- target age: baby, kids, junior, adult, senior
  d_mature          boolean,                -- mature content (non family-safe)

  d_conditions      map<varchar, varchar>,  -- condition (multi-valued): new, used, refurbished
  d_colors          map<varchar, varchar>,  -- colors (non-standard names)
  d_color_group     map<varchar, varchar>,  -- color group (standard names, no ordering)
  d_sizes           map<varchar, varchar>,  -- size (non-standard names)
  d_pattern         map<varchar, varchar>,  -- pattern (e.g. polka dot)
d_material          map<varchar, varchar>,  -- material (e.g. nylon)
```

```
  d_flags           map<varchar, varchar>,      -- static display-flags from merchant, e.g. "more colors"
  d_urls            map<varchar, varchar>,      -- product urls
  d_th_img_urls     map<varchar, varchar>,      -- thumb image urls (for listing pages)
  d_la_img_urls     map<varchar, varchar>,      -- large image urls (for product pages, listing quick view)
  d_zo_img_urls     map<varchar, varchar>,      -- zoom image urls (for product page zoom)
  d_sw_img_urls     map<varchar, varchar>,      -- swatch image urls (swatches of fabric)
  d_keywords        map<varchar, varchar>,      -- for keyword searches
  d_stores          map<varchar, varchar>,      -- for store information

  -- collection attributes that hold multiple feed fields
  d_others          map<varchar, varchar>,      -- use sparingly, custom stuff: e.g. mobile NM merchant_api_json

  -- inventory & price attributes
  i_price           float,                      -- regular price
  i_price_rl        float,                      -- regular price range (low, high) for group = suite etc.
  i_price_rh        float,
  i_price_tags      map<varchar, varchar>,

  i_sale_price      float,                      -- sale price
  i_sale_price_rl   float,                      -- sale price range (low, high) for group = suite etc.
  i_sale_price_rh   float,
  i_sale_price_tags map<varchar, varchar>,
  -- i_sale_dates      map<timestamp, timestamp>,  -- [start_timestamp, end_timestamp)

  i_status          varchar,                    -- in stock, available for order, out of stock, pre order
  i_level           int,                        -- number of units available
  i_have_avail      boolean,                    -- is the item sellable?
  i_last_avail_ts   timestamp,                  -- when was it last available?
  i_buyable         boolean,                    -- is the item buyable?
  i_have_local_avail map<varchar, varchar>,     -- bnm_store.m_bnid => { latlon: ..., level: InventoryLevelType, ... }

  -- category attributes
  cat_goog          varchar,                    -- google taxonomy, e.g. Apparel & Accessories > Clothing > Jeans
  cat_nodes         map<varchar, varchar>,      -- leaf category nodes { category-id : {category-name: ... } }
  cat_crumbs        map<varchar, varchar>,      -- merge page data breadcrumbs, e.g. { 'Clearance > Dresses' : 'cat1 > cat2'
}

  -- blobs
  b_paa             blob,                       -- PAAnnotatedPage in compressed byte array form
  b_page            blob,                       -- PageData in compressed byte array form
  b_variant         blob,                       -- Variant for the product as served by mobile frontend apis

  -- canon urls
  c_canon_url       varchar,
  c_canon_key       varchar,
  c_url_flags       varchar,

  -- status flags
  s_deleted         int,                        -- 0 false, 1 true
  s_status_tags     list<varchar>,

  -- page_data:

  primary key ((merchant_id, m_pid))
);
-------------------------------------------------------------------------------------------------------------------------------
____
-- Generic Key-Value Store: Stores miscellaneous items such as category trees, static assets, etc.
--
create table m_mob_misc (
 merchant_id      varchar,                     -- merchant id
  m_rk            varchar,                     -- generic row key
```

```
  ts              timestamp,              -- timestamp for versioning
  mi_ns           varchar,                -- namespace of data
  mi_nk           varchar,                -- namespace key

  cat_tree        varchar,

  nm_assets       varchar,
  nm_midday       varchar,

  scrape_cat      varchar,
  scrape_search   varchar,
  scrape_update   boolean,

  others          map<varchar, varchar>,

  primary key ((merchant_id, m_rk), ts, mi_ns, mi_nk)
) with clustering order by (ts desc);
```

[0053] An example schema for storing non-product information (e.g., for all or a group of merchants) is shown below.

```
-- Web Page View:
-- -------------
-- A web-page is a materialized-view that is accessible via url. A web-page may be composed of
-- many entities (some of which may correspond to other models in the system, while others being represented as
-- a bag of words or not represented).
--
-- key: mck
--
create table v_web_page (
  merchant_id     varchar,
  m_ck            varchar,              -- PartitionKeyType: merchant canon-key
  ts              timestamp,            -- versioning timestamp
  src             varchar,              -- ProjectSrc: source of the data
  ptype           varchar,              -- PageType: page-type

  -- Instead of duplicating the various column schemas from the various models that contributed to this web-page,
  -- we simply treat that data as uninterpreted key-value pairs and delegate to the model logic for recreating
  -- structured data.

  -- m_product stores Product proto, which includes PageData & PAAnnotation
  -- m_product_quality stores ProductQuality proto

  b_page          blob,               -- PageData in compressed byte array form
  c_canon_url     varchar,
  c_url_flags     varchar,
  s_deleted       int,                -- 0 false, 1 true
  s_status_tags   list<varchar>,      -- ArrayList serialized to json

  primary key (m_ck, ts, src, ptype)
) with clustering order by (ts desc);
```

[0054] An example schema for personalization (e.g., for all or a group of merchants) is shown below.

```
//-----------------------------------------------------------------------
create table shared_items (
  merchant_id  varchar,
  id       varchar,                  // item id
  ts       timestamp,                  // timestamp - only for versioning, HEAD = '9999-01-01+0000'

  // current owner
  coid     varchar,                   // owner id
  copp     varchar,                   // owner passphrase
  cots     timestamp,                   // ownership timestamp

  // last owner (to avoid race during ownership transfer)
  loid     varchar,                  // owner id
  lots     timestamp,                   // ownership timestamp
  lopp     varchar,                  // owner passphrase

  title    varchar,                 // visible name of the item
  cts      timestamp,                   // creation timestamp
  inv      boolean,                  // is the item no longer valid? (unset = valid)

  // attributes
  attrs    map<varchar, varchar>,

  //
  // Children of this item (e.g. products belonging to a favorites list).
  // e.g. products are children of favorites:
  // i.e. k: p=prod1234  v: { price: 30, title: studded shoe, brand: prada }
  //
  // NOTE: collections do not support writetime, so if we need to know when updates happened,
  // we would need another column or key inside the collection.
  //
  parts    map<varchar, varchar>,

  primary key ((merchant_id, id), ts, coid, copp)
) with clustering order by (ts desc);

//-----------------------------------------------------------------------
// objects that refer to a user.
//    user <--- item (via action)
//    user <--- user (via handle + item)
create table user_ref (
  merchant_id   varchar,
  id       varchar,
  refid    varchar,
  ts       timestamp,
  hl       varchar,

  attrs    map<varchar, varchar>,

  primary key ((merchant_id, id), ts, refid)
) with clustering order by (ts desc, refid asc);
// These tables store user profile preferences.
create table user_profile_1 (
 merchant_id varchar,
  id varchar,  // user id or cohort id ("feed" is special user)

  p_ns varchar,  // primary key namespace (reserved)
  p_nk varchar, // primary key namekey (reserved)

  last_update timestamp,
  user_profile_response blob, // UserProfileResponse protobuf as byte array
```

```
  num_pref int,
  total_visits int,

  primary key ((merchant_id, id), p_ns, p_nk)
);

// This table stores user facts
create table user_facts (
  merchant_id   varchar,
  id varchar,  // user id or cohort id ("feed" is special user)

  p_ns varchar,  // primary key namespace (reserved)
  p_nk varchar, // primary key namekey (reserved)

  last_update timestamp, // timestamp of latest user fact
  user_facts blob, // UserProfileResponse protobuf as byte array
  primary key ((merchant_id, id), p_ns, p_nk)
);

// Table for storing product recommendations
create table product_recommendations (
  merchant_id   varchar,
  id varchar,  // product id

  p_ns varchar,  // primary key namespace (reserved)
  p_nk varchar, // primary key namekey (reserved)

  last_update timestamp, // timestamp of latest recommendation
  recommendations blob, // ProductRecommendations protobuf as byte array
  primary key ((merchant_id, id), p_ns, p_nk)
);
```

[0055] An example architecture and processes for ingesting data from merchant data sources by the data storage layer will now be described with respect to Figure 2.

[0056] **Figure 2** is a functional block diagram illustrating an architecture for ingesting data from merchant data sources at the data storage layer in accordance with some embodiments. In one embodiment, the above-described merchant data sources, including merchant data feeds, from the data sources domain are ingested in real-time at the data storage layer (e.g., distributed data storage layer, such as further described herein). In one embodiment, merchant data feed retrieving engine 136 is implemented as shown in Figure 2 and as further described below.

[0057] As shown, a merchant can call a feed input service 202 to provide a new/updated merchant data feed in real-time via a real-time ingestion engine 206 to the data storage layer, which includes a back-end US-East-1 cluster 208 including a search index 210 (e.g., for searching and indexing of merchant data to provide a search index of the merchant data, which can be implemented using a commercially available, open source platform such as SolrCloud, as further described herein) and a data store 212 (e.g., for storing the merchant data and for supporting primary key-based searching of the merchant data, which can be

implemented using a commercially available, open source platform such as Cassandra, as further described herein). In some cases, this data can be a subset of the overall set of product attributes of the merchant's product data (e.g., an incremental merchant data feed) that are time sensitive (e.g., quantity, in/out of stock, price, and/or other time sensitive related product attributes). For example, pricing and inventory can be time sensitive information that can be updated using the feed input service to facilitate real-time ingestion of such time sensitive product attributes at the data storage layer.

[0058] As also shown, feed input service 202 and real-time ingestion engine 206 are in communication through a real-time queue 204 to facilitate an internal queue update of the product information in the merchant data feed in real-time to the data storage layer. As used herein, real-time generally refers to providing the merchant data feed to the data storage layer within seconds (e.g., within a few seconds) for small workloads (e.g., a data feed that includes approximately 5,000 to 10,000 items, in which each item is approximately 10 KB in size), and for certain spikes in merchant data feed updates, real-time generally refers to providing the merchant data feed to the data storage layer within minutes (e.g., within a few minutes, or possibly more than a few minutes for very large and sudden spikes in merchant data feed updates) for large workloads (e.g., a data feed that includes approximately 100 million items, in which each item is approximately 10 KB in size). In some cases, such as is often the case in the context of products and associated attributes, minutes is usually within tolerable limits for most merchants.

[0059] **Figure 3** is a functional block diagram illustrating a distributed and fast data storage layer for large scale web data services including an example application that utilizes the data storage layer in accordance with some embodiments. As shown, merchant data sources include merchant web pages 302, merchant data feed 304, and merchant pixel logs 306 (e.g., providing behavioral data on a merchant's web site, such as similarly described above with respect to Figure 1). Merchant pixel logs 306 are provided via merchant web pages 302 as input along with merchant data feed 304 to a merge product information engine 308. In one embodiment, merge product information engine 308 merges various merchant data sources, including merchant web pages 302, merchant data feed 304, and merchant pixel logs 306 as shown in this example, to provide an accurate and consistent view of a merchant's web site related data and product catalog. The merged product information is provided from the merge product information engine 308 to data store 310 (e.g., a data store of the distributed data storage layer, such as similarly described herein with respect to Figure 1).

[0060] As also similarly described above with respect to Figure 1, the data storage layer can provide various applications/services using the merged product information of a given merchant maintained in a data store. As shown, data store 310 is in communication with a "What's Hot" application 312 and other applications/services 314 and 316, which can then communicate with a given user's browser, shown as user browser 318 in the merchant domain.

[0061] For example, "What's Hot" application 312 is an example application/service that can determine which products are popular (e.g., such as products that are currently popular that are available for sale on a merchant's web site) based on real-time merged product information for a merchant's web site related data. "What's Hot" application 312 can then display any such popular products on the merchant website (e.g., such popular products can be dynamically displayed using a widget on the merchant's web site or other techniques can be used for automatically displaying such popular products on the merchant's web site that can then be viewed by a user via their user browser). In an example implementation, the "What's Hot" application can display products that receive the highest number of user visits in a given time period (e.g., within the last one hour, one day, one week, or some other time period). In this example, for each product, the "What's Hot" application can display fresh and accurate information including price, out-of-stock, and/or other product related information based on real-time merged product information for a merchant's web site related data maintained at data store 310. In some cases, the products may go on sale and their price may be reduced or they may be sold out. For example, for each product, the "What's Hot" application can query data store 310 using a product ID as the primary key. In response, data store 310 can return the most up-to-date and accurate information about the product including, for example, title, description, price, and out-of-stock information associated with the product ID that is stored in the data store. This current product information can then be displayed to the user via the user's browser.

[0062] In one embodiment, the data storage layer provider can provide or support many different applications/services that utilize real-time merged product information for a merchant's web site related data maintained in the data store. For example, the applications/services can utilize real-time merged product information for a merchant's web site related data maintained in the data store of the data storage layer by searching such product information based on different search queries (e.g., search strings, product IDs, and/or other search queries) to obtain additional, updated, and/or more detailed information for one or more products provided by a given merchant, such as similarly described above

with respect to the "What's Hot" application/service. Depending on the application, the search and merchant data extraction can be performed in a front-end environment under very low latency conditions or in a back-end environment under very high throughput conditions, such as further described below with respect to various embodiments.

[0063] An organic search related application is another example application/service that can be facilitated by a data storage layer using the various techniques described herein. For example, assume that a given merchant has an organic search box on the merchant's web site that allows a user to submit search string queries to attempt to find products/services of interest to the user on the merchant's web site. By leveraging a deep understanding of the content on a merchant's web site and web's content, an organic search application can be implemented to automatically match a given user's intent (e.g., consumer intent) to the merchant's content to identify relevant products/services on the merchant's web site, which can facilitate increased merchant revenue by improving organic search ranking of relevant products/services for a given user.

[0064] A related searches application is another example application/service that can be facilitated by a data storage layer using the various techniques described herein. For example, a related searches application can utilize real-time merged product information for a merchant's web site related data maintained in the data store of the data storage layer to automatically determine and display relevant links to a page to flatten a web site structure and enhance content richness of a merchant's web site.

[0065] A related products application is another example application/service that can be facilitated by a data storage layer using the various techniques described herein. For example, a related products application can utilize real-time merged product information for a merchant's web site related data maintained in the data store of the data storage layer to automatically identify and add content to a web page of a merchant's web site to make that web page more discoverable and easy for search-engine bots to crawl and index (e.g., to improve search engine results when users perform organic searches using web search engines, such as search engines available from Google, Microsoft Bing, Yahoo, and/or other search engine service providers). As a result, the related products application can make it easier for users to discover such web pages on a merchant's web site using organic search and discover related products via browsing the merchant's web site.

[0066] A thematic pages application is another example application/service that can be facilitated by a data storage layer using the various techniques described herein. For example, a thematic pages application can utilize real-time merged product information for a merchant's web site related data maintained in the data store of the data storage layer to automatically create, edit or retire recommended pages that match products with intent (e.g., a user's intent, based on, for example, a user's behavior/browsing activity on the merchant's web site and/or other merchants' web sites).

[0067] A Search, Navigation, and Personalization (SNAP) application is another example application/service that can be facilitated by a data storage layer using the various techniques described herein. For example, a SNAP application can utilize real-time merged product information for a merchant's web site related data maintained in the data store of the data storage layer to automatically personalize site search, content, and navigation to make each visit personal for every user across all devices (e.g., across all of a user's devices, such as a given user's smart phone, tablet, and laptop). SNAP can integrate consumers' behaviors, a given merchant's web site content, and broader web-wide data available on demand when a shopper uses natural language search queries to search the merchant's web site. In an example implementation, SNAP generates for each and every consumer a uniquely personal experience tailored to their preferences while matching current intent.

[0068] A compass application is another example application/service that can be facilitated by a data storage layer using the various techniques described herein. For example, a compass application can utilize real-time merged product information for a merchant's web site related data maintained in the data store of the data storage layer to automatically understand a merchant's products and identifies opportunities to maximize revenue. In some cases, such content understanding enables action on the relationships between the attributes and types of products sold by the merchant via the merchant's web site.

[0069] In one embodiment, the data storage layer can be implemented using cloud/remote computing services, such as provided by various cloud computing service providers (e.g., Amazon, HP, IBM, and other companies offer such commercially available cloud computing services). For example, the data storage layer can be implemented using Amazon Web Services (AWS) for data storage and computing purposes performed by the data storage layer as described herein. Various AWS instance types that can be used to implement the data storage layer are further described below.

[0070] For example, Amazon Elastic Compute Cloud(R) (EC2) is a commercially available cloud computing service from Amazon that can be used to implement the data storage layer for compute purposes. Amazon EC2 is hosted in multiple locations worldwide. These locations are generally composed of regions and availability zones. Each region is typically a separate geographic area. Each region typically has multiple, isolated locations known as availability zones.

[0071] In an example implementation, the data storage layer described herein includes a dedicated Cassandra cluster(s). For example, dedicated Cassandra clusters can be implemented using the following example hardware environment: 60 X i2.xlarge (e.g., across three availability zones in US-East-1 region), 12 X i2.xlarge (e.g., across three availability zones in US-West-1 region), and 6 X i2.xlarge (e.g., across three availability zones in EU-West-1 region).

[0072] In an example implementation, the data storage layer described herein also includes a dedicated SolrCloud cluster(s). For example, dedicated SolrCloud clusters can be implemented using the following example hardware environment: 12 X r3.2xlarge (e.g., across three availability zones in US-East-1 region), 6 X r3.2xlarge (e.g., across three availability zones in US-West-1 region), and 6 X r3.2xlarge (e.g., across three availability zones in EU-West-1 region).

[0073] In an example implementation, the data storage layer described herein can also include on-demand clusters. On demand clusters can be distributed across one or more availability zones and can also be continuously optimized for cost efficiency. For example, Amazon EC2 Spot Instances can be used for the on-demand cluster(s). A resource manager, such as further described below), can continuously optimize a cluster size for cost and usage efficiencies. An example hardware configuration for the on-demand clusters at a particular time is as follows: 14 X m1.medium, 7 X m1.large, 11 X r3.large, 4 X r3.xlarge (e.g., across four availability zones in US-East-1 region), 9 X hi1.4xlarge, 1 X c3.8xlarge, and 2 X c3.large (e.g., across four availability zones in US-West-1 region).

[0074] As will now be apparent to one of ordinary skill in the art, various other hardware environments and/or cloud service providers can be used to efficiently and effectively implement the data storage layer to perform the various techniques disclosed herein. As another example, a data storage layer provider can implement the data storage

layer using hardware in its own data center(s) and/or using a combination of hardware in its own data center(s) and one or more cloud service providers.

[0075] In one embodiment, the data storage is a high-availability storage layer, horizontally scalable, supports primary key look-up of product/service merchant product information, supports search by various attributes of a document (e.g., a web page), and provides a reliable and fast data storage layer infrastructure. Each of these aspects of the distributed data storage layer is further described below.

[0076] In an example implementation, the data storage is a high-availability storage layer that is fault-tolerant and can provide high up-time to customers (e.g., 99 percent up-time or some other metric that corresponds to a high up-time to customers). For example, the distributed cluster architecture of the data storage layer is resilient in the presence of failures of individual nodes in the system.

[0077] In an example implementation, the data storage is also horizontally scalable (e.g., scale-out, infinitely scalable). For example, the data storage layer can scale the infrastructure by adding more nodes to the system. Generally, there should not be an upper limit to the amount by which the system can be scaled using the techniques disclosed herein.

[0078] In an example implementation, the data storage layer can also support look-up by primary key for an application/service supported/provided by the data storage layer provider. For example, as similarly described above, various applications/services can utilize the merchant product information stored at the data storage layer to perform primary key look-ups into the merchant product information to facilitate various applications/services.

[0079] In an example implementation, the data storage can also support scans of the entire data store for an application/service supported/ provided by the data storage layer provide. For example, as similarly described above, various applications/services can utilize the merchant product information stored at the data storage layer to perform scans of all merchant products to facilitate various applications/services.

[0080] In an example implementation, the data storage can also support search by various attributes for an application/service supported/provided by the data storage layer provider. For example, various applications/services can utilize the merchant product information stored at the data storage layer to perform searches by various attributes of a

document that is a canonical/internal representation of a web page of a merchant's web site to facilitate various applications/services.

[0081] In an example implementation, the data storage layer can also provide predictable and fast performance for all read/write/scan/search requests. For example, 99 percent of read, write, scan, and/or search requests can be successfully completed within a given time threshold in an example implementation.

[0082] In one embodiment, the data storage layer includes a highly available, fault tolerant, and horizontally scalable key-value storage solution. Example storage solutions that can be used for implementing such a data storage layer include the following commercially available, open source storage solutions: Apache Cassandra, Apache HBase, BerkeleyDB, and LevelDB. As will now be apparent to one of ordinary skill in the art, various other commercially available and/or open source storage solutions can similarly be used to implement the data storage layer.

[0083] Various properties of the data store layer can include the following requirements for the data storage layer to implement the various techniques described herein, as further described below.

[0084] Each data item (e.g., data items from merchant data sources) is replicated multiple times and stored on different physical and/or virtual nodes so that merchant data can still be retrieved if one or more of these nodes are unavailable. For example, this replication approach facilitates fault tolerance for the data storage layer.

[0085] A failure of one or a few nodes does not affect the availability of the system, which is implemented as a distributed system. Also, transient problems like network connections would not affect the availability of the system (e.g., due to the replication and geographical distribution architecture of the data storage layer implementation disclosed herein).

[0086] Data can be replicated across physically isolated data centers. In this case, data is still available for read/write as long as at least one copy of the data is available.

[0087] A cluster can be scaled to store X times the current data size by, for example, simply adding X times more nodes to the cluster. For example, the data storage layer is horizontally scalable as similarly discussed above.

[0088] In some cases, applications/services may be sensitive to the "latency" of read/write requests. The "latency" of the request as seen by the client (e.g., user browser) can be defined as the time interval between which the request is made by the client and a response is received by the client. For example, some applications may require certain latency requirements for read and write requests. In some cases, TP (top percentile) 95/99 --> t guarantees can be provided within a time t, as a service level agreement (SLA).

[0089] In some cases, applications may perform bulk operations on the data store of the data storage layer. In some cases, a bulk operation performs a read/write for many rows in the data store (e.g., for different primary keys) in the same operation. For bulk operations, it is desired that the data storage layer provide high throughput measured as the maximum number of simultaneous read/write operations that can be supported.

[0090] In one embodiment, the data storage layer also includes a reverse search index that can support efficient look-ups by secondary attributes (e.g., a reverse look-up index). A secondary attribute as used herein generally refers to any attribute of a web page that is not a part of the primary key as defined by the application. This may be generally referred to as search. The process of adding documents to the search index (e.g., or simply referred to as the index) is generally referred to as indexing. As an example implementation, various commercially available, open source platforms that can be used to provide such search functionality include Apache Lucene and Apache Solr. However, a limitation of these platforms is that they do not support horizontal scalability as described above, because these platforms cannot be scaled simply by adding more nodes to the system. Apache SolrCloud is an example of a system that can be scaled horizontally by sharding the data. A shard as used herein generally refers to any subset of the original dataset. Data can generally be partitioned into multiple shards such that they are mutually exclusive but together they comprise the entire data set. For example, using this approach, a reverse look-up index can be provided by the data storage layer to support various search use cases, such as a user query for "Samsung" to retrieve a particular Samsung product, even if the primary key (e.g., product ID) is not known from the query.

[0091] Another property of the data storage layer disclosed herein is data consistency. There are several models for data consistency including strong consistency and eventual consistency, which are each described below. In some cases, strong consistency can be defined as the property of a system that ensures all accesses are seen by all parallel processes (e.g., or nodes, processors, etc.) in the same order (sequentially). In some cases, eventual

consistency informally guarantees that, if no new updates are made to a given data item, eventually all accesses to that item will return the last updated value.

[0092] In one embodiment, a system for the data storage layer is disclosed that provides eventual consistency for all data maintained (e.g., stored) at the data storage layer. Also, if a data item is updated in the data store of the data storage layer, that data item can be updated in the reverse search index on a best-effort basis. Thus, in this case, there is no strong consistency requirement for such data updates in this example implementation of a system for the data storage layer, but eventual consistency is provided by this example implementation of a system for the data storage layer.

[0093] In one embodiment, a centralized data store also facilitates data sharing, in which data computed by one application can be stored in the data store at the data storage layer and can generally be used at no additional cost by one or more other applications/services provided or supported by the data storage layer provider. A centralized data store also facilitates centralized operations, in which one or more other applications can use the same cluster of nodes that prevents each application from maintaining its own cluster and, therefore, can reduce overall maintenance overhead.

[0094] **Figure 4** is a functional block diagram illustrating a topology of a distributed system for implementing the data storage layer in accordance with some embodiments. In one embodiment, the data storage layer provides fast and guaranteed performance in terms of response time, such as similarly described above, using various techniques described herein. For example, a topology of a distributed system for implementing the data storage layer as shown in Figure 4 can provide consistent and fast performance for all applications across geographical locations, such as US East, US West, EU West, and/or other geographies, as further described below.

[0095] As similarly discussed above, the input for processing by the back-end is the merchant data feed. In one embodiment, the topology for the data storage layer includes a back-end cluster and a front-end cluster shown, which each include a search index (e.g., implemented using Apache SolrCloud or another indexing framework/platform, facilitating look-up by attribute/reverse index lookup, such as similarly described above) and a data store (e.g., implemented using Cassandra or another data store, facilitating primary key look-up, such as similarly described above). As shown, the data storage layer includes a back-end cluster that includes a back-end US East 424 that includes a search index 426 and a data store

428. As also shown, the back-end cluster is in communication with the front-end cluster that includes the following: a front-end US West 432 that includes a search index 434 and a data store 436, a front-end US East 438 that includes a search index 440 and a data store 442, and a front-end EU 444 that includes a search index 446 and a data store 448.

[0096] It is important to note the distinction between the front-end cluster and the back-end cluster. In this example implementation, the merchant data is replicated between the back-end cluster and the front-end cluster.

[0097] For example, the back-end cluster can be primarily used by applications performing batch operations that aim to maximize the throughput for the cluster. MapReduce (e.g., such as using Hadoop to efficiently distribute jobs in a distributed environment, such as can be implemented by the AWS Elastic MapReduce service) is an example of such an application. As shown, back-end job manager 420 is in communication with back-end US East 424 via a resource aware adaptive capacity/throttling engine 422 for distributing jobs to the back-end cluster or copy on demand dynamic clusters, such as further described below. In one embodiment, the back-end job manager uses the resource aware adaptive capacity/throttling engine for distributing jobs and creating/terminating additional back-end copy-on-demand clusters (e.g., as needed/based on demand and/or other configured requirements).

[0098] For example, the front-end cluster can primarily be used by applications that are sensitive to latency and are configured to ensure that a given percentile of requests (e.g., 99 percentile or some other performance threshold SLA commitment or target) are services within a given time threshold. The front-end cluster may be available in different geographical locations as shown in Figure 4, such as US East and US West (e.g., geolocation distribution to facilitate fast response regardless of network latency). The front-end cluster for a merchant is typically located in a geographical location physically close to a merchant data center to optimize network latencies between the merchant data center and the data storage layer data center.

[0099] In one embodiment, data replication from the back-end cluster to the front-end cluster is eventually consistent, such as similarly described above. For example, if no new updates are made to the data item, then all access to data items in the front-end cluster will return the last updated value. In some implementations, the web pages stored in the data

store are also stored in the search index to allow for search by secondary attributes, such as similarly described above.

[00100] In an example implementation, the back-end job manager (e.g., an elastic Hadoop service) can be elastic and scale to large volumes of data. However, the Cassandra database and SolrCloud search index are generally not capable of scaling that quickly to large volumes of data. As such, this can result in scaling problems that denigrate compute performance of the system for the distributed data storage layer. For example, if there is a terabyte of merchant data that is stored on three clusters on three different machines/nodes, and three new machines/nodes are to be added, such as shown in Figure 4, then the time cost to redistribute/replicate that terabyte of merchant data across the three new machines/nodes can require a significant amount of time, such as 24 hours or longer.

[00101] Accordingly, a dynamic clustering solution using a copy on demand search index and data store is disclosed to solve these technical problems in accordance with some embodiments. Referring to Figure 4, a dynamic cluster includes a copy on demand 402 that includes a search index 404 and data store 406, a copy on demand 408 that includes a search index 410 and data store 412, and a copy on demand 414 that includes a search index 416 and data store 418 as shown. For example, the copy on demand search index and data store can be provided by capturing a snapshot of the data and the search index in a dynamic cluster. As a result, such a copy on demand cluster(s) can scale linearly at the same pace as, for example, an AWS EMR job can scale.

[00102] In one embodiment, resource aware adaptive capacity/throttling engine 422 can determine whether to perform a back-end job using a copy on demand cluster for processing a particular job for a given merchant. For example, if a large data feed processing job for a Merchant A is received from the back-end job manager, then resource aware adaptive capacity/throttling engine 422 can delegate the job to be performed using a copy on demand dynamic cluster for Merchant A to efficiently perform the job for Merchant A without disrupting job processing for other merchants that have merchant data stored and processed using the back-end cluster, such as further described below.

[00103] In one embodiment, data processing is performed using a new copy on demand cluster layer in the topology of the data storage layer that facilitates copy on demand to provide for state-based data processing, and then the back-end cluster of the data storage layer is only responsible for storing the data, not processing the data (e.g., processing and

merging of updated merchant data can be handled by the copy on demand dynamic cluster layer, such as described herein). In one embodiment, copy on demand is implemented using a copy on demand cluster layer, such as shown in Figure 4, to perform per merchant data feed processing. For example, this allows for a processing demand of updating one merchant's data feed to not impact another merchant's data feed processing to facilitate performance isolation between merchants (e.g., processing data feed of a large merchant does not impact other merchants). Also, using this copy on demand cluster technique can more quickly replicate (state) data of a database for a given merchant to perform that processing of the merchant's data feed and verifying/maintaining data consistency (e.g., to update and reconcile updated merchant data received in a new merchant data feed with existing merchant product information), such as similarly described above. In an example implementation, this copy on demand cluster technique provides for a copy/snapshot of the SolrCloud cluster and/or a copy/snapshot of the Cassandra cluster to facilitate copy on demand data processing as described herein.

[00104] In one embodiment, fixed pricing for cloud-processing services pricing is used to implement job processing for the back-end using the copy-on-demand techniques described above.

[00105] In one embodiment, demand-based pricing for cloud-processing services pricing is used to implement job processing for the back-end using the copy-on-demand techniques described above. In one embodiment, to reduce cost and ensure high availability of machines (e.g., physical and/or virtual servers) for scaling at a rapid rate, a resource manager 430 is provided to reduce or optimize costs associated with such demand-based pricing for cloud-processing services pricing used to implement job processing for the back-end using the copy-on-demand techniques described above. For example, resource manager 430 can select machines (e.g., physical/virtual compute resources of a cloud computing service) based on a back-end job manager requirement (e.g., based on an AWS EMR job requirement) and the market price of the instances on a market place for a cloud computing service provider (e.g., the Amazon Spot Instance market place for AWS or another cloud computing service provider, in which there is demand-based pricing for cloud-processing services).

[00106] In one embodiment, resource manager 430 uses a resource aware adaptive capacity/throttling engine 422 to meet resource requirements for performing back-end job processing using the back-end cluster and/or copy on demand dynamic cluster. In an

example implementation, resource aware adaptive capacity/throttling engine 422 implements the following rules: (1) fixed capacity (e.g., defined limits); (2) enforced limits; (3) quotas per application (e.g., to meet service level commitment (SLA) per merchant, such as within 1-hour to update merchant pricing)); and/or (4) dynamic rate limiting.

[00107] The following algorithm is an example of how resource manager 430 can optimize for the least expensive computation resource for performing a given job(s). Similarly, the below shown algorithm can be adjusted for optimizing for disk resource or memory resource utilization as will now be apparent to one of ordinary skill in the art.

```
maxCpuPerUnitPrice = 0
optimalInstanceType = null
For each instance_type in (Availability Zone, Region) {
  cpuPerUnitPrice = instance.cpuCores/instance.spotPrice
  if (maxCpuPerUnitPrice < cpuPerUnitPrice) {
    optimalInstanceType = instance_type;
  }
}
```

[00108] For example, given the potential fluctuations in the spot price of an instance on the market place, for ensuring availability and the stability of the on-demand cluster, the spot price can be computed as a moving average over the time axis to avoid selecting instances that have very high peak demand and variations in prices. As such, the resource manager can reduce or optimize a cost for providing job processing for the back-end using the copy-on-demand techniques described above.

[00109] Another technical problem that can arise is that the front-end data centers can have a requirement to provide very low latencies for data access and search. The main contention on the front-end data centers typically arises from the actual throughput from front-end applications and data copy for new data generated by the back-end jobs. To reduce the contention of the data generated by back-end jobs, a column-level replication strategy is performed, in which data is replicated from the back-end to the front-end on an individual column level (e.g., for each attribute of a product in the merchant data, (custom) rules can be applied to select which data is to be replicated to the front-end and what data should only be maintained/stored at the back-end). For example, a front-end application may not require the description of a product for display, however, the back-end application may need to generate a search index that is based on the description of the product. In this case, using a column-level replication strategy, a workload on the front-end is significantly reduced by not replicating such product description data to the front-end. This column-level replication

strategy also allows for dynamic run-time decisions that can be merchant-based without requiring any development effort to change the view on the front-end.

[00110] Accordingly, a column-level replication is performed to reconcile merchant data that is stored at the back-end cluster with a subset of merchant data that is copied to and stored at the front-end cluster of the distributed data storage layer that facilitates dynamic data modeling (e.g., at run-time/real-time) in accordance with some embodiments. For example, intelligently subsetting the merchant data (e.g., dynamic data modeling) at real-time, such as by only replicating the title and an image of a given merchant's product data (e.g., not product description, SKUs, etc.) to the front-end clusters (e.g., only replicating a subset of columns from the database based on rules, which can be default or custom rules for a given merchant), reduces the amount of data for replication between the back-end and front-end clusters per merchant. This approach also facilitates capacity and performance of the data storage layer in communication with the front-end layer (e.g., application layer).

[00111] An example filter for efficiently controlling data flow between the back-end and front-end clusters is shown below, which provides an example schema for filtering product attribute data to subset product information in order to reduce replication as similarly described above for implementing this column-level replication strategy.

```
DataCenter: pagedb-front-end:
ColumnFamily : \w+[.]m4042_product
Exclusion Rule : *   // Remove all
ColumnFamily: \w+[.]sears_thematic
Exclusion Rule : *
ColumnFamily: \w+[.]m\d\d\d\d_product
Exclusion Rule : b_paa,c_url_flags,s_deleted,s_status_tags
ColumnFamily: \w+[.]\w+_thematic
Exclusion Rule : crumb_nav,left_nav,theme_rank,rendered_page
ColumnFamily: \w+[.]br5057_product
Exclusion Rule :
b_paa,b_page,c_canon_key,c_canon_url,c_url_flags,cat_crumbs,cat_goog,cat_nodes,d_ag
e_group,d_brand,d_color_group,d_colors,d_conditions,d_description,d_flags,d_gender,d_ke
ywords,d_la_img_urls,d_material,d_mature,d_model_name,d_others,d_pattern,d_sizes,d_st
ores,d_sw_img_urls,d_th_img_urls,d_title,d_urls,d_zo_img_urls,i_buyable,i_have_avail,i_ha
ve_local_avail,i_last_avail_ts,i_level,i_status,l_first_live_ts,l_is_live,l_last_live_ts,l_launch_ts
,pg_i_refs,pg_o_refs,pg_type,s_deleted,s_status_tags
ColumnFamily: \w+[.]br4042_product
Exclusion Rule:
b_paa,b_page,c_canon_key,c_canon_url,c_url_flags,cat_crumbs,cat_goog,cat_nodes,d_ag
e_group,d_brand,d_color_group,d_colors,d_conditions,d_description,d_flags,d_gender,d_ke
ywords,d_la_img_urls,d_material,d_mature,d_model_name,d_others,d_pattern,d_sizes,d_st
```

ores,d_sw_img_urls,d_th_img_urls,d_title,d_urls,d_zo_img_urls,i_buyable,i_have_avail,i_have_local_avail,i_last_avail_ts,i_level,i_status,l_first_live_ts,l_is_live,l_last_live_ts,l_launch_ts,pg_i_refs,pg_o_refs,pg_type,s_deleted,s_status_tags

[00112] In one embodiment, user requests to the front-end are prioritized over back-end processing to facilitate performance and service for user requests for applications/services provided by the data storage layer service provider. For example, another mechanism through which latencies for front-end can be reduced is by implementing a prioritization mechanism whereby a request originating from a front-end application is given higher precedence compared to a back-end (application) request.

[00113] **Figure 5** is a functional block diagram illustrating a data storage layer in communication with an application in accordance with some embodiments. As shown, a data store 502 (e.g., implemented using Cassandra, such as similarly described above) is in communication with a search index 504 (e.g., implemented using SolrCloud, such as similarly described above). As also shown, an application 506 is in read/write/scan communication with data store 502, and search index 504 is in read access communication with application 506 (e.g., an application/service provided by the data storage layer provider, such as a web-related application/service as similarly described above).

[00114] In some cases, applications can store information from various external sources. For example, an application can store web page information from web-wide web data (e.g., web pages from web sites, such as from merchant web sites, and/or other web sites). As another example, an application can store merchant data feed information about products and other web pages provided by a given merchant.

[00115] For example, applications can then compute derived information based on the above information and store the derived information in the data store by writing the data to the data store as shown in Figure 5.

[00116] **Figure 6** is a functional block diagram illustrating a data storage layer in communication with a web search related application in accordance with some embodiments. In particular, a web search related application interacts with the data storage layer as similarly shown in and described above with respect to Figure 5.

[00117] A canon key as used herein generally refers to a canonical representation of a URL. Pagedata as used herein generally refers to a canonical representation of a web page.

In some cases, pagedata can be retrieved from the HTML content of the web page. In the case of the merchant data feed, pagedata can be constructed from the information provided in the merchant data feed. Generally, pagedata provides a set of attributes describing a given web page. In the data store, the primary key for look-up as used herein generally refers to the canon key. A row in the data store as used herein generally refers to a tuple containing all attributes corresponding to a given primary key. A title of a web page in some cases can be defined as the HTML element defined by the tag <title>. A heading of a web page in some cases can be defined as the HTML element defined by tag <h1>. A query can generally be defined as a collection of words separated by white spaces.

[00118] In some embodiments, a web page is classified using the below described categories of web pages. A product web page generally represents an individual item that may be offered for sale by a third-party merchant. A category web page generally represents a collection of products. In some cases, these web pages can be manually generated or undergo a quality procedure to ensure high quality. A search web page generally represents a collection of products that is generated dynamically based on a search query. Other web pages generally represent any web page that does not belong to any of the above categories.

[00119] Referring to Figure 6, an example web search related application can perform the following behaviors that include performing various read and write operations from and to data store 602 as further described below.

[00120] A merchant web site crawl data engine 604 crawls/acquires web pages from the WWW crawled data including a merchant web site. For example, merchant web site crawl data engine 604 can parse the HTML content of a web page and store a canonical representation as pagedata. In particular, this crawl related pagedata can be stored in data store 602 using a canon key as the primary key.

[00121] A merchant feed data engine 606 parses a merchant data feed provided by the merchant. For example, merchant feed data engine 606 can store a canonical representation of each item in the merchant data feed as pagedata. In particular, this merchant feed related pagedata can be stored in data store 602 using a canon key as the primary key.

[00122] A merge crawl/feed data engine 608 performs a merge operation. In one embodiment, a merge operation includes a process of merging attributes for the same canon key from different sources, including, for example, crawl data from the WWW crawled data and the merchant feed data. For example, a sample attribute for a product is a price attribute

that represents the price for sale of the product. In some cases, it may be possible that a price retrieved from WWW crawl data is different from the price retrieved from feed data for the same canon key. In this case, the merge process can be performed to compute a final value for the price attribute from all data sources to ensure data consistency as similarly described above. The merged crawl/feed data is then stored in data store 602.

[00123] A product attribute computation engine 610 computes product attributes for each product in a given merchant's data sources. In an example implementation, for all pages, product attribute computation engine 610 reads crawl pagedata and feed pagedata from the data store, and computes merged pagedata using the merge process as similarly described above. For all products, product attribute computation engine 610 then reads the merged pagedata and computes product attributes. The computed product attributes for each product for the merchant are then stored in data store 602.

[00124] A search index 612 is generated for the merchant related data in data store 602 as similarly described above. In an example implementation, for all pages in data store 602, a reverse search index is automatically generated to facilitate searching of the merchant related data by any attribute, such as similarly described above. For example, the search index can allow for searching all products that contain the query "green lace dress" in a title or heading field or in a description for the web page.

[00125] A scan engine 614 is in communication with data store 602. For example, various applications/services can utilize the merchant product information stored in data store 602 of the data storage layer to perform scans using scan engine 614 of all merchant products to facilitate various applications/services, such as similarly described above.

[00126] A metric/counter engine 616 is in communication with data store 602. For example, metric/counter engine 616 can determine various metrics or counters of data items (e.g., merchant products or other information) stored in data store 602, such as similarly described above.

[00127] A search engine 618 is in communication with data store 602. For example, search engine 618 can perform a search of data items (e.g., merchant products or other information) stored in data store 602 using the reverse search index to perform reverse index look-up operations or other search operations, such as similarly described above.

[00128] **Figure 7** is a flow diagram illustrating the process for providing a distributed and fast data storage layer for large scale web data services in accordance with some embodiments. In one embodiment, process 700 is performed by a distributed data storage layer, such as described above with respect to Figures 1-6.

[00129] The process begins at 702. At 702, storing web related data for a plurality of merchants in a distributed data storage layer is performed. For example, the web related data can include crawled web site data, feed data, and/or user behavior data for each of the plurality of merchants.

[00130] At 704, copying on demand a data store and a search index for web related data for a first merchant to perform a job for updating the web related data for the first merchant is performed. For example, a copy on demand data store and search index can be provided by capturing a snapshot of the data and the search index in a dynamic cluster, which can be used to facilitate improved scalability of back-end job processing of merchant data, such as similarly described above.

[00131] At 706, copying data from the data store and search index for web related data for the first merchant from the copy on demand cluster (e.g., dynamic cluster) to a primary cluster. For example, data from the copy on demand search index and data store, which is stored in the copy on demand cluster (e.g., dynamic cluster), can be copied to the primary cluster (e.g., main cluster), which can be used for performing front-end processing operations of merchant data, such as similarly described above.

[00132] **Figure 8** is another flow diagram illustrating the process for providing a distributed and fast data storage layer for large scale web data services in accordance with some embodiments. In one embodiment, process 800 is performed by a distributed data storage layer, such as described above with respect to Figures 1-6.

[00133] The process begins at 802. At 802, merchant data is received at a distributed data storage layer. For example, the merchant data can include crawled web site data, feed data, and/or user behavior data for each of the plurality of merchants.

[00134] At 804, merging product information is performed to ensure data consistency of the merchant data stored at the distributed data storage layer. For example, assume that a price retrieved from the WWW crawl data is different from the price retrieved from feed data for the merchant data. In this case, the product information can be merged to determine a

final value for price from all data sources to ensure data consistency, such as using various techniques as similarly described above.

[00135] At 806, merged product information is stored in a data store of the distributed data storage layer. For example, the merged product information can be stored in a data store of a back-end cluster of the distributed data storage layer.

[00136] **Figure 9** is a flow diagram illustrating the process for providing a distributed and fast data storage layer for large scale web data services in accordance with some embodiments. In one embodiment, process 900 is performed by a distributed data storage layer, such as described above with respect to Figures 1-6.

[00137] The process begins at 902. At 902, merchant web site crawl data is received at a distributed data storage layer. For example, the merchant data can include crawled web site data, which can also include user behavior data for each of the plurality of merchants.

[00138] At 904, merchant feed data is received at the distributed data storage layer. For example, the merchant feed data can include product information, such as (a subset of) a product catalog (e.g., in an XML or other data format).

[00139] At 906, merging product information is performed to ensure data consistency of the merchant product information at the distributed data storage layer. For example, assume that a price retrieved from the WWW crawl data is different from the price retrieved from feed data for the merchant data. In this case, the product information can be merged to determine a final value for price from all data sources to ensure data consistency, such as using various techniques as similarly described above.

[00140] At 908, product attributes are computed for the merged product information. For example, merged pagedata can be computed using the merge process as similarly described above. For all products, the merged pagedata can then be used to compute product attributes for each product offered for sale by the merchant.

[00141] At 910, a search index is generated for the merchant product information (e.g., the merged merchant product information). For example, a reverse search index can be automatically generated to facilitate searching of the merchant related data by any attribute, such as similarly described above.

[00142] At 912, whether the process is to be repeated for another merchant is determined. If so, processing returns to 902. If not, then the process is completed (e.g., until new or updated merchant data is received at the distributed data storage layer).

[00143] Although the foregoing embodiments have been described in some detail for purposes of clarity of understanding, the invention is not limited to the details provided. There are many alternative ways of implementing the invention. The disclosed embodiments are illustrative and not restrictive.

CLAIMS

1. A system for a distributed data storage layer for web data services, comprising:

the distributed data storage layer for storing web related data for a plurality of merchants; and

a copy on demand dynamic cluster including a data store and a search index for storing web related data for a first merchant to perform a job for updating the web related data for the first merchant using a processor.

2. The system recited in claim 1, wherein the copy on demand dynamic cluster including the data store and the search index for storing the web related data for the first merchant corresponds to a snapshot of the data store and the search index stored in the dynamic cluster of the distributed data storage layer.

3. The system recited in claim 1, wherein the web related data for the plurality of merchants is stored in a back-end cluster of the distributed data storage layer.

4. The system recited in claim 1, wherein the web related data for the plurality of merchants is stored in a back-end cluster of the distributed data storage layer, and wherein the web related data includes crawled web site data, feed data, and user behavior data.

5. The system recited in claim 1, wherein the distributed data storage layer merges and stores crawled web site data and feed data for each of the plurality of merchants.

6. The system recited in claim 1, further comprising:

an application that is in communication with the distributed data storage layer and a web site associated with the first merchant.

7. The system recited in claim 1, further comprising:

an application that is in communication with the distributed data storage layer and a web site associated with the first merchant, wherein the application is also in communication with a user browser.

8. The system recited in claim 1, further comprising:

a merge product information engine for merging crawled web site data and feed data for each of the plurality of merchants.

9. The system recited in claim 1, further comprising:

a merge product information engine for merging crawled web site data and feed data for each of the plurality of merchants, wherein the merge product information engine

maintains data consistency for the crawled web site data and feed data for each of the plurality of merchants.

10. The system recited in claim 1, further comprising:

a product attribute computation engine for reading crawl related product attribute data and feed related product data stored in the data store of the distributed data storage layer and for computing crawl related product attribute data and feed related product data for each of the plurality of merchants, wherein merged crawl related product attribute data and feed related product data for each of the plurality of merchants is stored in the data store of the distributed data storage layer.

11. The system recited in claim 1, further comprising:

a back-end job manager for delegating jobs for processing by a back-end cluster or the copy on demand dynamic cluster.

12. The system recited in claim 1, further comprising:

a resource aware adaptive capacity/throttling engine for determining resource requirements for performing back-end job processing based on one or more rules.

13. The system recited in claim 1, further comprising:

a resource manager for optimizing a cluster size for a back-end cluster and/or the copy on demand dynamic cluster for performing back-end job processing.

14. The system recited in claim 1, further comprising:

front-end cluster for storing a subset of the web related data for each of the plurality of merchants in the distributed data storage layer.

15. A method of a distributed data storage layer for web data services, comprising:

storing web related data for a plurality of merchants in the distributed data storage layer; and

copying on demand a data store and a search index for web related data for a first merchant to perform a job for updating the web related data for the first merchant.

16. The method of claim 15, wherein the copy on demand of the data store and the search index for storing the web related data for the first merchant corresponds to a snapshot of the data store and the search index in a dynamic cluster.

17. The method of claim 15, wherein the web related data for the plurality of merchants is stored in a back-end cluster of the distributed data storage layer.

18. A computer program product for a distributed data storage layer for web data services, the computer program product being embodied in a tangible computer readable storage medium and comprising computer instructions for:

storing web related data for a plurality of merchants in the distributed data storage layer; and

copying on demand a data store and a search index for web related data for a first merchant to perform a job for updating the web related data for the first merchant.

19. The computer program product recited in claim 18, wherein the copy on demand of the data store and the search index for storing the web related data for the first merchant corresponds to a snapshot of the data store and the search index in a dynamic cluster.

20. The computer program product recited in claim 18, wherein the web related data for the plurality of merchants is stored in a back-end cluster of the distributed data storage layer.

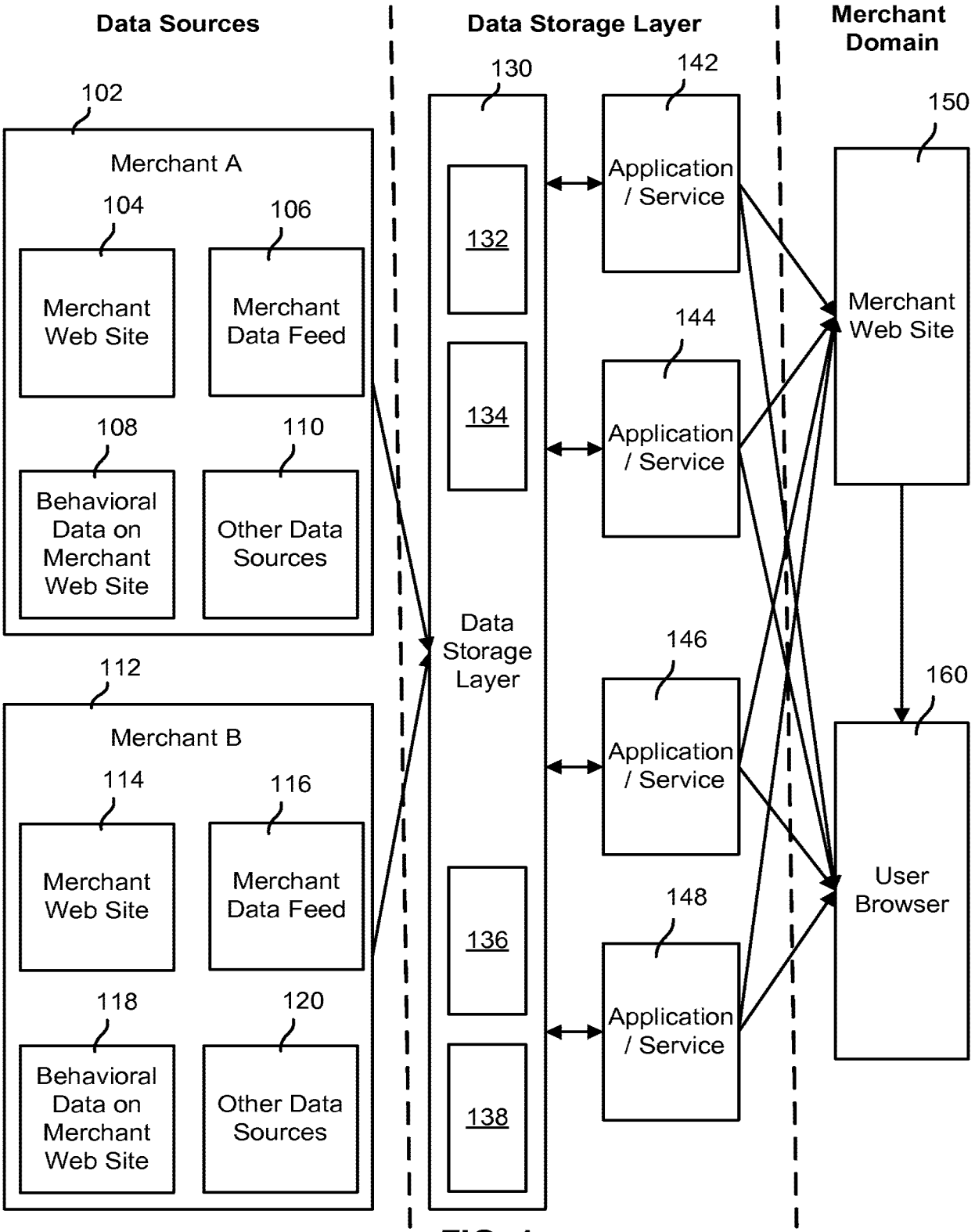

Data Sources
Data Storage Layer
Merchant Domain
102
Merchant A
104
Merchant Web Site
106
Merchant Data Feed
108
Behavioral Data on Merchant Web Site
110
Other Data Sources
112
Merchant B
114
Merchant Web Site
116
Merchant Data Feed
118
Behavioral Data on Merchant Web Site
120
Other Data Sources
Data Storage Layer
130
132
134
136
138
142
Application / Service
144
Application / Service
146
Application / Service
148
Application / Service
150
Merchant Web Site
160
User Browser

FIG. 1

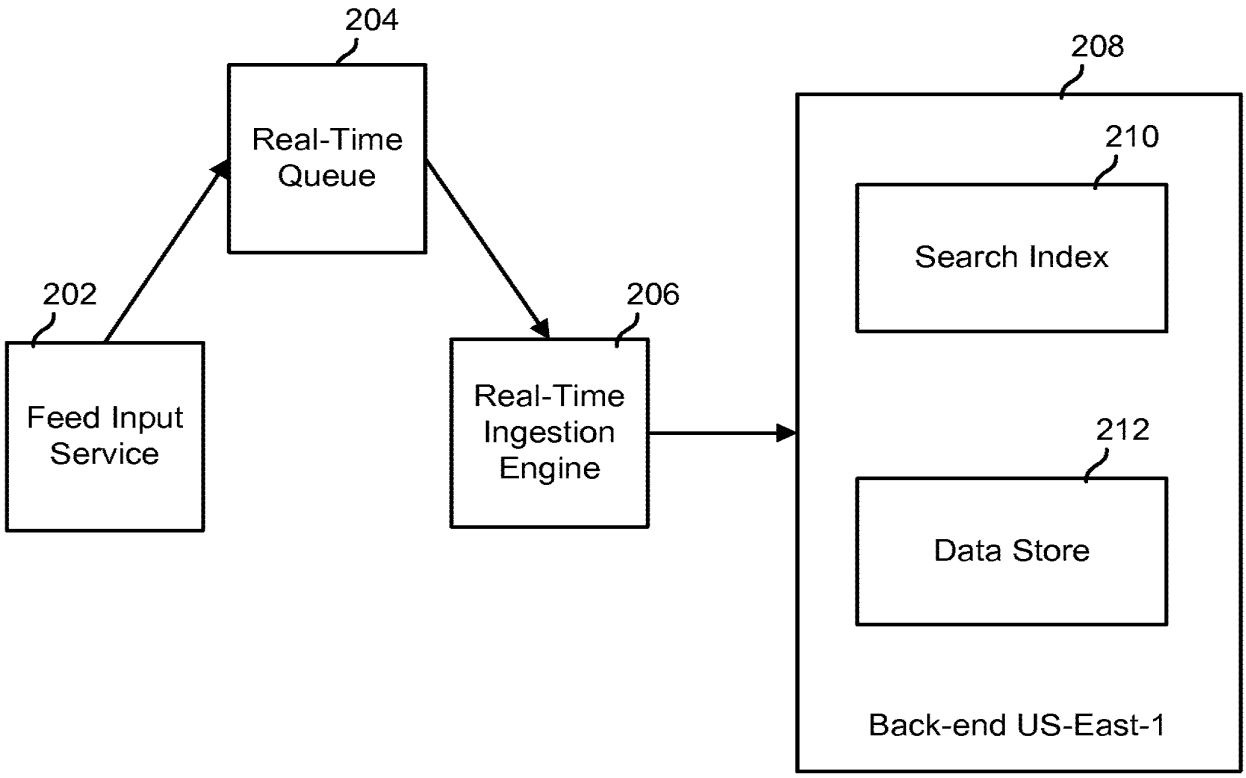

204
Real-Time Queue
208
210
Search Index
202
Feed Input Service
206
Real-Time Ingestion Engine
212
Data Store
Back-end US-East-1

FIG. 2

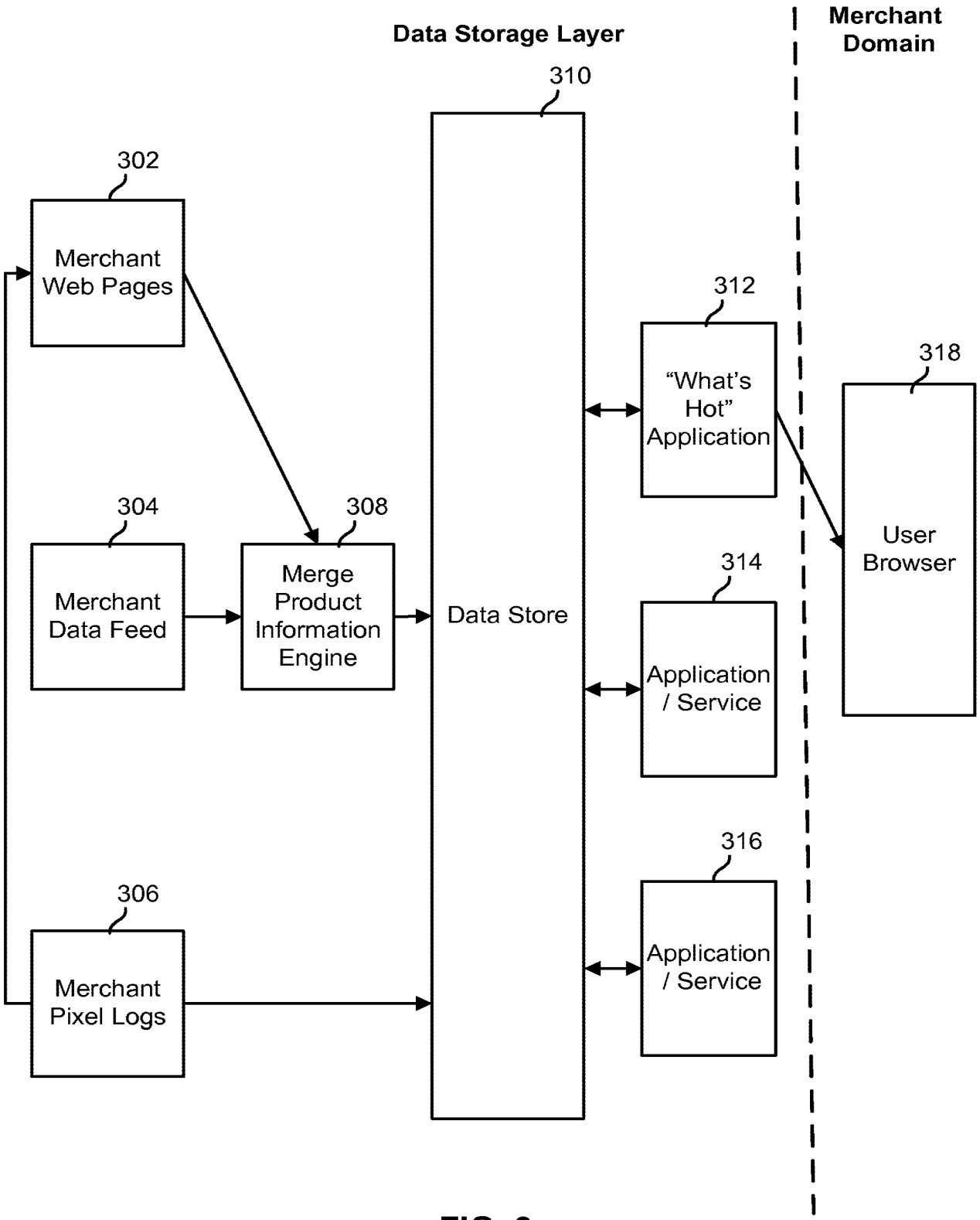

Data Storage Layer
Merchant Domain
310
302
Merchant Web Pages
312
318
"What's Hot" Application
User Browser
304
308
Merchant Data Feed
Merge Product Information Engine
Data Store
314
Application / Service
316
306
Application / Service
Merchant Pixel Logs

FIG. 3

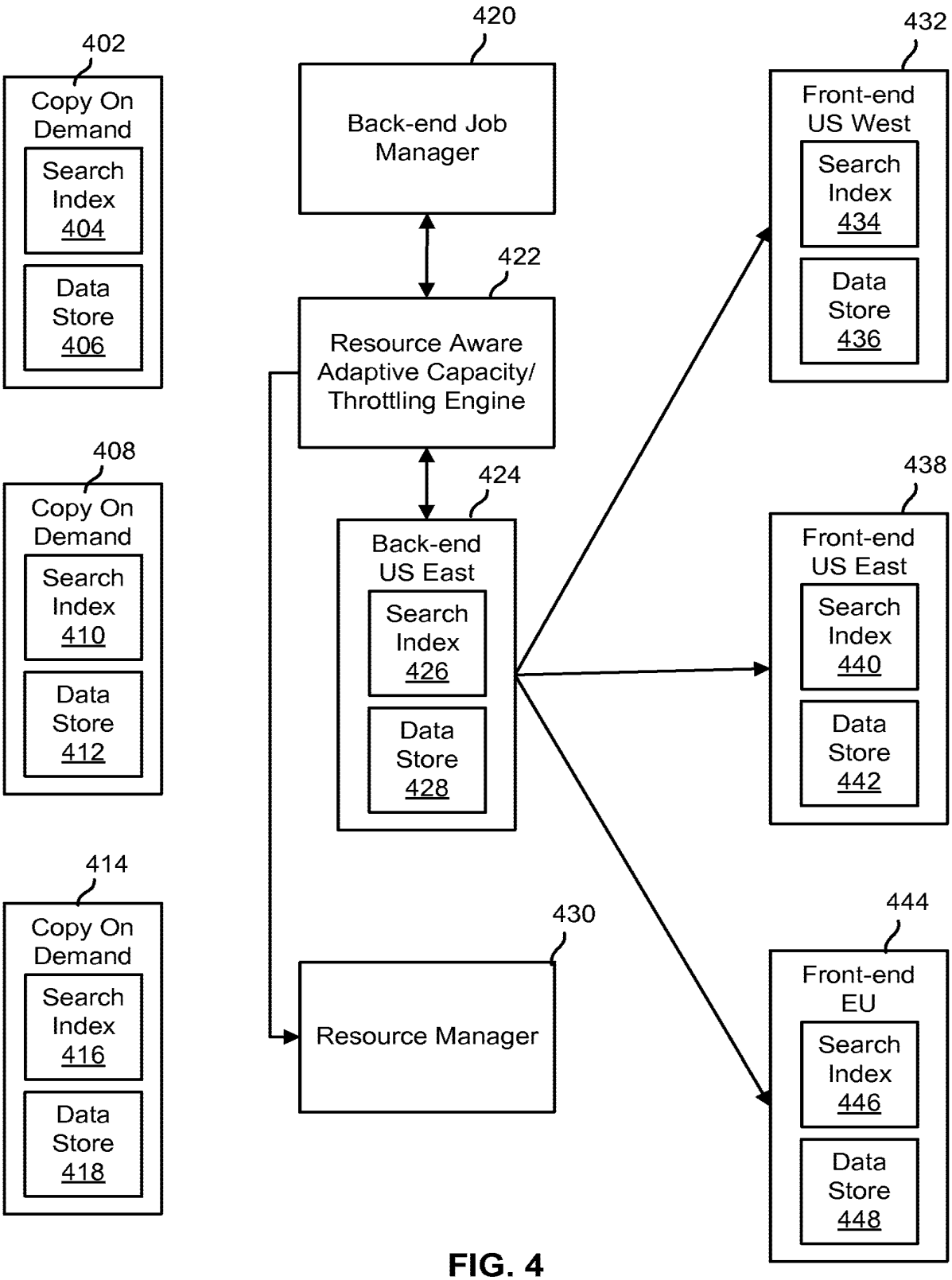

402
Copy On Demand
Search Index 404
Data Store 406
408
Copy On Demand
Search Index 410
Data Store 412
414
Copy On Demand
Search Index 416
Data Store 418
420
Back-end Job Manager
422
Resource Aware Adaptive Capacity/ Throttling Engine
424
Back-end US East
Search Index 426
Data Store 428
430
Resource Manager
432
Front-end US West
Search Index 434
Data Store 436
438
Front-end US East
Search Index 440
Data Store 442
444
Front-end EU
Search Index 446
Data Store 448

FIG. 4

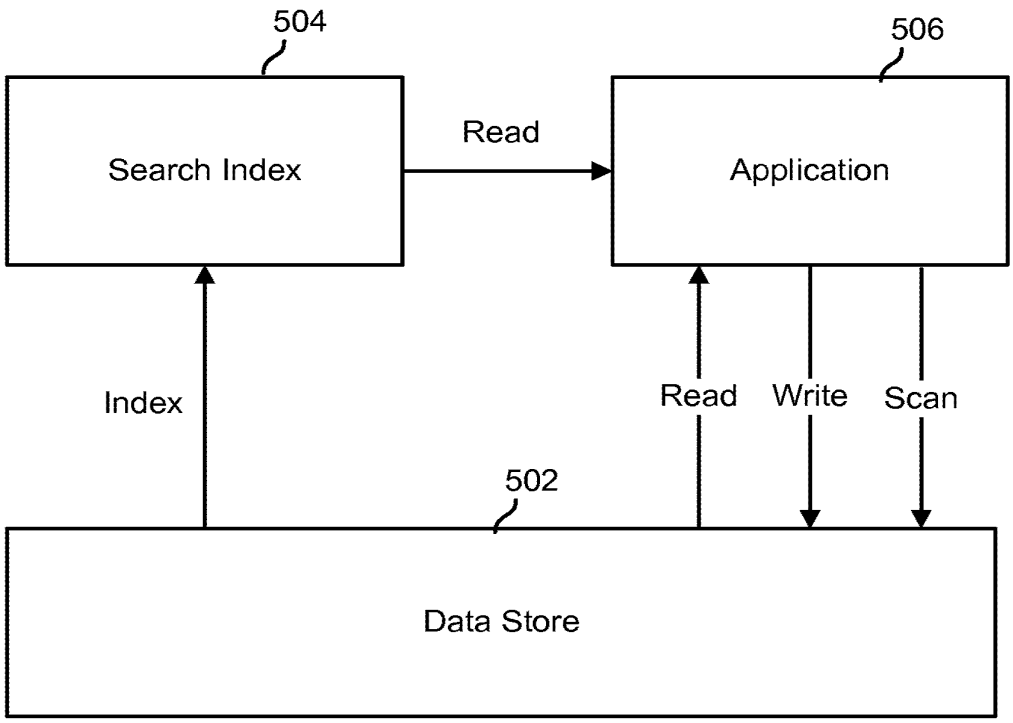

504
506
Search Index
Read
Application
Index
Read
Write
Scan
502
Data Store

FIG. 5

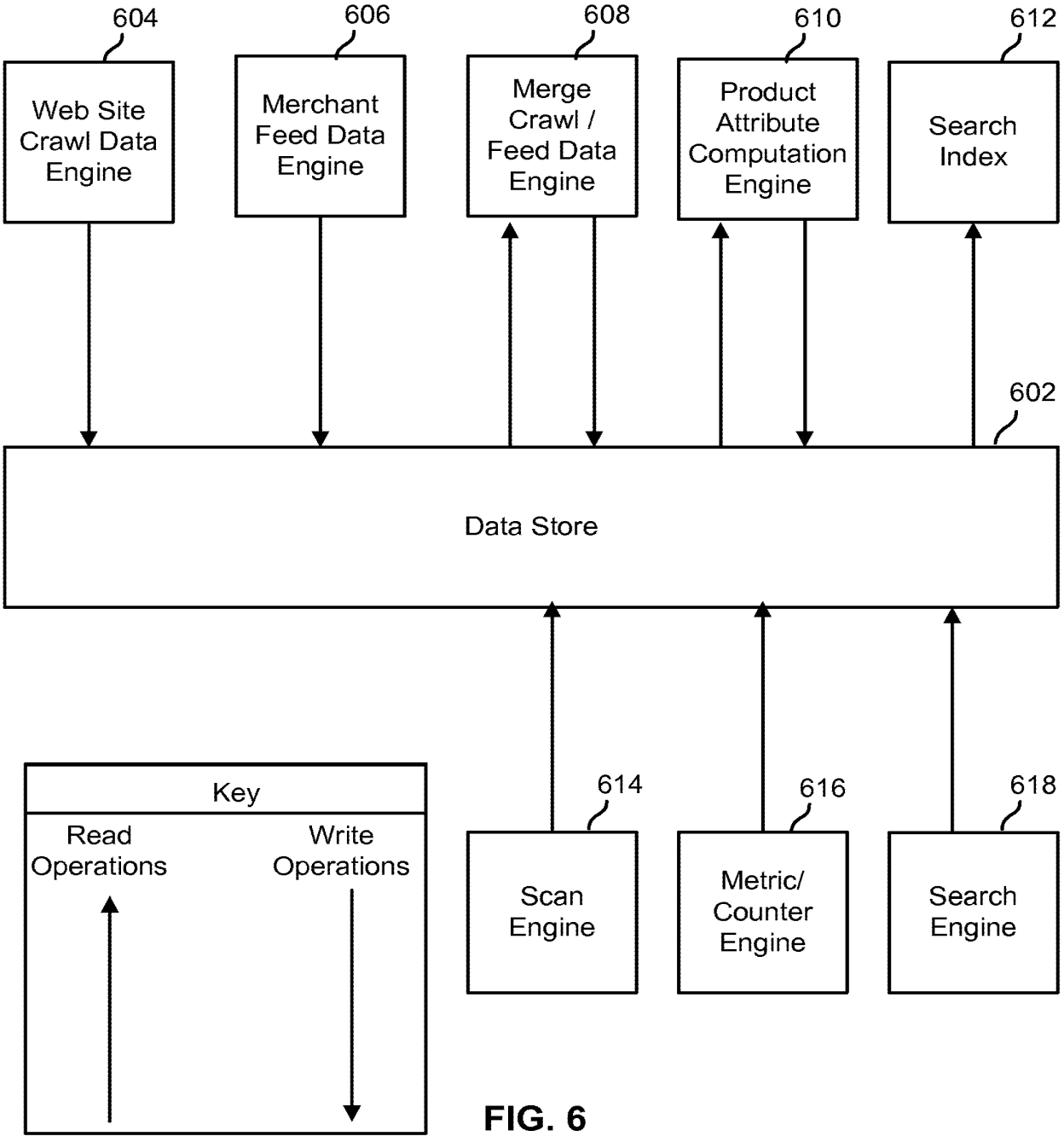

604
606
608
610
612
Web Site Crawl Data Engine
Merchant Feed Data Engine
Merge Crawl / Feed Data Engine
Product Attribute Computation Engine
Search Index
602
Data Store
Key
Read Operations
Write Operations
614
616
618
Scan Engine
Metric/ Counter Engine
Search Engine

FIG. 6

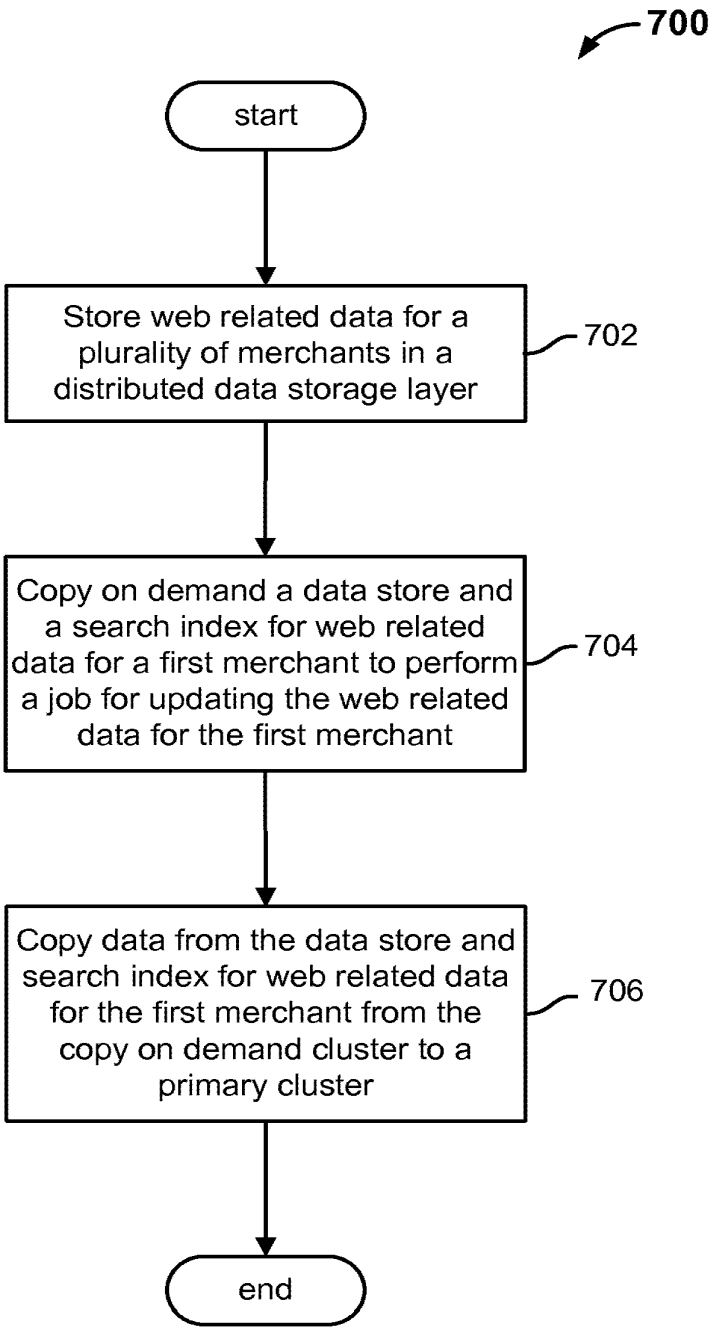

700
start
Store web related data for a plurality of merchants in a distributed data storage layer
702
Copy on demand a data store and a search index for web related data for a first merchant to perform a job for updating the web related data for the first merchant
704
Copy data from the data store and search index for web related data for the first merchant from the copy on demand cluster to a primary cluster
706
end

FIG. 7

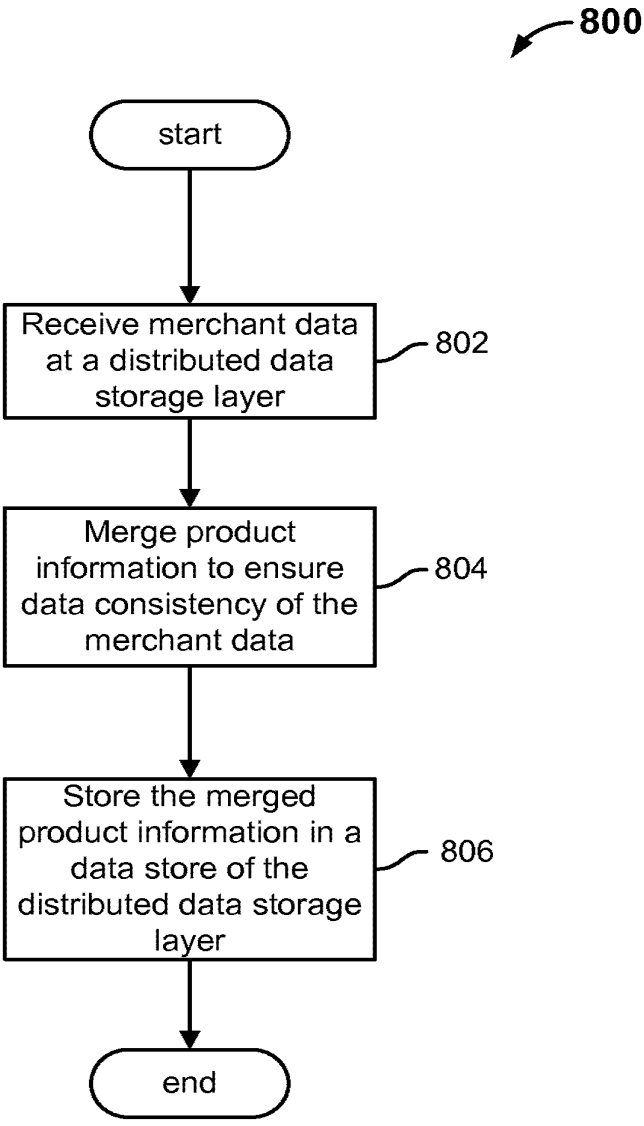

800
start
Receive merchant data at a distributed data storage layer
802
Merge product information to ensure data consistency of the merchant data
804
Store the merged product information in a data store of the distributed data storage layer
806
end

FIG. 8

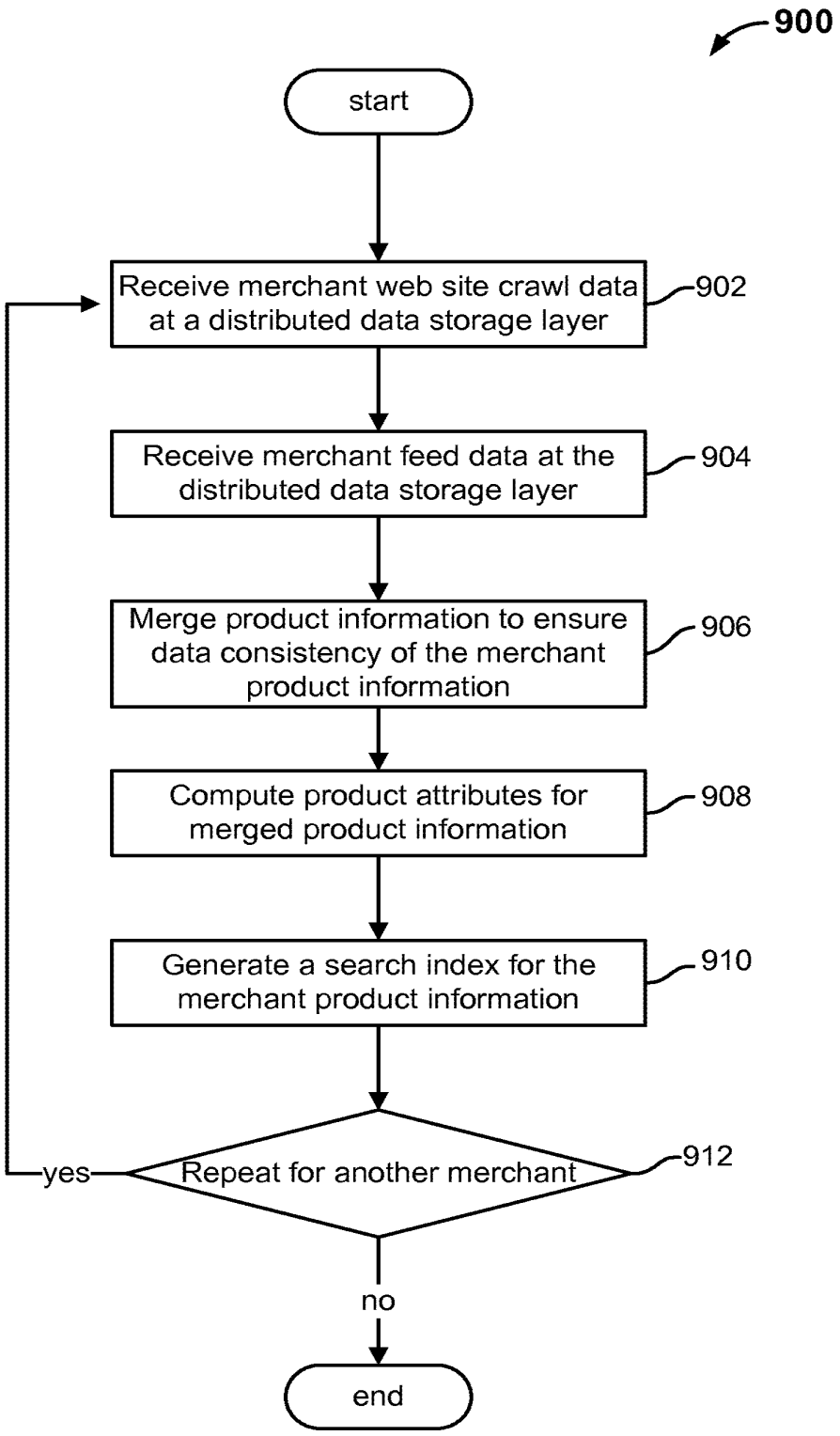

900
start
Receive merchant web site crawl data at a distributed data storage layer
902
Receive merchant feed data at the distributed data storage layer
904
Merge product information to ensure data consistency of the merchant product information
906
Compute product attributes for merged product information
908
Generate a search index for the merchant product information
910
Repeat for another merchant
912
yes
no
end

FIG. 9

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2014/070131

A. CLASSIFICATION OF SUBJECT MATTER
IPC(8) - H04L 12/28 (2015.01)
CPC - H04L 67/1097 (2014.12)
According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC(8) - G06F 12/00, 15/16, 15/173, 17/30, 3/048, H04L 9/00, 12/28, A61B 5/04 (2015.01)
USPC - 711/118, 709/226, 229, 246, 707/10, 203, 781, 715/784, 600/544, 706/46, 370/255

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
CPC - G06F 17/30289, 3/067, 3/06, H04L 67/1097, 41/042, 29/08549 (2014.12) (keyword delimited)

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

Orbit, Google Patents, Google Scholar
Search terms used: distributed data storage, web related, web related, merchant, vendor, product information, web site data, cluster size

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X – Y	US 2013/0304761 A1 (REDLICH et al.) 14 November 2013 (14.11.2013) entire document	1-12, 15-20 ---- 13, 14
Y	US 2007/0294197 A1 (JACKSON) 20 December 2007 (20.12.2007) entire document	13, 14
A	US 6,557,076 B1 (COPELAND et al.) 29 April 2003 (29.04.2003) entire document	1-20
A	US 6,714,980 B1 (MARKSON et al.) 30 March 2004 (30.03.2004) entire document	1-20
A	US 2003/0149373 A1 (READE et al.) 07 August 2003 (07.08.2003) entire document	1-20
A	US 2005/0038833 A1 (COLRAIN et al.) 17 February 2005 (17.02.2005) entire document	1-20
A	US 2006/0168259 A1 (SPILOTRO et al.) 27 July 2006 (27.07.2006) entire document	1-20

☐ Further documents are listed in the continuation of Box C. ☐

* Special categories of cited documents:
"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed
"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
03 February 2015	17 MAR 2015
Name and mailing address of the ISA/US Mail Stop PCT, Attn: ISA/US, Commissioner for Patents P.O. Box 1450, Alexandria, Virginia 22313-1450 Facsimile No. 571-273-3201	Authorized officer: Blaine R. Copenheaver PCT Helpdesk: 571-272-4300 PCT OSP: 571-272-7774

Form PCT/ISA/210 (second sheet) (July 2009)

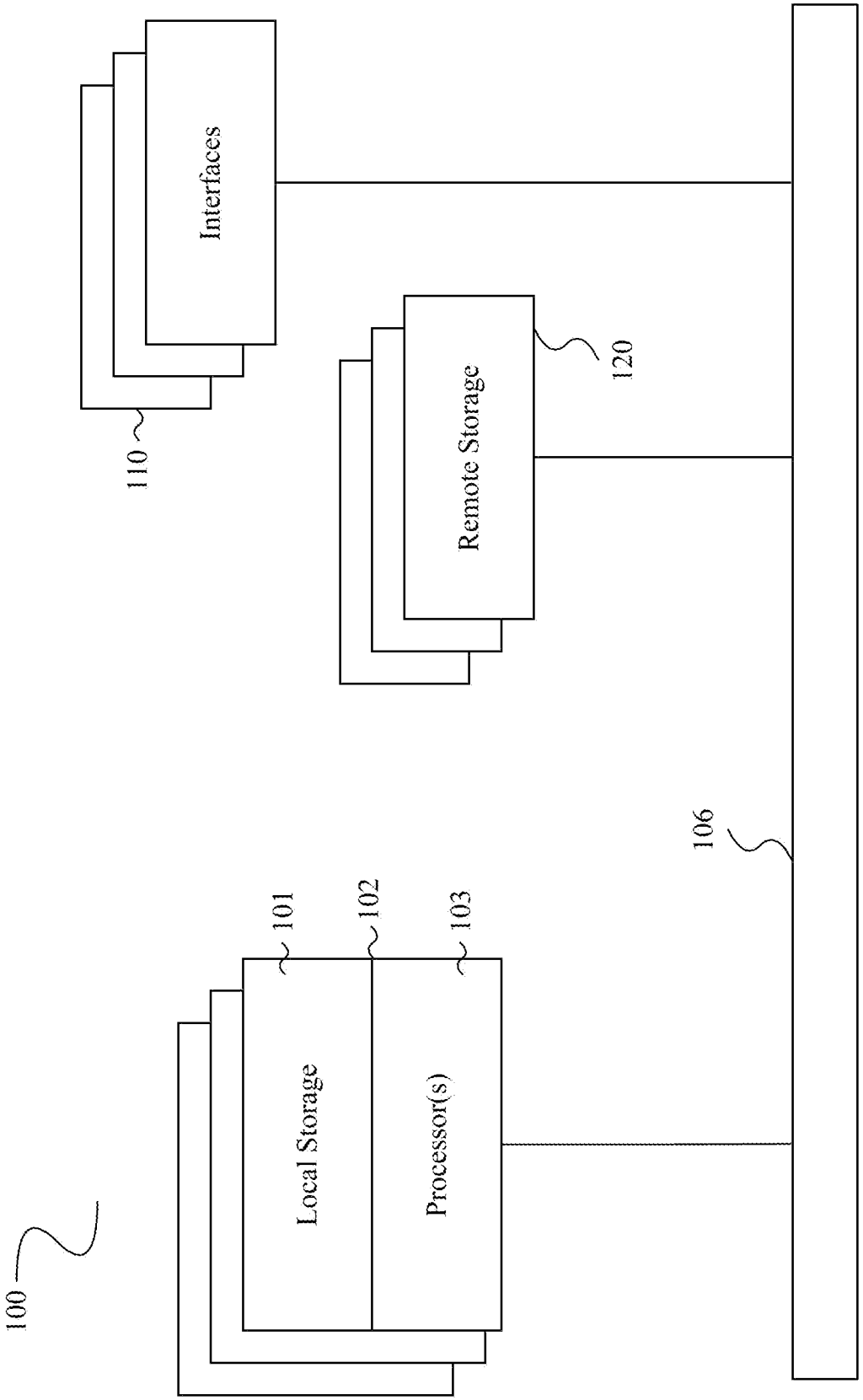

100
Local Storage
Processor(s)
101
102
103
106
Remote Storage
120
Interfaces
110

Fig. 1

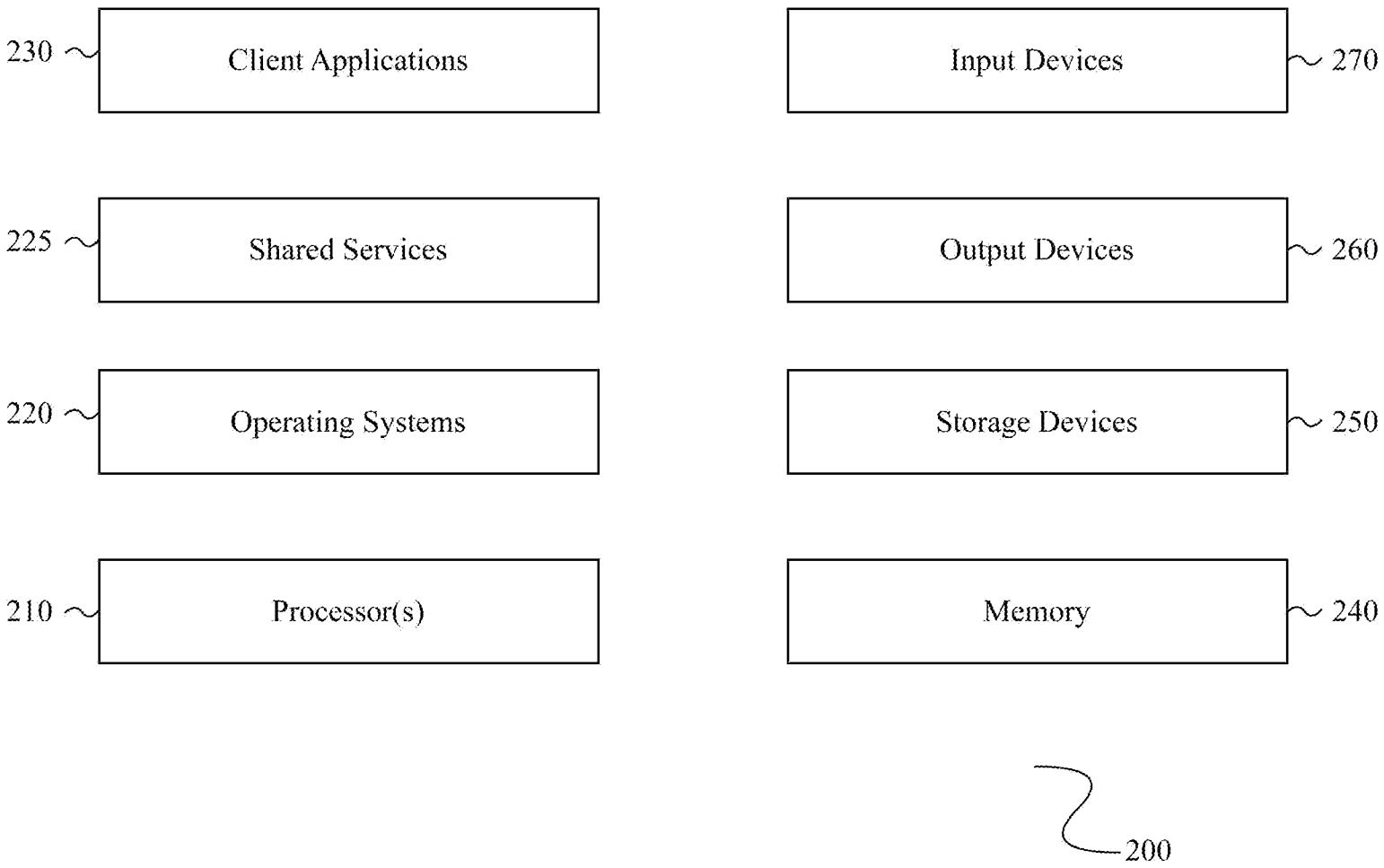

230
Client Applications
225
Shared Services
220
Operating Systems
210
Processor(s)
Input Devices
270
Output Devices
260
Storage Devices
250
Memory
240
200

Fig. 2

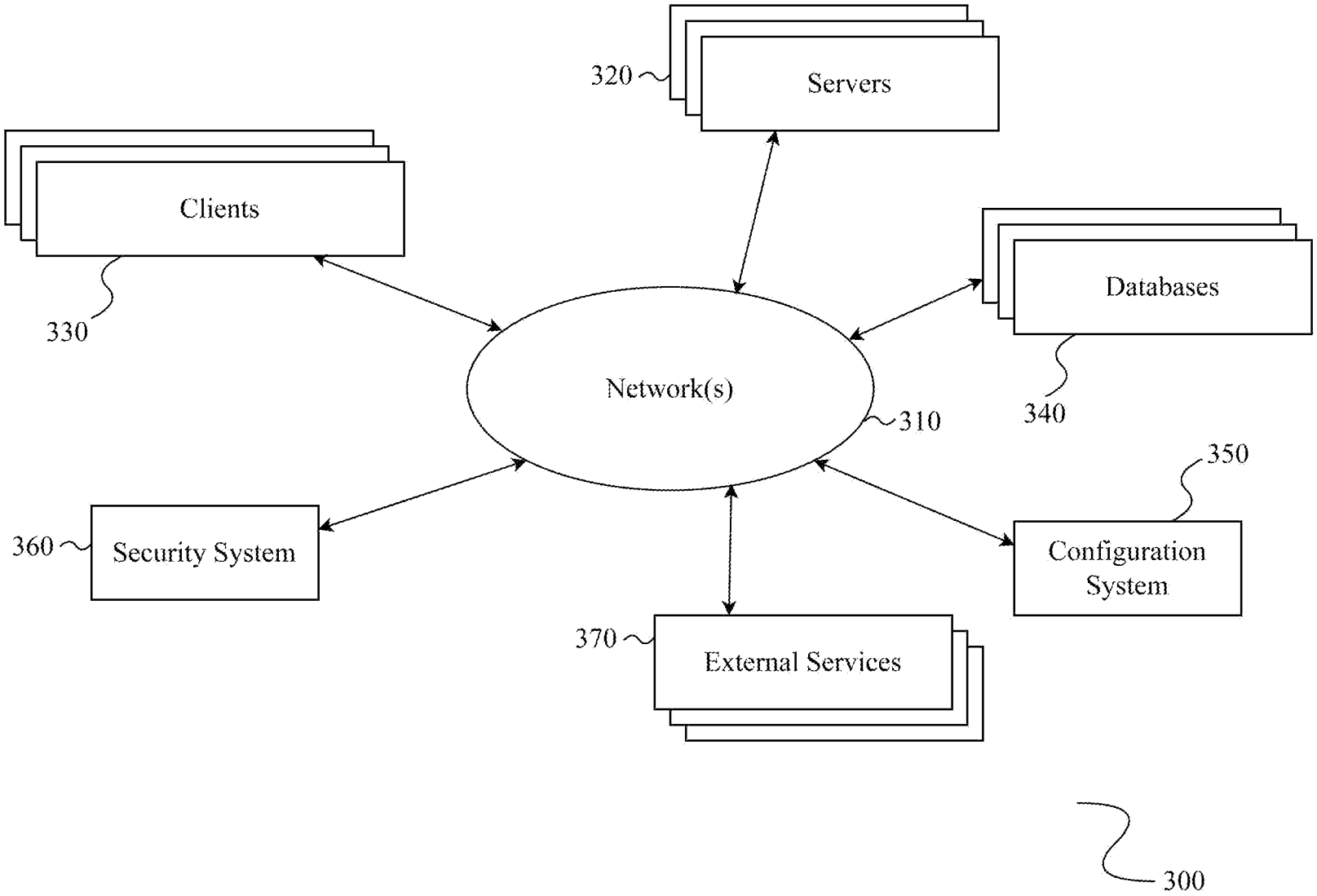

320
Servers
Clients
330
Network(s)
310
Databases
340
350
Configuration System
360
Security System
370
External Services
300

Fig. 3

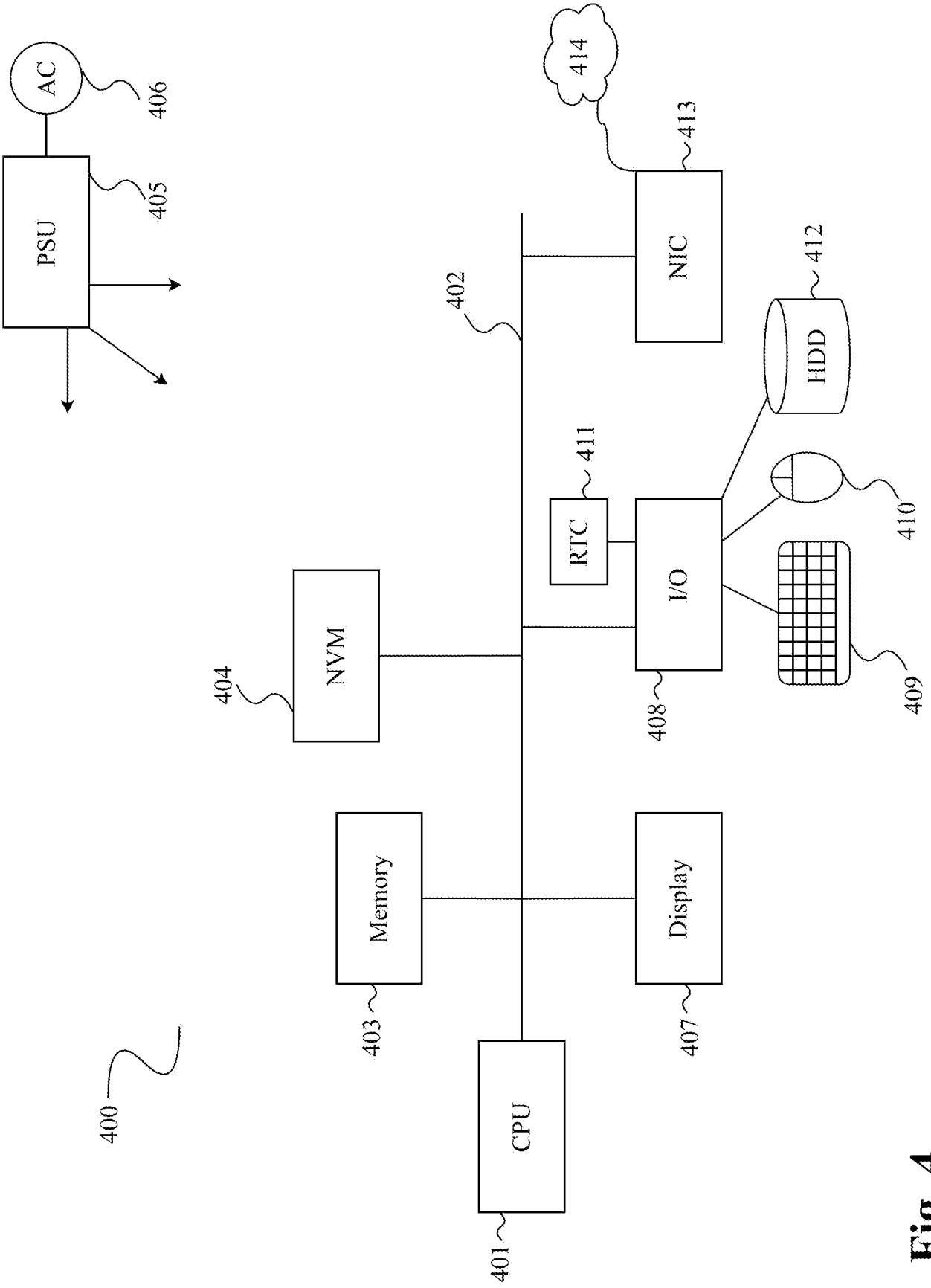

400
AC
406
PSU
405
414
413
NIC
402
412
HDD
411
RTC
I/O
410
409
404
NVM
408
403
Memory
Display
407
CPU
401

Fig. 4

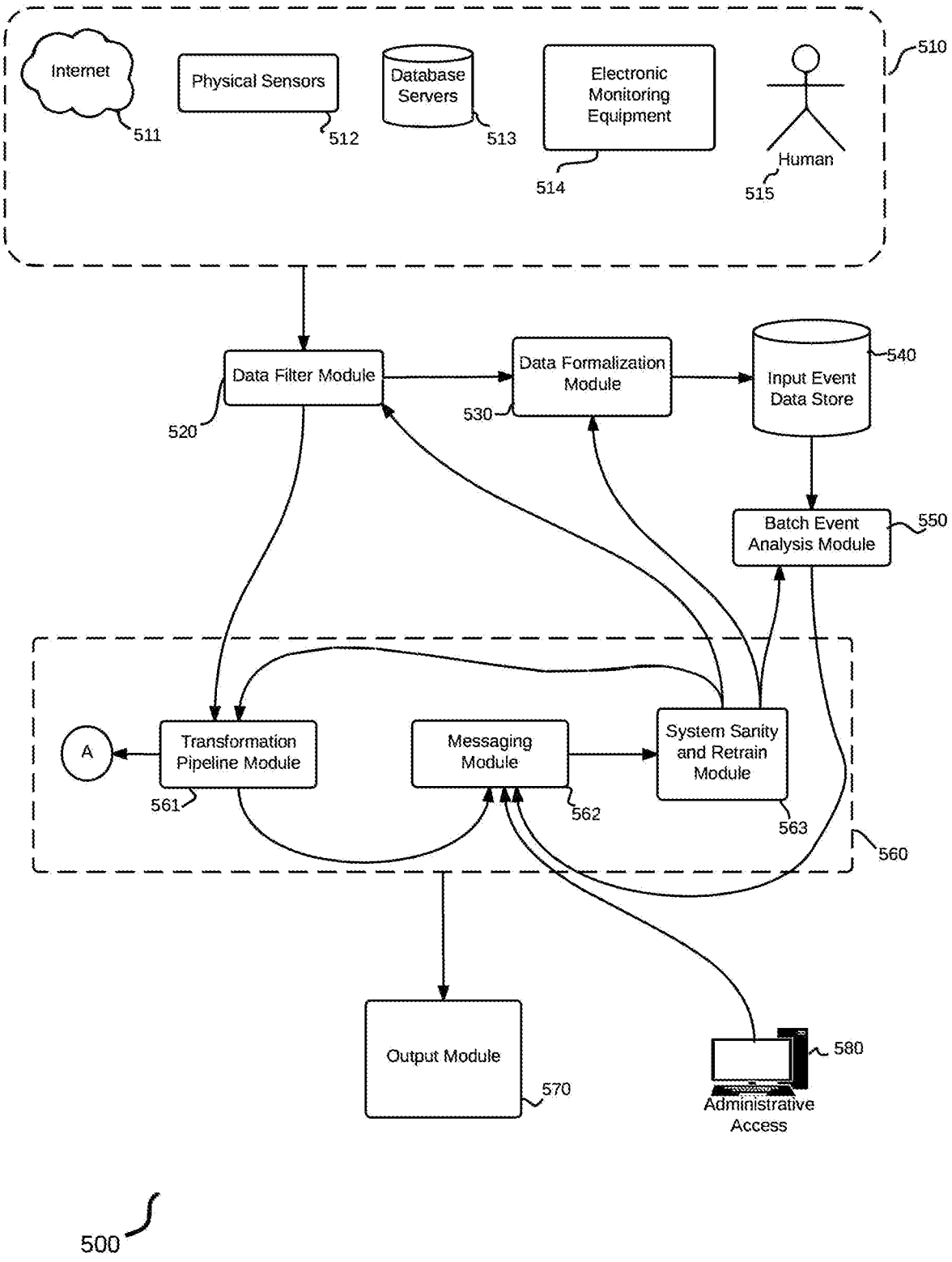
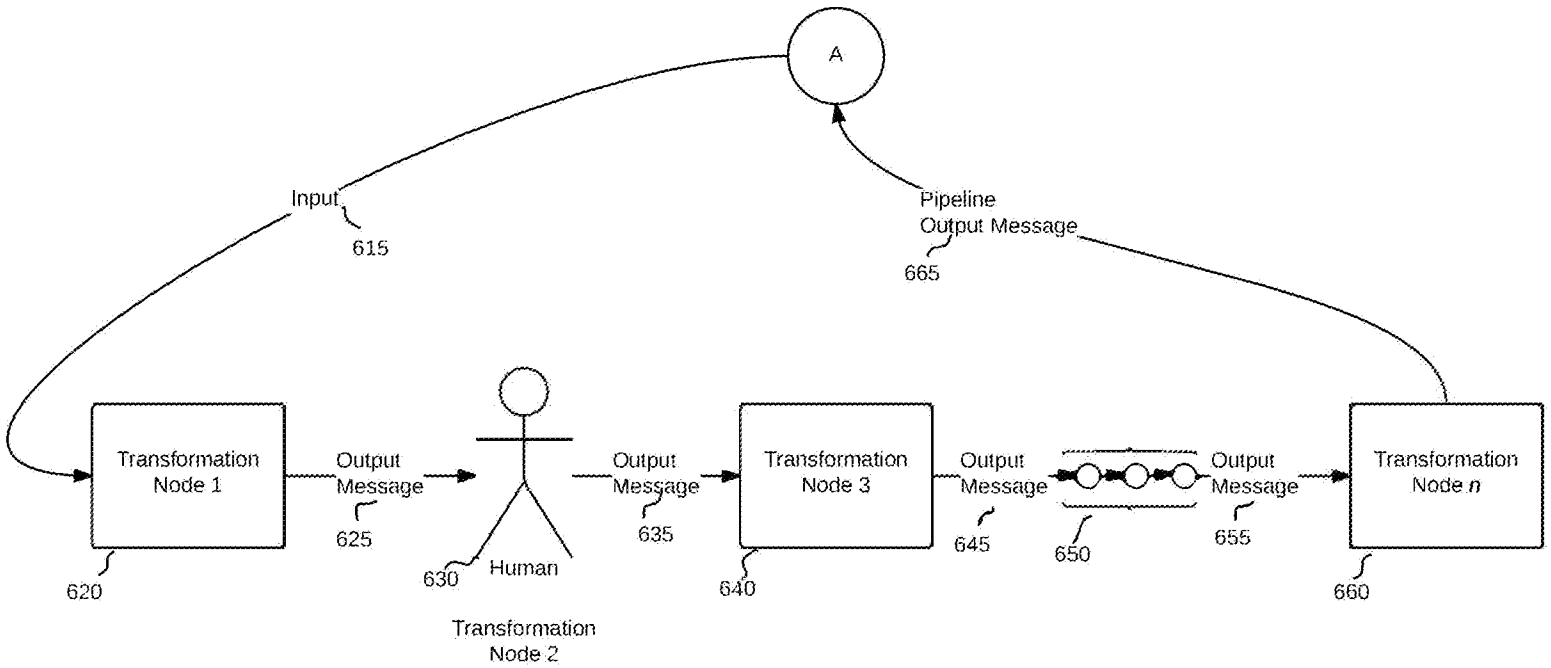

Internet
511
Physical Sensors
512
Database Servers
513
Electronic Monitoring Equipment
514
Human
515
510
Data Filter Module
520
Data Formalization Module
530
Input Event Data Store
540
Batch Event Analysis Module
550
A
Transformation Pipeline Module
561
Messaging Module
562
System Sanity and Retrain Module
563
560
Output Module
570
580
Administrative Access
500

Fig. 5



600

Fig. 6 (Prior Art)

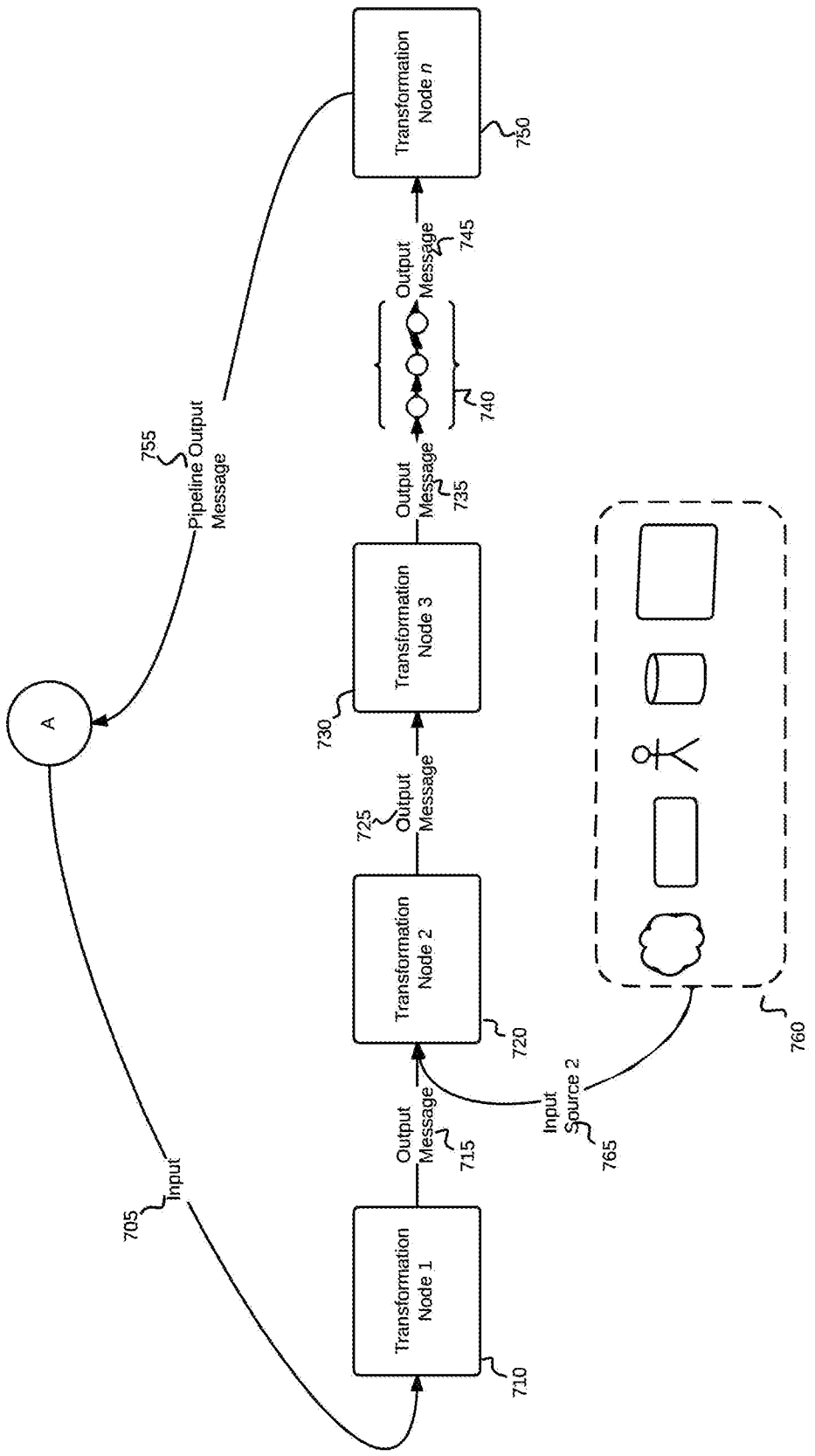

A
Input
705
Transformation Node 1
710
Output Message
715
Transformation Node 2
720
Output Message
725
Transformation Node 3
730
Output Message
735
740
Output Message
745
Transformation Node n
750
Pipeline Output Message
755
Input Source 2
765
760
700

Fig. 7

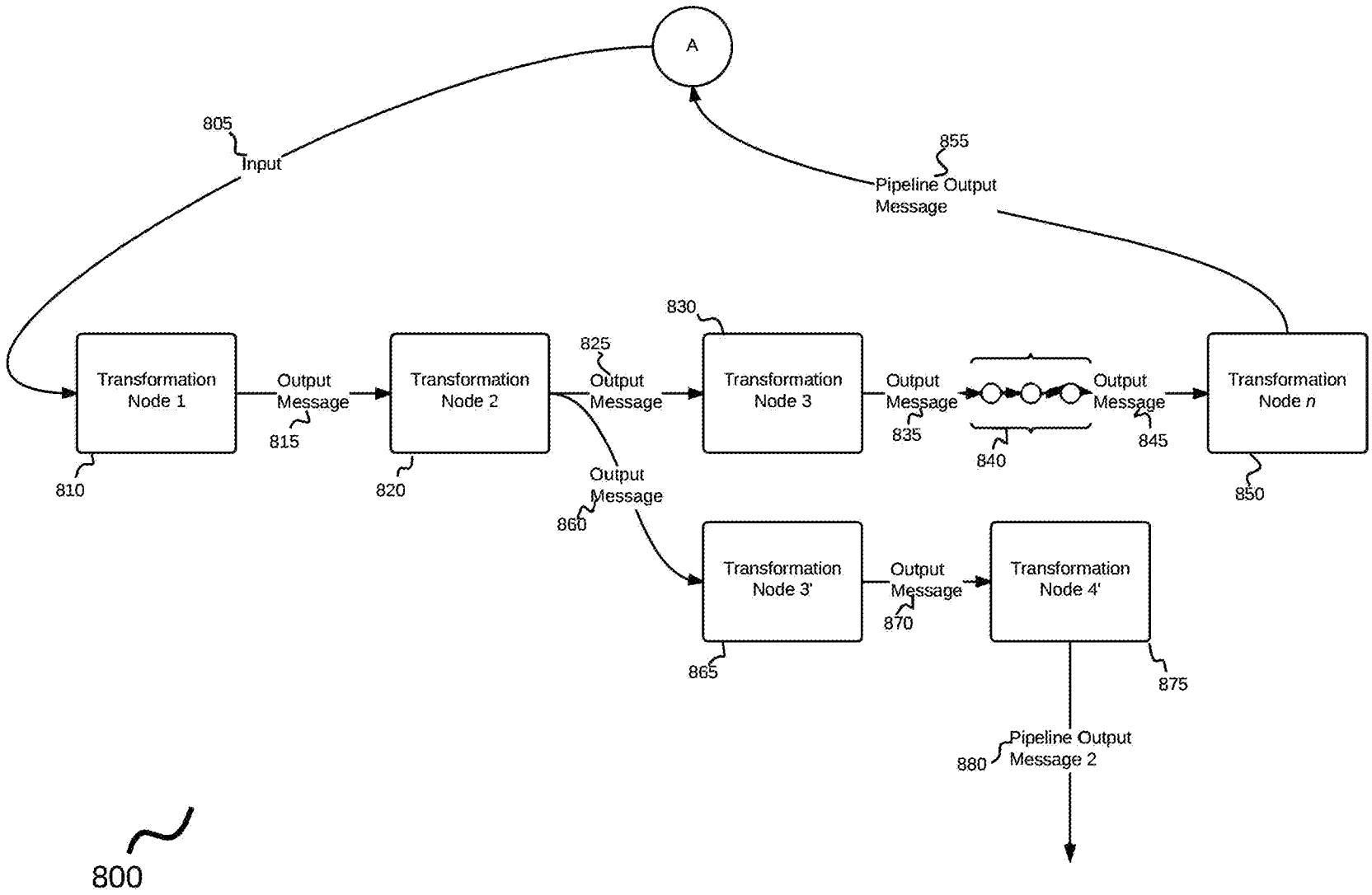
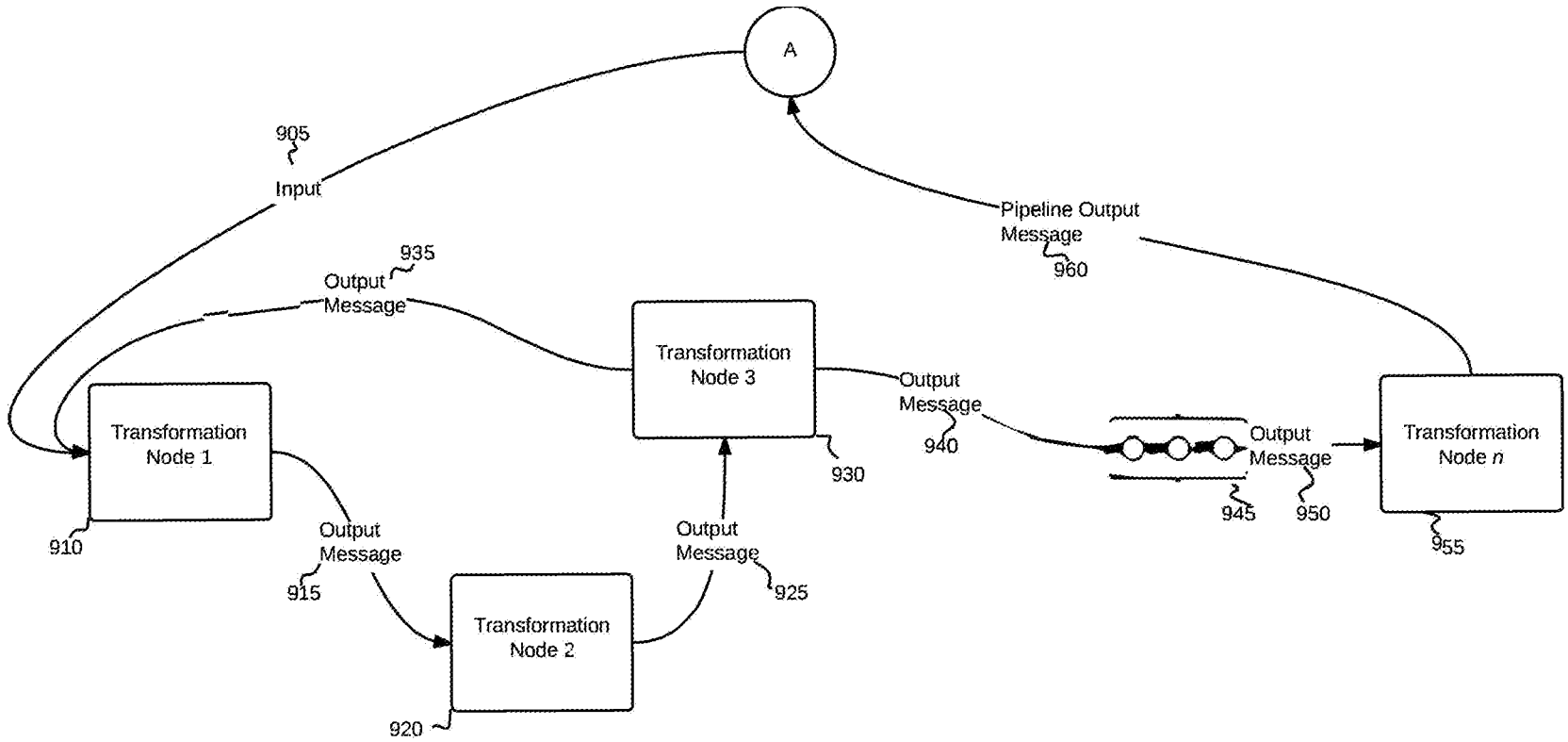

A
805
Input
855
Pipeline Output Message
Transformation Node 1
Output Message
815
Transformation Node 2
825
Output Message
830
Transformation Node 3
Output Message
835
840
Output Message
845
Transformation Node n
850
810
820
Output Message
860
Transformation Node 3'
Output Message
870
Transformation Node 4'
865
875
880
Pipeline Output Message 2
800

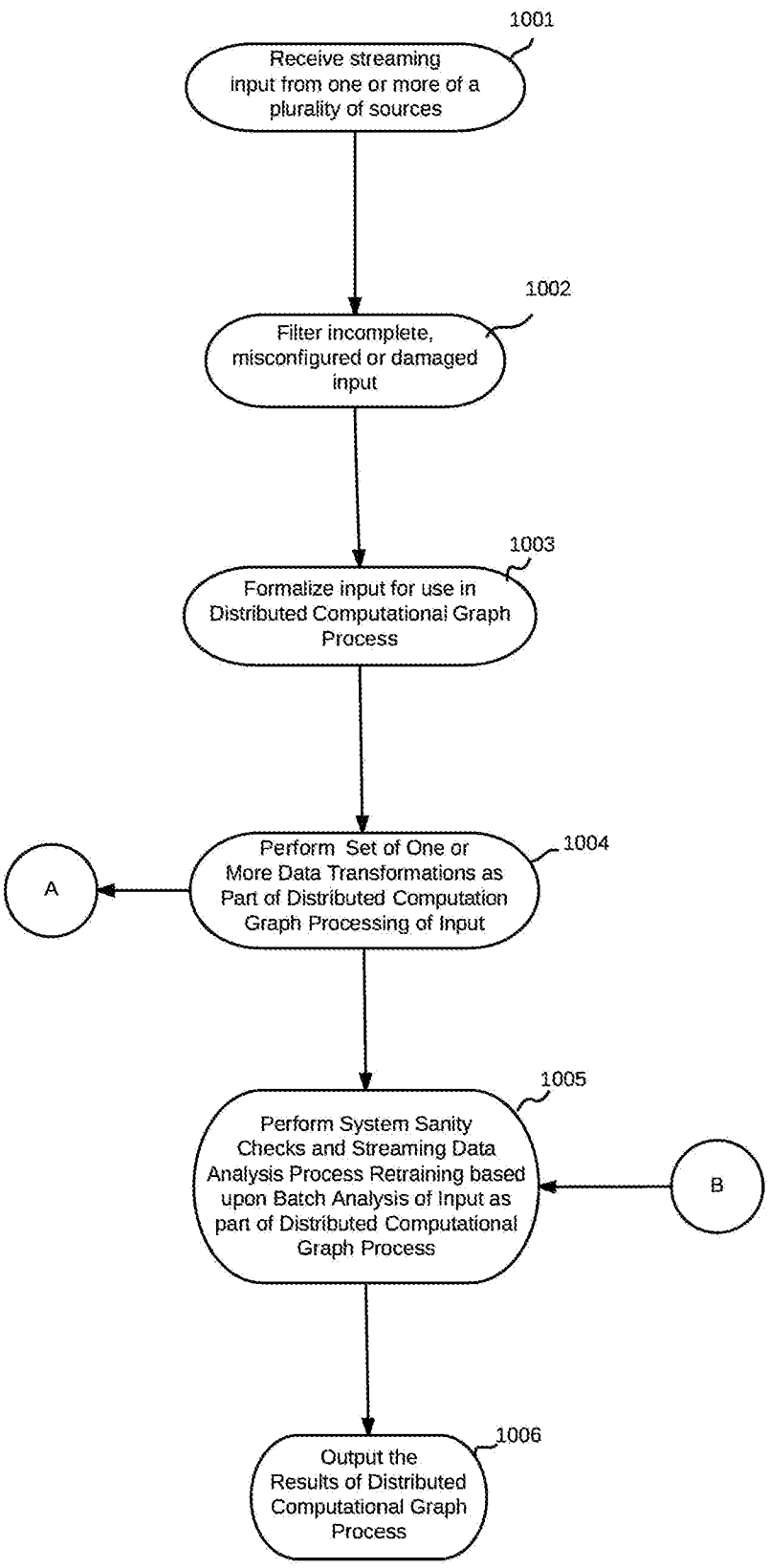
Fig. 8


A
905
Input
Pipeline Output Message
960
935
Output Message
Transformation Node 3
Output Message
940
930
Transformation Node 1
910
Output Message
915
Transformation Node 2
920
Output Message
925
Output Message
945
950
Transformation Node n
955

900

Fig. 9

Fig. 10


1000
1001
Receive streaming input from one or more of a plurality of sources
1002
Filter incomplete, misconfigured or damaged input
1003
Formalize input for use in Distributed Computational Graph Process
1004
Perform Set of One or More Data Transformations as Part of Distributed Computation Graph Processing of Input
A
1005
Perform System Sanity Checks and Streaming Data Analysis Process Retraining based upon Batch Analysis of Input as part of Distributed Computational Graph Process
B
1006
Output the Results of Distributed Computational Graph Process

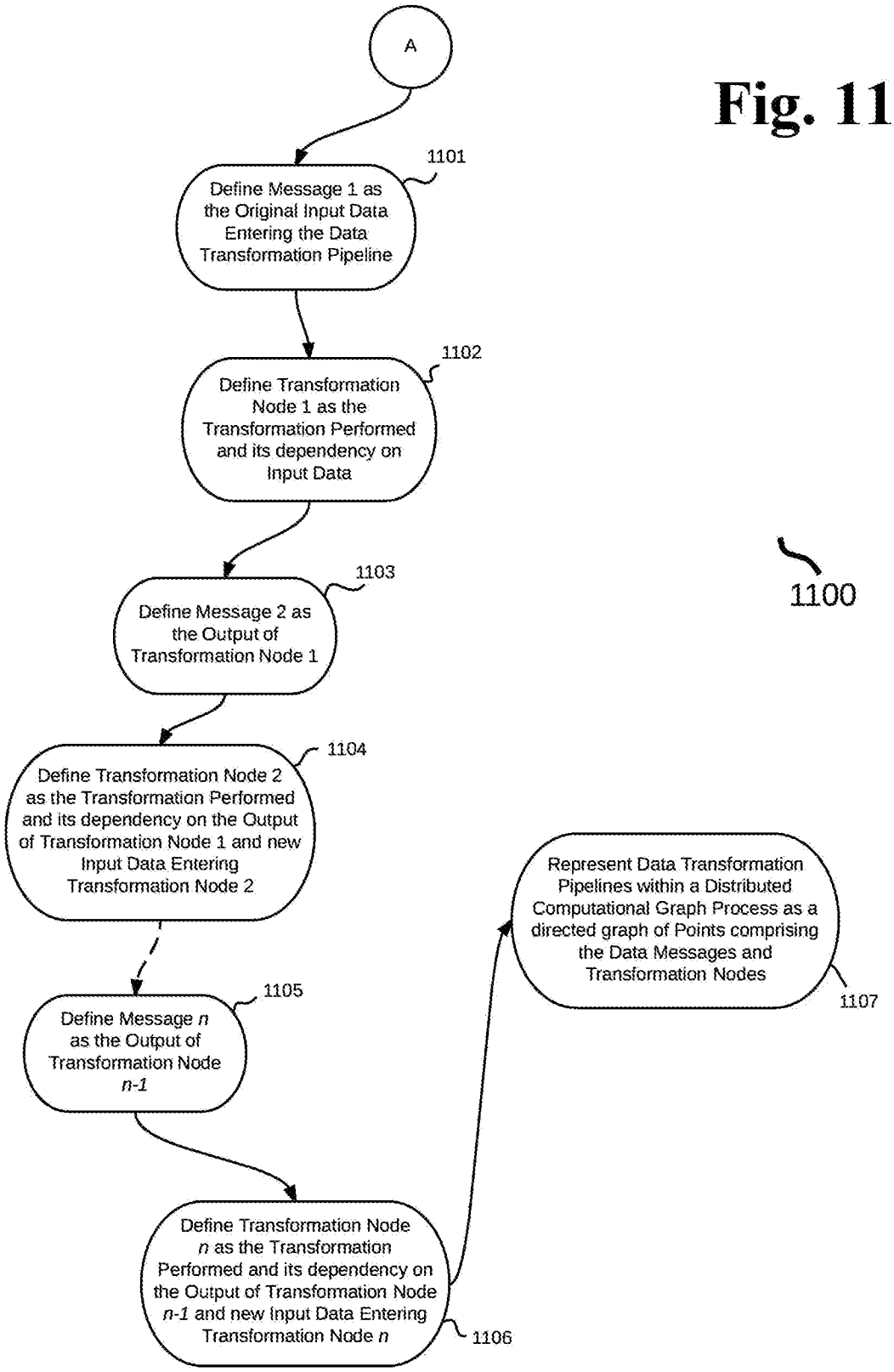

A
1101
Define Message 1 as the Original Input Data Entering the Data Transformation Pipeline
1102
Define Transformation Node 1 as the Transformation Performed and its dependency on Input Data
1103
Define Message 2 as the Output of Transformation Node 1
1104
Define Transformation Node 2 as the Transformation Performed and its dependency on the Output of Transformation Node 1 and new Input Data Entering Transformation Node 2
1105
Define Message *n* as the Output of Transformation Node *n-1*
Define Transformation Node *n* as the Transformation Performed and its dependency on the Output of Transformation Node *n-1* and new Input Data Entering Transformation Node *n*
1106
Represent Data Transformation Pipelines within a Distributed Computational Graph Process as a directed graph of Points comprising the Data Messages and Transformation Nodes
1107
1100

Fig. 11

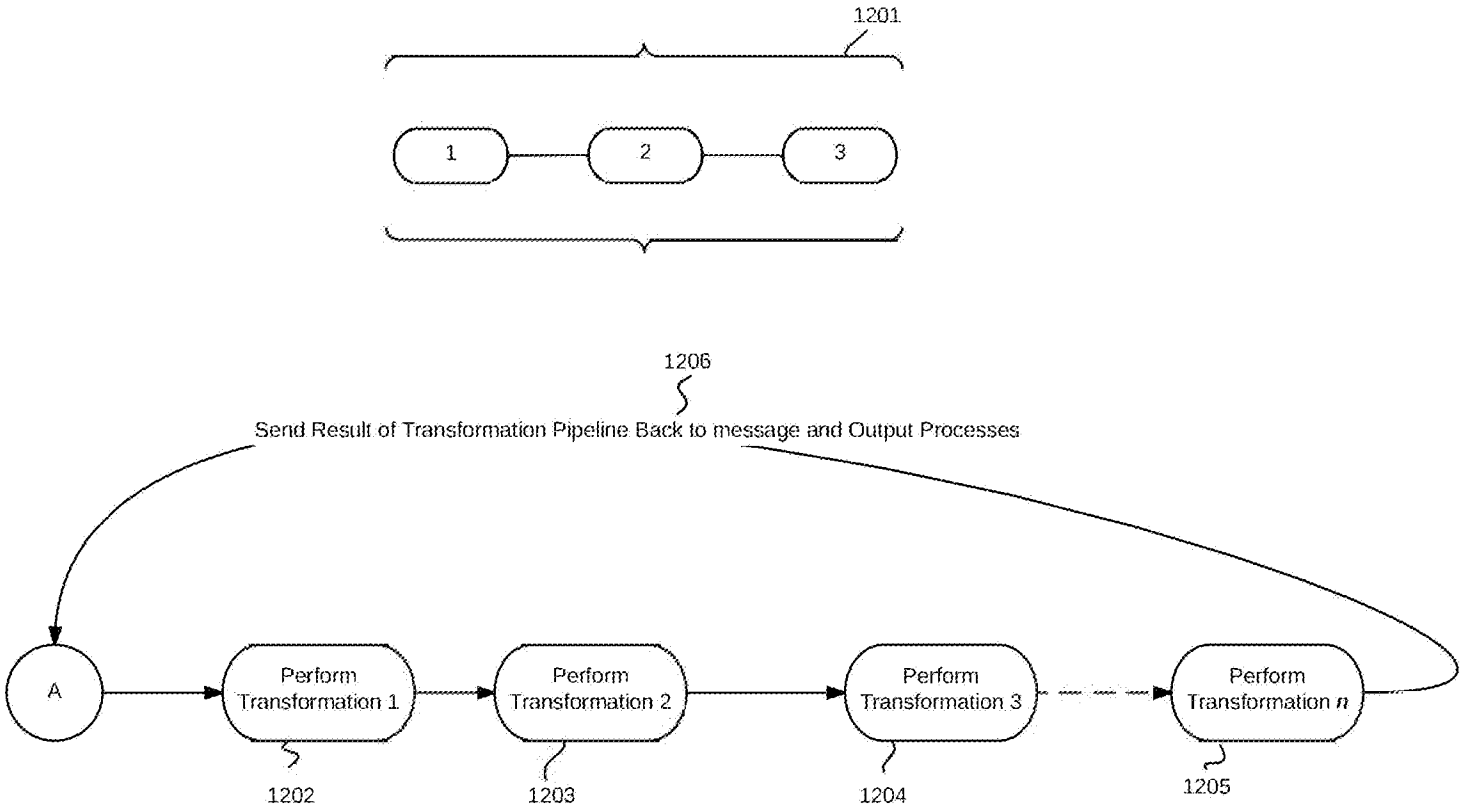


Fig. 12 (Prior Art)

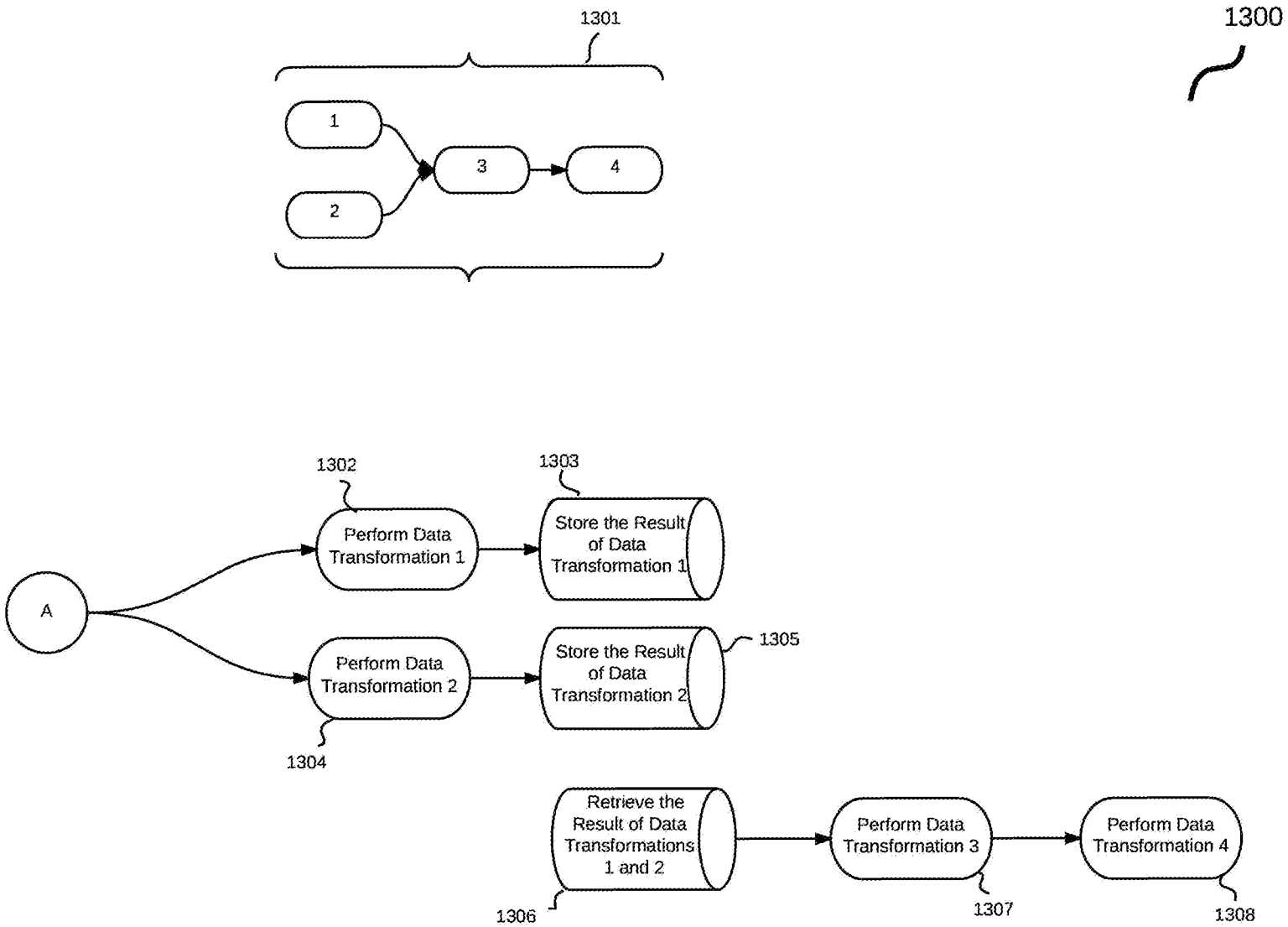

1300
1301
1
2
3
4
1302
1303
Perform Data Transformation 1
Store the Result of Data Transformation 1
A
Perform Data Transformation 2
Store the Result of Data Transformation 2
1305
1304
Retrieve the Result of Data Transformations 1 and 2
Perform Data Transformation 3
Perform Data Transformation 4
1306
1307
1308

Fig. 13

Fig. 14

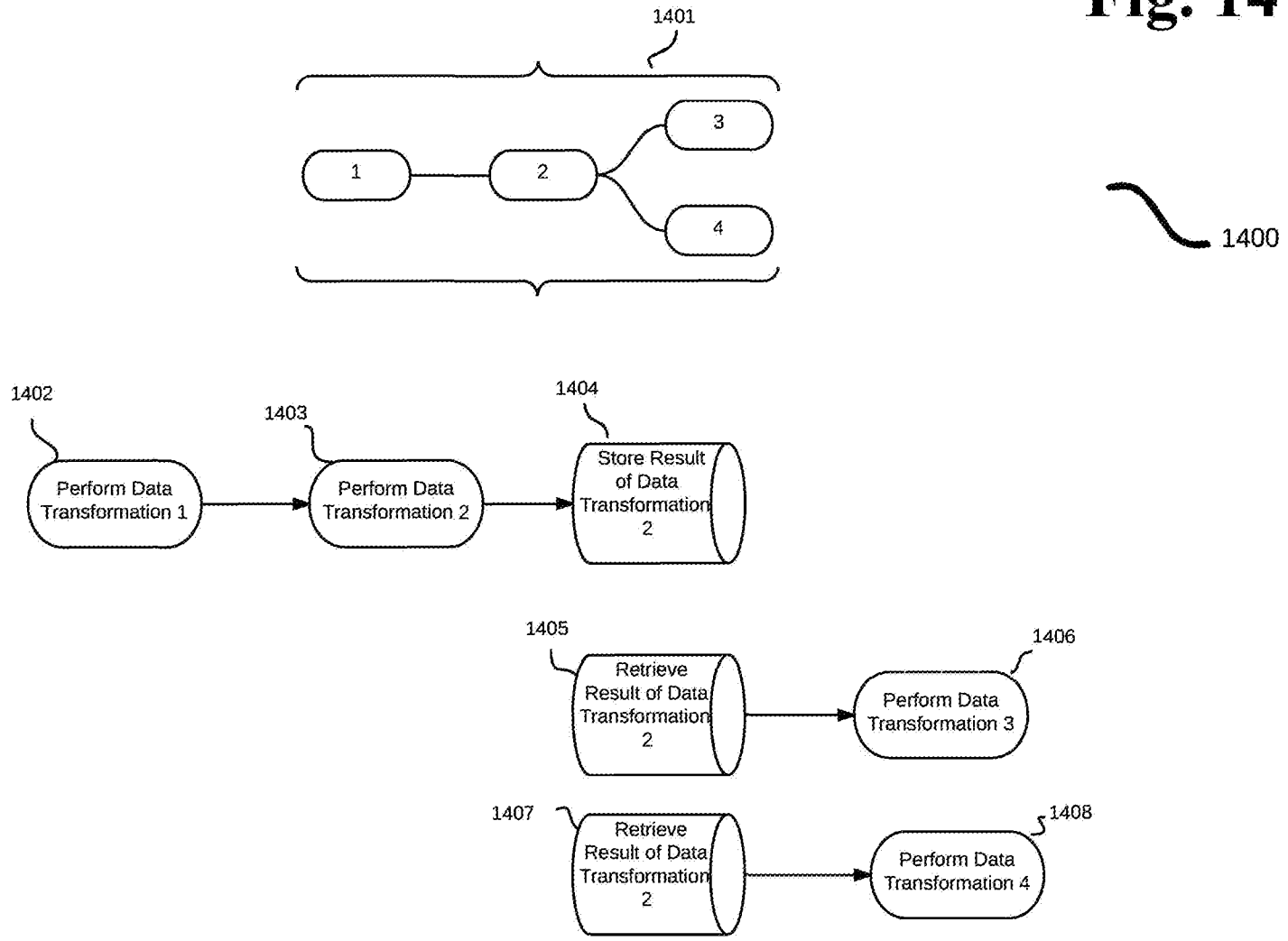

1401
1
2
3
4
1400
1402
1403
1404
Perform Data Transformation 1
Perform Data Transformation 2
Store Result of Data Transformation 2
1405
1406
Retrieve Result of Data Transformation 2
Perform Data Transformation 3
1407
1408
Retrieve Result of Data Transformation 2
Perform Data Transformation 4

Fig. 15

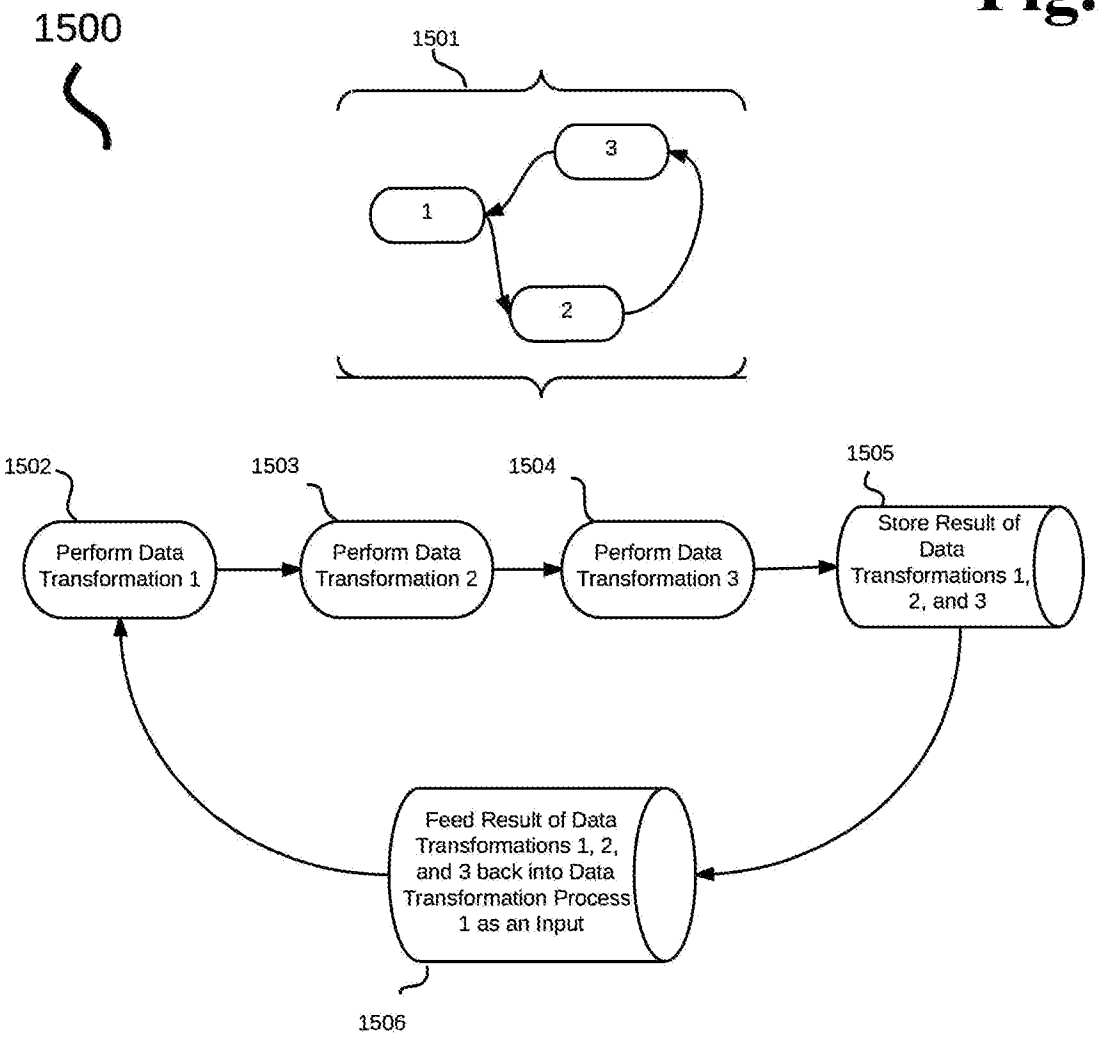
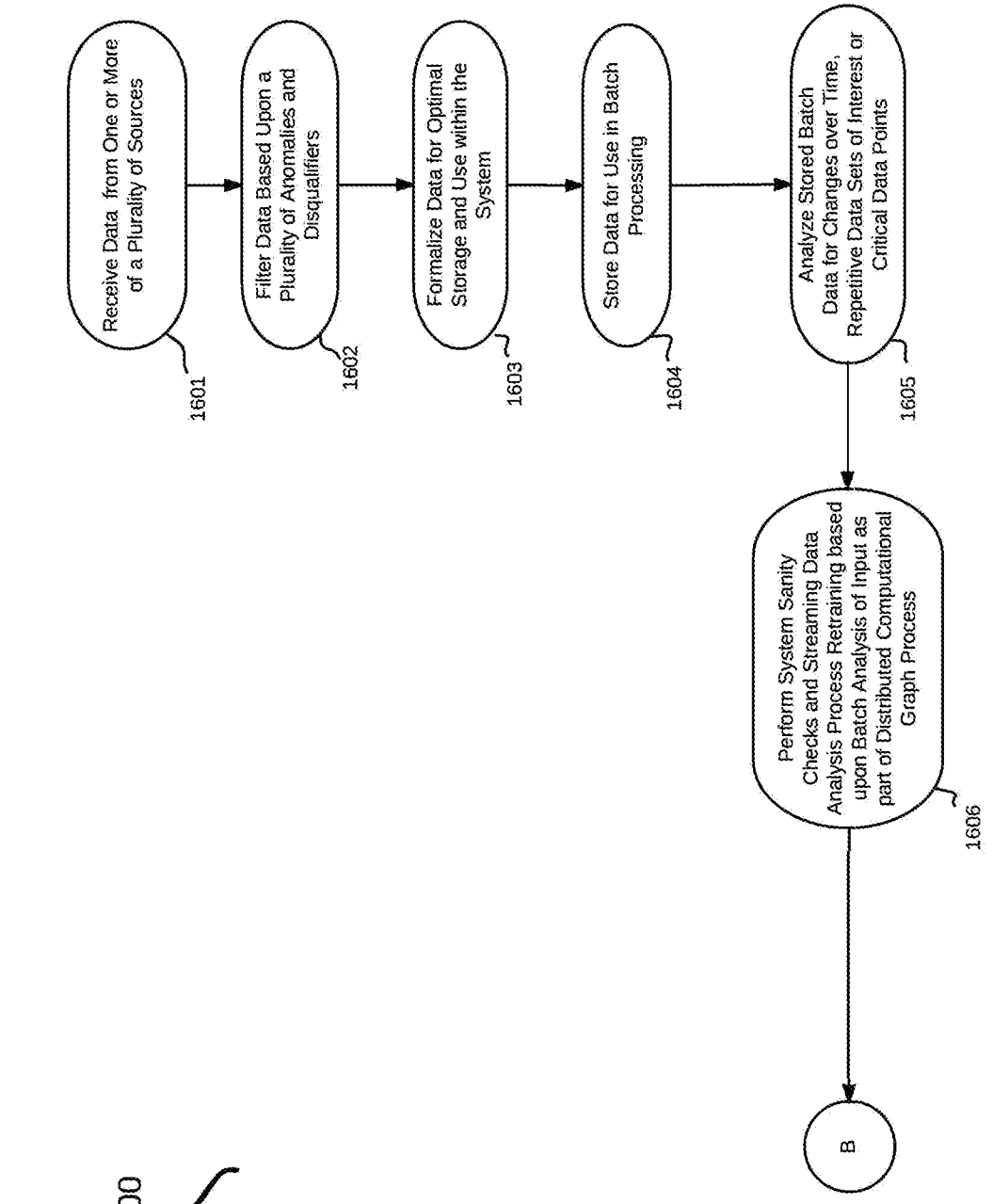

1500
1501
3
1
2
1502
1503
1504
1505
Perform Data Transformation 1
Perform Data Transformation 2
Perform Data Transformation 3
Store Result of Data Transformations 1, 2, and 3
Feed Result of Data Transformations 1, 2, and 3 back into Data Transformation Process 1 as an Input
1506

Fig. 16


1600
Receive Data from One or More of a Plurality of Sources
1601
Filter Data Based Upon a Plurality of Anomalies and Disqualifiers
1602
Formalize Data for Optimal Storage and Use within the System
1603
Store Data for Use in Batch Processing
1604
Analyze Stored Batch Data for Changes over Time, Repetitive Data Sets of Interest or Critical Data Points
1605
Perform System Sanity Checks and Streaming Data Analysis Process Retraining based upon Batch Analysis of Input as part of Distributed Computational Graph Process
1606
B

PTO/AIA/01 (06-12)
Approved for use through 01/31/2014. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

DECLARATION (37 CFR 1.63) FOR UTILITY OR DESIGN APPLICATION USING AN APPLICATION DATA SHEET (37 CFR 1.76)

Title of Invention	RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH

As the below named inventor, I hereby declare that:

This declaration is directed to:

- [x] The attached application, or
- [ ] United States application or PCT international application number _____ filed on _____.

The above-identified application was made or authorized to be made by me.

I believe that I am the original inventor or an original joint inventor of a claimed invention in the application.

I hereby acknowledge that any willful false statement made in this declaration is punishable under 18 U.S.C. 1001 by fine or imprisonment of not more than five (5) years, or both.

WARNING:

Petitioner/applicant is cautioned to avoid submitting personal information in documents filed in a patent application that may contribute to identity theft. Personal information such as social security numbers, bank account numbers, or credit card numbers (other than a check or credit card authorization form PTO-2038 submitted for payment purposes) is never required by the USPTO to support a petition or an application. If this type of personal information is included in documents submitted to the USPTO, petitioners/applicants should consider redacting such personal information from the documents before submitting them to the USPTO. Petitioner/applicant is advised that the record of a patent application is available to the public after publication of the application (unless a non-publication request in compliance with 37 CFR 1.213(a) is made in the application) or issuance of a patent. Furthermore, the record from an abandoned application may also be available to the public if the application is referenced in a published application or an issued patent (see 37 CFR 1.14). Checks and credit card authorization forms PTO-2038 submitted for payment purposes are not retained in the application file and therefore are not publicly available.

LEGAL NAME OF INVENTOR

Inventor: Andrew Sellers Date (Optional): 28 Oct 2015

Signature: Andrew Sellers

Note: An application data sheet (PTO/SB/14 or equivalent), including naming the entire inventive entity, must accompany this form or must have been previously filed. Use an additional PTO/AIA/01 form for each additional inventor.

This collection of information is required by 35 U.S.C. 115 and 37 CFR 1.63. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to take 1 minute to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC PAYMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
18/779,029	07/21/2024 06:41:16 PM Z ET	15305FTNK

Title of Invention

RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	-
CONFIRMATION #	4119	FILED BY	Aura Morrey
PATENT CENTER #	66453776	AUTHORIZED BY	Brian Galvin
CUSTOMER #	82318	FILING DATE	-
CORRESPONDENCE ADDRESS	-	FIRST NAMED INVENTOR	Jason Crabtree

Payment Information

PAYMENT METHOD	PAYMENT TRANSACTION ID	PAYMENT AUTHORIZED BY
CARD / 1009	E20247KI42119400	Aura Morrey

FEE CODE	DESCRIPTION	ITEM PRICE($)	QUANTITY	ITEM TOTAL($)
1830	PROCESSING FEE, EXCEPT IN PROVISIONAL APPLICATIONS	140.00	1	140.00
1817	REQUEST FOR PRIORITIZED EXAMINATION	4200.00	1	4200.00
1111	UTILITY PATENT APPL. SEARCH FEE	700.00	1	700.00
1011	BASIC FILING FEE - UTILITY (PAPER FILING ALSO REQUIRES NON-ELECTRONIC FILING FEE UNDER 1.16(T))	320.00	1	320.00

1311	PATENT APPL. EXAMINATION FEE	800.00	1	800.00

TOTAL AMOUNT: **$6,160.00**

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

Doc Code: TRACK1.REQ
Document Description: TrackOne Request

PTO/AIA/424 (04-14)

CERTIFICATION AND REQUEST FOR PRIORITIZED EXAMINATION UNDER 37 CFR 1.102(e) (Page 1 of 1)

First Named Inventor:	Jason Crabtree	Nonprovisional Application Number (if known):	
Title of Invention:	RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH		

APPLICANT HEREBY CERTIFIES THE FOLLOWING AND REQUESTS PRIORITIZED EXAMINATION FOR THE ABOVE-IDENTIFIED APPLICATION.

1. The processing fee set forth in 37 CFR 1.17(i)(1) and the prioritized examination fee set forth in 37 CFR 1.17(c) have been filed with the request. The publication fee requirement is met because that fee, set forth in 37 CFR 1.18(d), is currently $0. The basic filing fee, search fee, and examination fee are filed with the request or have been already been paid. I understand that any required excess claims fees or application size fee must be paid for the application.

2. I understand that the application may not contain, or be amended to contain, more than four independent claims, more than thirty total claims, or any multiple dependent claims, and that any request for an extension of time will cause an outstanding Track I request to be dismissed.

3. The applicable box is checked below:

I. ☑ Original Application (Track One) - Prioritized Examination under § 1.102(e)(1)

i. (a) The application is an original nonprovisional utility application filed under 35 U.S.C. 111(a). This certification and request is being filed with the utility application via EFS-Web.

---OR---

(b) The application is an original nonprovisional plant application filed under 35 U.S.C. 111(a). This certification and request is being filed with the plant application in paper.

ii. An executed inventor's oath or declaration under 37 CFR 1.63 or 37 CFR 1.64 for each inventor, **or** the application data sheet meeting the conditions specified in 37 CFR 1.53(f)(3)(i) is filed with the application.

II. ☐ Request for Continued Examination - Prioritized Examination under § 1.102(e)(2)

i. A request for continued examination has been filed with, or prior to, this form.
ii. If the application is a utility application, this certification and request is being filed via EFS-Web.
iii. The application is an original nonprovisional utility application filed under 35 U.S.C. 111(a), or is a national stage entry under 35 U.S.C. 371.
iv. This certification and request is being filed prior to the mailing of a first Office action responsive to the request for continued examination.
v. No prior request for continued examination has been granted prioritized examination status under 37 CFR 1.102(e)(2).

Signature /BRIAN R. GALVIN/	Date July 21, 2024
Name (Print/Typed) Brian R. Galvin	Practitioner Registration Number 50966

Note: *This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4(d) for signature requirements and certifications. Submit multiple forms if more than one signature is required.**

☑ *Total of 1 forms are submitted.

Privacy Act Statement

The **Privacy Act of 1974 (P.L. 93-579)** requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether disclosure of these records is required by the Freedom of Information Act.
2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.
3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.
4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).
5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.
6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).
7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (*i.e.*, GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.
8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspection or an issued patent.
9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

ABSTRACT

A system for predictive analysis of very large data sets using a distributed computational graph has been developed. Data receipt software receives streaming data from one or more sources. In a batch data pathway, data formalization software formats input data for storage. A batch event analysis server inspects stored data for trends, situations, or knowledge. Aggregated data is passed to message handler software. System sanity software receives status information from message handler and optimizes system performance. In the streaming pathway, transformation pipeline software manipulates the data stream, provides results back to the system, receives directives from the system sanity and retrain software.

Innovation, Sciences et Développement économique Canada
Office de la Propriété Intellectuelle du Canada

Innovation, Science and Economic Development Canada
Canadian Intellectual Property Office

CA 2930026 C 2020/06/16

(11)(21) **2 930 026**

(12) **BREVET CANADIEN**
CANADIAN PATENT

(13) **C**

(86) **Date de dépôt PCT/PCT Filing Date:** 2014/11/11
(87) **Date publication PCT/PCT Publication Date:** 2015/05/14
(45) **Date de délivrance/Issue Date:** 2020/06/16
(85) **Entrée phase nationale/National Entry:** 2016/05/06
(86) **N° demande PCT/PCT Application No.:** US 2014/065052
(87) **N° publication PCT/PCT Publication No.:** 2015/070232
(30) **Priorités/Priorities:** 2013/11/11 (US14/077,162); 2013/11/11 (US14/077,171)

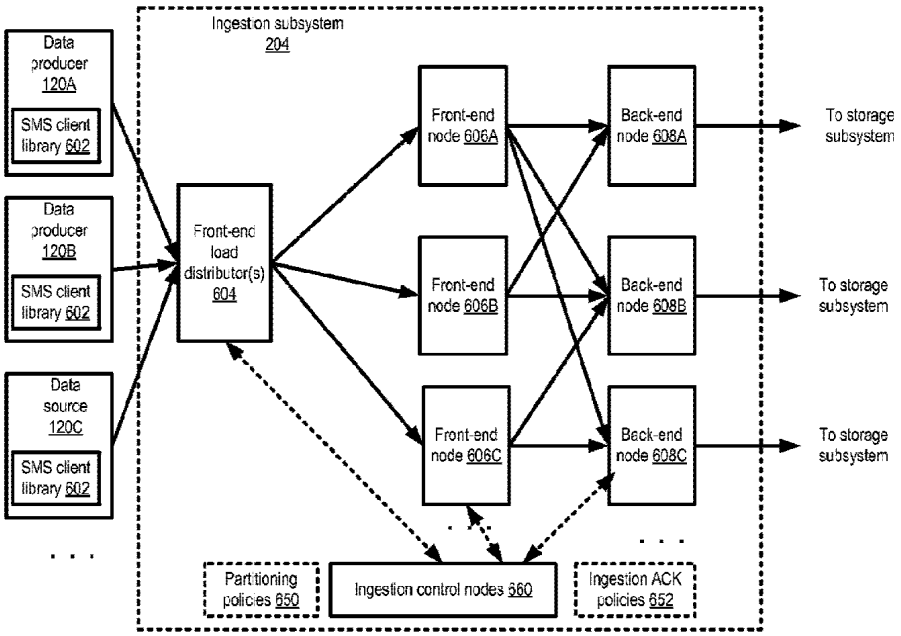
(51) **Cl.Int./Int.Cl.** ***G06F 17/00*** (2019.01), ***G06F 16/00*** (2019.01)

(72) **Inventeurs/Inventors:**
THEIMER, MARVIN MICHAEL, US;
GHARE, GAURAV D., US;
DUNAGAN, JOHN DAVID, US;
BURGESS, GREG, US;
XIONG, YING, US

(73) **Propriétaire/Owner:**
AMAZON TECHNOLOGIES, INC., US

(74) **Agent:** GOWLING WLG (CANADA) LLP

(54) **Titre : TECHNIQUES D'INGESTION ET DE PERSISTANCE DE FLUX DE DONNEES**
(54) **Title: DATA STREAM INGESTION AND PERSISTENCE TECHNIQUES**



(57) **Abrégé/Abstract:**
A programmatic interface is implemented, enabling a client of a stream management service to select a data ingestion policy for a data stream. A client request selecting an at-least-once ingestion policy is received. In accordance with the at-least-once policy, a client may transmit an indication of a data record one or more times to the service until a positive acknowledgement is received. In response to receiving a plurality of transmissions indicating a particular data record, respective positive acknowledgements are sent to the client. Based on a persistence policy selected for the stream, copies of the data record are stored at one or more storage locations in response to one particular transmission of the plurality of transmissions.

50 rue Victoria • Place du Portage 1 • Gatineau, (Québec) K1A 0C9 • www.opic.ic.gc.ca
50 Victoria Street • Place du Portage 1 • Gatineau, Quebec K1A 0C9 • www.cipo.ic.gc.ca

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
14 May 2015 (14.05.2015)

WIPO | PCT

(10) International Publication Number
WO 2015/070232 A1

(51) International Patent Classification:
G06F 17/30 (2006.01)

(21) International Application Number:
PCT/US2014/065052

(22) International Filing Date:
11 November 2014 (11.11.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
14/077,171 11 November 2013 (11.11.2013) US
14/077,162 11 November 2013 (11.11.2013) US

(71) Applicant: **AMAZON TECHNOLOGIES, INC.** [US/US]; P.O. Box 8102, Reno, NV 89507 (US).

(72) Inventors: **THEIMER, Marvin, Michael**; 410 Terry Avenue North, Seattle, WA 98109-5210 (US). **GHARE, Gaurav, D.**; 410 Terry Avenue North, Seattle, WA 98109-5210 (US). **DUNAGAN, John, David**; 410 Terry Avenue North, Seattle, WA 98109-5210 (US). **BURGESS, Greg**; 410 Terry Avenue North, Seattle, WA 98109-5210 (US). **XIONG, Ying**; 410 Terry Avenue North, Seattle, WA 98109-5210 (US).

(74) Agent: **KOWERT, Robert, C.**; Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C., P.O. Box 398, Austin, TX 78767-0398 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report (Art. 21(3))*

(54) **Title**: DATA STREAM INGESTION AND PERSISTENCE TECHNIQUES

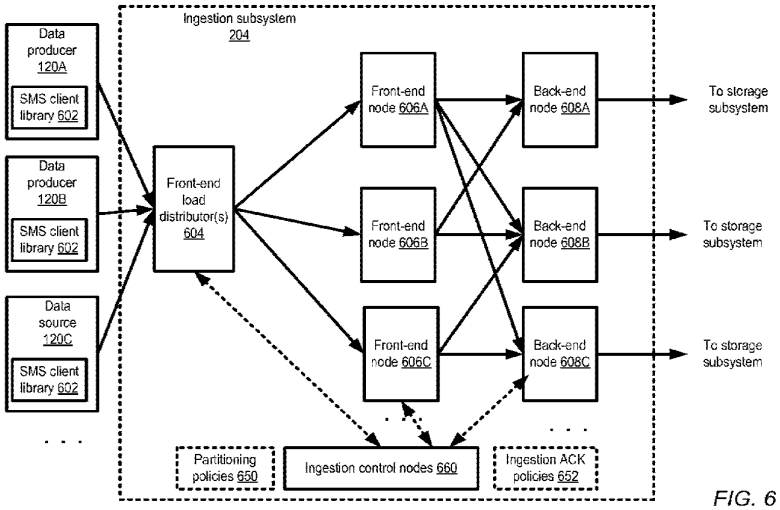


FIG. 6

(57) **Abstract**: A programmatic interface is implemented, enabling a client of a stream management service to select a data ingestion policy for a data stream. A client request selecting an at-least-once ingestion policy is received. In accordance with the at-least-once policy, a client may transmit an indication of a data record one or more times to the service until a positive acknowledgement is received. In response to receiving a plurality of transmissions indicating a particular data record, respective positive acknowledgements are sent to the client. Based on a persistence policy selected for the stream, copies of the data record are stored at one or more storage locations in response to one particular transmission of the plurality of transmissions.

WO 2015/070232 A1

DATA STREAM INGESTION AND PERSISTENCE TECHNIQUES

BACKGROUND

[0001] As the costs of data storage have declined over the years, and as the ability to interconnect various elements of the computing infrastructure has improved, more and more data pertaining to a wide variety of applications can potentially be collected and analyzed. For example, mobile phones can generate data indicating their locations, the applications being used by the phone users, and so on, at least some of which can be collected and analyzed in order to present customized coupons, advertisements and the like to the users. The analysis of data collected by surveillance cameras may be useful in preventing and/or solving crimes, and data collected from sensors embedded at various location within airplane engines, automobiles or complex machinery may be used for various purposes such as preventive maintenance, improving efficiency and lowering costs.

[0002] The increase in volumes of streaming data has been accompanied by (and in some cases made possible by) the increasing use of commodity hardware. The advent of virtualization technologies for commodity hardware has provided benefits with respect to managing large-scale computing resources for many types of applications, allowing various computing resources to be efficiently and securely shared by multiple customers. For example, virtualization technologies may allow a single physical computing machine to be shared among multiple users by providing each user with one or more virtual machines hosted by the single physical computing machine, with each such virtual machine being a software simulation acting as a distinct logical computing system that provides users with the illusion that they are the sole operators and administrators of a given hardware computing resource, while also providing application isolation and security among the various virtual machines. Furthermore, some virtualization technologies are capable of providing virtual resources that span two or more physical resources, such as a single virtual machine with multiple virtual processors that spans multiple distinct physical computing systems. In addition to computing platforms, some large organizations also provide various types of storage services built using virtualization technologies. Using such storage services, large amounts of data can be stored with desired durability levels.

[0003] Despite the availability of virtualized computing and/or storage resources at relatively low cost from various providers, however, the management and orchestration of the collection, storage and processing of large dynamically fluctuating streams of data remains a challenging proposition for a variety of reasons. As more resources are added to a system set up for handling large streams of data, for example, imbalances in workload between different parts of the system

may arise. If left unaddressed, such imbalances may lead to severe performance problems at some resources, in addition to underutilization (and hence wastage) of other resources. Clients may also be concerned regarding the security of their streaming data, or the results of analyzing streaming data, if such data or results are stored at facilities that the clients do not control. The failures that naturally tend to occur with increasing frequency as distributed systems grow in size, such as the occasional loss of connectivity and/or hardware failure, may also have to be addressed effectively to prevent costly disruptions of stream data collection, storage or analysis.

BRIEF DESCRIPTION OF DRAWINGS

[0004] FIG. 1 provides a simplified overview of data stream concepts, according to at least some embodiments.

[0005] FIG. 2 provides an overview of the flow of data among various subcomponents of a stream management system (SMS) and a stream processing system (SPS) comprising a collection of stream processing stages, according to at least some embodiments.

[0006] FIG. 3 illustrates examples of respective sets of programmatic interfaces that may be implemented at an SMS an SPS, according to at least some embodiments.

[0007] FIG. 4 illustrates an example web-based interface that may be implemented to enable SPS clients to generate graphs of stream processing stages, according to at least some embodiments.

[0008] FIG. 5 illustrates examples of programmatic record submission interfaces and record retrieval interfaces that may be implemented at an SMS, according to at least some embodiments.

[0009] FIG. 6 illustrates example elements of an ingestion subsystem of an SMS, according to at least some embodiments.

[0010] FIG. 7 illustrates example elements of a storage subsystem of an SMS according to at least some embodiments.

[0011] FIG. 8 illustrates example elements of a retrieval subsystem of an SMS and examples of interactions of the retrieval subsystem with an SPS, according to at least some embodiments.

[0012] FIG. 9 illustrates examples of redundancy groups that may be set up for nodes of an SMS or an SPS, according to at least some embodiments.

[0013] FIG. 10 illustrates a provider network environment in which the nodes of a given redundancy group may be distributed among a plurality of data centers, according to at least some embodiments.

[0014] FIG. 11 illustrates a plurality of placement destinations that may be selected for nodes of an SMS or an SPS, according to at least some embodiments.

[0015] FIG. 12a and 12b illustrate examples of security option requests that may be submitted by SPS clients and SMS clients, respectively, according to at least some embodiments.

[0016] FIG. 13a illustrates example interactions between a stream data producer and an ingestion node of an SMS, according to at least some embodiments.

[0017] FIG. 13b illustrates example elements of a sequence number that may be generated for an ingested data record at an SMS, according to at least some embodiments.

[0018] FIG. 14 illustrates examples of ordered storage and retrieval of stream data records at an SMS, according to at least some embodiments.

[0019] FIG. 15 illustrates an example of a stream partition mapping and corresponding configuration decisions that may be made for SMS and SPS nodes, according to at least some embodiments.

[0020] FIG. 16 illustrates an example of dynamic stream repartitioning, according to at least some embodiments.

[0021] FIG. 17 is a flow diagram illustrating aspects of operations that may be performed to support respective sets of programmatic interfaces for stream record ingestion and stream record retrieval, according to at least some embodiments.

[0022] FIG. 18a is a flow diagram illustrating aspects of operations that may be performed to configure stream processing stages, according to at least some embodiments.

[0023] FIG. 18b is a flow diagram illustrating aspects of operations that may be performed in response to invocations of components of a client library for configuration of stream processing worker nodes, according to at least some embodiments.

[0024] FIG. 19 is a flow diagram illustrating aspects of operations that may be performed to implement one or more recovery policies for stream processing, according to at least some embodiments.

[0025] FIG. 20 is a flow diagram illustrating aspects of operations that may be performed to implement a plurality of security options for data streams, according to at least some embodiments.

[0026] FIG. 21 is a flow diagram illustrating aspects of operations that may be performed to implement a partitioning policy for data streams, according to at least some embodiments.

[0027] FIG. 22 is a flow diagram illustrating aspects of operations that may be performed to implement dynamic repartitioning of data streams, according to at least some embodiments.

[0028] FIG. 23 is a flow diagram illustrating aspects of operations that may be performed to implement an at-least-once record ingestion policy for data stream records, according to at least some embodiments.

[0029] FIG. 24 is a flow diagram illustrating aspects of operations that may be performed to implement a plurality of persistence policies for data streams, according to at least some embodiments.

[0030] FIG. 25 illustrates an example of a stream processing system in which worker nodes of a processing stage coordinate their workloads using a database table, according to at least some embodiments.

[0031] FIG. 26 illustrates example entries that may be stored in a partition assignment table used for workload coordination, according to at least some embodiments.

[0032] FIG. 27 illustrates aspects of operations that may be performed by worker nodes of a stream processing stage to select partitions on which to perform processing operations, according to at least some embodiments.

[0033] FIG. 28 illustrates aspects of operations that may be performed by worker nodes of a stream processing stage to update a partition assignment table based on information obtained from a stream management service control subsystem, according to at least some embodiments.

[0034] FIG. 29 illustrates aspects of load balancing operations that may be performed by worker nodes of a stream processing stage, according to at least some embodiments.

[0035] FIG. 30 is a block diagram illustrating an example computing device that may be used in at least some embodiments.

[0036] While embodiments are described herein by way of example for several embodiments and illustrative drawings, those skilled in the art will recognize that embodiments are not limited to the embodiments or drawings described. It should be understood, that the drawings and detailed description thereto are not intended to limit embodiments to the particular form disclosed, but on the contrary, the intention is to cover all modifications, equivalents and alternatives falling within the spirit and scope as defined by the appended claims. The headings used herein are for organizational purposes only and are not meant to be used to limit the scope of the description or the claims. As used throughout this application, the word "may" is used in a permissive sense (i.e., meaning having the potential to), rather than the mandatory sense (i.e., meaning must). Similarly, the words "include," "including," and "includes" mean including, but not limited to.

DETAILED DESCRIPTION

[0037] Various embodiments of methods and apparatus for managing the creation, storage, retrieval, and processing of large-scale data streams designed to handle hundreds or even thousands of concurrent data producers and data consumers are described. The term "data stream", as used herein, refers to a sequence of data records that may be generated by one or

more data producers and accessed by one or more data consumers, where each data record is assumed to be an immutable sequence of bytes. A stream management service (SMS) may provide programmatic interfaces (e.g., application programming interfaces (APIs), web pages or web sites, graphical user interfaces, or command-line tools) to enable the creation, configuration and deletion of streams, as well as the submission, storage and retrieval of stream data records in some embodiments. Some types of stream operations (such as stream creation or deletion, or the kinds of dynamic repartitioning operations described below) that involve interactions with SMS control components may be referred to as "control-plane" operations herein, while operations such as data record submissions, storage and retrievals that typically (e.g., under normal operating conditions) do not require interactions with control components may be referred to herein as "data-plane" operations. Dynamically provisioned sets of compute, storage and networking resources may be used to implement the service in some such embodiments, based for example on various partitioning policies that allow the stream management workload to be distributed in a scalable fashion among numerous service components, as described below in further detail. The acronym SMS may be used herein to refer to a stream management service, and also to a stream management system comprising the collection of virtual and/or physical resources used to implement a stream management service.

[0038] Some customers of the SMS may develop applications that directly invoke the SMS programmatic interfaces in various embodiments. In at least some embodiments, however, in addition to the SMS interfaces, a higher-level abstraction or application-level processing framework may be provided for customers, which may simplify various aspects of stream processing for those clients that do not wish to develop applications using the lower-level stream management functions supported directly by the SMS. Such a framework may provide its own programmatic interfaces (built, for example, on top of the SMS interfaces), enabling customers to focus more on the business logic to be implemented using stream records than on lower-level stream management operations. The higher-level framework may be implemented as a stream processing service (SPS) with its own control-plane and data-plane components in some embodiments, which may provide advanced functionality such as automated resource provisioning for stream processing, automated failovers of processing nodes, the ability to construct arbitrary stream processing workflow graphs, support for ephemeral streams, dynamic repartitioning based on workload changes or other triggering conditions, and so on. In at least some embodiments, either the stream management service, the stream processing service, or both services, may be implemented as multi-tenant managed network-accessible services in a virtualization environment. That is, various physical resources (such as computer servers or

hosts, storage devices, networking devices and the like) may at least in some cases be shared among streams of different customers in such embodiments, without necessarily making the customers aware of exactly how the resources are being shared, or even making a customer aware that a given resource is being shared at all. Control components of the managed multi-tenant stream management and/or processing managed services may dynamically add, remove, or reconfigure nodes or resources being used for a particular stream based on various applicable policies, some of which may be client-selectable. In addition, the control components may also be responsible for transparently implementing various types of security protocols (e.g., to ensure that one client's stream application cannot access another client's data, even though at least some hardware or software may be shared by both clients), monitoring resource usage for billing, generating logging information that can be used for auditing or debugging, and so on. From the perspective of clients of the managed multi-tenant service(s), the control/administrative functionality implemented by the service(s) may eliminate much of the complexity involved in supporting large-scale streaming applications. In some scenarios, customers of such multi-tenant services may be able to indicate that they do not wish to share resources for at least some types of stream-related operations, in which case some physical resources may be designated at least temporarily as being single-tenant for those types of operations (i.e., limited to operations performed on behalf of a single customer or client).

[0039] A number of different approaches may be taken to the implementation of SMS and/or SPS control-plane and data-plane operations in various embodiments. For example, with respect to control-plane operations, in some implementations a redundancy group of control servers or nodes may be set up. The redundancy group may include a plurality of control servers, of which one server is designated as a primary server responsible for responding to administrative requests regarding various streams, while another server may be designated to take over as the primary in the event of a triggering condition such as a failure at (or loss of connectivity to) the current primary. In another implementation, one or more tables created at a network-accessible database service may be used to store control-plane metadata (such as partition maps) for various streams, and various ingestion, storage or retrieval nodes may be able to access the tables as needed to obtain the subsets of metadata required for data-plane operations. Details regarding various aspects of the SPS and the SMS data-plane and control-plane functionality in different embodiments are provided below. It is noted that in some embodiments in which a stream management service is implemented, a stream processing service providing higher-level primitives may not necessarily be implemented. In other embodiments, only high-level

programmatic interfaces of a stream processing service may be exposed to customers, and lower-level stream management interfaces used by the may not be made available to clients.

[0040] According to some embodiments, a stream management system may comprise a plurality of independently configurable subsystems, including a record ingestion subsystem primarily responsible for obtaining or collecting data records, a record storage subsystem primarily responsible for saving the data record contents in accordance with applicable persistence or durability policies, and a record retrieval subsystem primarily responsible for responding to read requests directed at the stored records. A control subsystem may also be implemented in some embodiments, comprising one or more administrative or control components responsible for configuring the remaining subsystems, e.g., by dynamically determining and/or initializing the required number of nodes for each of the ingestion, storage and retrieval subsystems at selected resources such as virtual or physical servers. Each of the ingestion, storage, retrieval and control subsystems may be implemented using a respective plurality of hardware and/or software components which may collectively be referred as "nodes" or "servers" of the subsystems. The various resources of an SMS may thus be logically said to belong to one of four functional categories: ingestion, storage, retrieval or control. In some implementations, respective sets of control components may be established for each of the other subsystems, e.g., independent ingestion control subsystems, storage control subsystems and/or retrieval control subsystems may be implemented. Each such control subsystem may be responsible for identifying the resources to be used for the other nodes of the corresponding subsystem and/or for responding to administrative queries from clients or from other subsystems. In some implementations, pools of nodes capable of performing various types of SMS and/or SPS functions may be set up in advance, and selected members of those pools may be assigned to new streams or new processing stages as needed.

[0041] Stream partitioning policies and associated mappings may be implemented in at least some embodiments, e.g., to distribute subsets of the data records between different sets of ingestion, storage, retrieval and/or control nodes. For example, based on the partitioning policy selected for a particular data stream as well as on other factors such as expectations of record ingestion rates and/or retrieval rates, a control component may determine how many nodes (e.g., processes or threads) should be established initially (i.e., at stream creation time) for ingestion, storage and retrieval, and how those nodes should be mapped to virtual and/or physical machines. Over time, the workload associated with a given stream may increase or decrease, which (among other triggering conditions) may lead to repartitioning of the stream. Such re-partitioning may involve changes to various parameters, such as the function to be used to

determine a record's partition, the partitioning keys used, the total number of partitions, the number of ingestion nodes, storage nodes or retrieval nodes, or the placement of the nodes on different physical or virtual resources. In at least some embodiments, the repartitioning may be implemented dynamically without interrupting the flow of the data records, using techniques described below in further detail. Different partitioning schemes and repartition-triggering criteria may be used for different data streams in some embodiments, e.g., based on client-provided parameters or on heuristics of the SMS control nodes. In some embodiments, it may be possible to limit the number and/or frequency of repartitions, e.g., based on client preferences, the expected lifetime of a stream, or other factors.

[0042] A number of different record ingestion policies and interfaces may be implemented in different embodiments. For example, in some embodiments, clients (e.g., executable components or modules configured to invoke the programmatic interfaces of the SMS on behalf of customers of the SMS) may utilize either in-line submission interfaces, or by-reference submission interfaces. For in-line submissions, the contents or body of the data record may be included as part of the submission request in such embodiments. In contrast, in a by-reference submission request, an address (such as a storage device address, a database record address, or a URL (Uniform record Locator)) may be provided from which the contents or body of the data record can be obtained. In some implementations, a hybrid submission interface may also or instead be supported, in which up the first N bytes of the data record may be included in-line, while the remaining bytes (if any) are provided by reference. In such a scenario, short records (whose bodies are less than N bytes long) may be fully specified by the submission request, while portions of longer records may have to be obtained from the corresponding address.

[0043] In addition to the different alternatives for specifying record contents during ingestion, in some embodiments a variety of acknowledgement or de-duplication related ingestion policies may also be implemented. For example, for some stream applications, clients may wish to ensure that each and every data record is ingested reliably by the SMS. In large distributed stream management environments, packets may be lost, or various failures may occur from time to time along the path between the data producers and the ingestion nodes, which could potentially result in some submitted data being lost. In some embodiments, therefore, an SMS may implement an at-least-once ingestion policy, in accordance with which a record submitter may submit the same record one or more times until a positive acknowledgement is received from the ingestion subsystem. Under normal operating conditions, a record may be submitted once, and the submitter may receive an acknowledgement after the receiving ingestion node has obtained and stored the record. If the acknowledgement is lost or delayed, or if the

record submission request itself was lost, the submitter may resubmit the same data record one or more times, until eventually an acknowledgement is received. The ingestion node may, for example, generate an acknowledgement for each submission, regardless of whether it is a duplicate or not, based on an expectation that the record would not be resubmitted if an acknowledgement had already been received by the submitter. The ingestion node may, however, be responsible in at least some embodiments for recognizing that the same data record has been submitted multiple times, and for avoiding storing new copies of the duplicate data unnecessarily. In one embodiment, at least two versions of an at-least-once ingestion policy may be supported – one version (which may be termed "at-least-once ingestion, no-duplication") in which the SMS is responsible for de-duplicating data records (i.e., ensuring that data is stored at the SMS storage subsystem in response to only one of a set of two or more submissions), and one version in which duplication of data records storage by the SMS is permitted (which may be termed "at-least-once, duplication-permitted"). The at-least-once, duplication-permitted approach may be useful for stream applications in which there are few or no negative consequences of data record duplication, and/or for stream applications that perform their own duplicate elimination. Other ingestion policies may also be supported, such as a best-effort ingestion policy in which acknowledgements are not required for every data record submitted. The loss of a few data records may be acceptable if a best-effort ingestion policy is in effect in at least some embodiments. Clients may select which ingestion policies they wish to use for various streams in various embodiments.

[0044] With respect to the storage of stream records, a number of alternative policies may also be supported in at least some embodiments. For example, a client may be able to choose a persistence policy from among several supported by the SMS, which governs such aspects of record storage as the number of copies of a given data record that are to be stored, the type of storage technology (e.g., volatile or non-volatile RAM, rotating disk-based storage, solid state devices (SSDs), network attached storage devices, and the like) to be used for the copies, and so on. For example, if a client selects an N-replica persistence policy to disk-based storage, a data record submission may not be considered complete until N copies of the record have been safely written to N respective disk devices. In at least some embodiments in which disk-based storage devices are used, the SMS storage subsystem may attempt to write incoming data records of a given partition sequentially to disk, e.g., to avoid the performance impact of disk seeks. Sequence numbers may be generated for (and stored with) data records using various techniques as described below, including for example timestamp-based techniques that enable ordered record retrieval based on ingestion times. Data records of a given partition may be stored

together, e.g., contiguously on disk, and separately from the data records of other partitions in at least some embodiments. In some implementations, in accordance with a retention policy (selected by a client or by the SMS) or a de-duplication time window policy (indicating the time period, subsequent to a submission of any given data record, during which the SMS may be required to ensure that no duplicates of that given data record are stored in the SMS storage subsystem, even if some duplicates are submitted), at least some data records may be archived to a different types of storage service and/or deleted after a time period from the SMS. Such removal operations may be referred to herein as stream "trimming". Clients may submit stream trimming requests in some embodiments, e.g., notifying the SMS that specified data records are no longer needed and can therefore be deleted from the perspective of the client submitting the trimming request, or explicitly requesting the deletion of specified data records. In scenarios in which there may be multiple clients consuming the data records of a given stream, the SMS may be responsible for ensuring that a given record is not deleted or trimmed prematurely, before it has been accessed by all the interested consumers. In some implementations, if there are N data consumers of a given stream, before deleting a given record R of the stream, he SMS may wait until it has determined that all N data consumers have read or processed R. The SMS may determine that R has been read by all the consumers based on respective trimming requests from the consumers, for example, or based on respective indications of how far within the stream the data consumers have progressed. In some embodiments, some types of data consumers (such as testing-related applications) may accept the deletion of at least a small subset of data records before they have been accessed. Accordingly, applications may be able to notify the SMS regarding the acceptability of data deletion prior to retrieval in at least some embodiments, and the SMS may schedule deletions in accordance with the notifications. In some embodiments, an archival policy may be implemented, e.g., as part of the data retention policy, indicating for example the types of storage devices to which stream data records should be copied, and the scheduling policies to be used for such copies.

[0045] In at least some embodiments, a plurality of programmatic interfaces may also be supported for record retrieval. In one embodiment, an iterator-based approach may be used, in which one programmatic interface (e.g., getIterator) may be used to instantiate and position an iterator or cursor at a specified logical offset (e.g., based on sequence number or timestamp) within a partition of a stream. A different programmatic interface (such as getNextRecords) may then be used to read a specified number of data records sequentially starting from the current position of the iterator. The instantiation of an iterator may in effect allow a client to specify an arbitrary or random starting position for record retrieval within the stream partition. If a client

wishes to read data records in a random access pattern in such an embodiment, the client may have to repeatedly create new iterators. In rotating disk based storage systems, disk seeks required for frequent random accesses may impact I/O response times significantly. Accordingly, as an incentive to clients to read stream data records sequentially rather than randomly, different (e.g., higher) billing rates may be applied to random read accesses than are applied to sequential read accesses in at least some embodiments. Thus, for example, a client may be billed X currency units per getIterator call, and Y currency units per record retrieved via getNextRecords, with X > Y in some implementations. When alternative client interfaces are supported for other operation categories (such as ingestion), in at least some embodiments the billing rates or prices for the alternatives may also differ – e.g., a client may be charged more for a by-reference submission request than for an on-line submission request, just as a client may be charged more for random reads than for sequential reads. Other factors may also influence billing in various embodiments, such as the sizes of the data records, the distribution of write versus read requests over time, the persistence policies selected, and so on.

[0046] According to some embodiments, a stream processing service (SPS) may allow clients to specify arbitrarily complex processing workflows comprising numerous processing stages, in which the output of the processing performed at a given stage may be used as input for zero or more other stages. Partitioning policies (similar to those described for the SMS for ingesting, storing and retrieving data records) may be used to divide the processing workload among a plurality of worker nodes at various stages in some embodiments. In one such embodiment, programmatic SPS interfaces may be implemented enabling clients to specify various configuration settings for any given stage, including for example the input data source(s) for the stage (e.g., one or more streams from which data records are to be retrieved, together with the partitioning policies for the streams), the processing operations to be performed at the stage, and a descriptor or specification for output or result distribution from the stage (e.g., whether the output is to be saved to storage locations, sent to a network endpoint, or fed into one or more other processing stages in the form of a different stream). In at least some embodiments, the processing operations specified for an SPS stage may be idempotent: that is, if a given processing operation is performed multiple times on the same input data, the result of the operation does not differ from the result that would have been obtained if the operation were performed just once. Recoveries from failures (e.g., a worker node failure at an SPS stage) may be simplified if the processing operations are idempotent, as described below in further detail. According to some embodiments, non-idempotent processing operations may be permitted at some or all SPS stages.

[0047] Based at least in part on configuration information such as the input stream partitioning policies and then nature of the processing operations received via the SPS programmatic interfaces, in various embodiments SPS control servers may determine how many worker nodes are to be set up initially for various stages of a processing workflow. The performance capabilities of the resources to be used for the worker nodes (e.g., the virtual or physical machines being used) may also be taken into account when determining the initial number and placement of the worker nodes. The selected number of worker nodes (which may in some implementations each comprise an executable thread or an executable process) may be instantiated. Each worker node may be configured, for example, to obtain data records from the appropriate input sources (e.g., from retrieval nodes of one or more stream partitions), perform the specified processing operations on the data records, and transmit the results of the processing to the specified output destination(s). In addition, in at least some embodiments, a checkpoint scheme may be implemented, in accordance with which a given worker node may be configured to store progress records or checkpoints indicative of the portion of a partition that has been processed at that worker node, with the assumption that the partition records are being processed sequentially. The worker node may, for example, write a progress record to persistent storage periodically in some implementations (e.g., once every N seconds or once every R data records have been processed), and/or in response to checkpoint requests from an SPS control server.

[0048] The progress records may be used for rapid recovery from worker node failures in some embodiments. For example, an SPS control server may monitor the health status of the various worker nodes over time, e.g., using a heartbeat mechanism and/or by monitoring resource utilization levels (such as CPU utilization, I/O device utilization, or network utilization levels). In response to a determination by the SPS control server that a particular worker node is in an undesired or unhealthy state (e.g., if it is unresponsive or overloaded), a replacement worker node may be instantiated to take over the responsibilities of the particular worker node. The replacement worker node may access the most recent progress record stored by the replaced worker node to identify the set of data records that the replacement worker node should process. In embodiments in which the processing operations are idempotent, even if some operations are repeated (e.g., because the most recent progress record was written some time prior to the replacement worker's instantiation), the overall results of the processing would not be affected by the failure and replacement. In some implementations, in addition to storing progress records indicating the subset of a given stream or partition that has been processed by it, a worker node may also be configured to store accumulated application state information. For example, if a stream processing workflow is responsible for determining client billing amounts for a particular

service based on analyzing streaming data records that indicate service usage metrics, a worker node may periodically store the cumulative billing amounts determined for various clients.

[0049] In at least some embodiments, the SPS control servers may also be configured to respond to various other triggers such as changing workload levels or detected workload imbalances (e.g., if the ingestion rates for one partition become disproportionately higher than those of others) by initiating other actions, such as requesting dynamic repartitioning of the input streams for various stages, changing the number of worker nodes assigned to a given partition at a given stage., assigning higher-performance worker nodes to some stages, or transferring worker nodes from one physical resource to another physical resource with a different performance capability. In some embodiments, e.g., in response to a determination by an SPS control server that a best-effort recovery policy is to be implemented for a given stage rather than a checkpoint-based recovery policy, progress records of the type described above may not be stored by worker nodes of at least some SPS stages. In some implementations of such a best-effort recovery policy, a replacement worker node may simply process new data records as they are received, without requiring access to progress records. In some embodiments, if a client wishes to implement a best-effort recovery policy at an SPS stage, the stream processing operations performed at the stage need not necessarily be idempotent. In some embodiments in which non-idempotent processing operations are to be performed on stream records at an SPS stage, checkpoint-based recovery may not be supported, and a different recovery scheme such as best-effort recovery may be used. In at least one embodiment, only idempotent stream processing operations may be allowed at SPS stages.

[0050] The data records of some streams may contain sensitive or confidential information, or the processing operations performed at the SPS stages may comprise the use of proprietary algorithms whose discovery by competitors may be problematic. Clients may thus be concerned about the security of various categories of stream management and processing operations, especially if the operations are performed using resources located at provider network data centers that are not fully controlled by the clients themselves. Networks set up by an entity such as a company or a public sector organization to provide one or more network-accessible services (such as various types of cloud-based database, computing or storage services) accessible via the Internet and/or other networks to a distributed set of clients may be termed provider networks herein. In some embodiments, clients may be able to choose from among a plurality of security-related options for their data streams. As described above, a combined SPS and SMS configuration may comprise nodes belonging to a number of different functional categories, such as control nodes for the SMS and/or the SPS, SMS ingestion nodes, SMS storage nodes, SMS

retrieval nodes, and SPS processing or worker nodes. The security-related choices made available to clients may include options for placement or locations of various types of nodes in some embodiments. For example, in one embodiment, a client may be able to request that SPS worker nodes for one or more processing stages of a stream workflow be implemented at computing devices located on client-owned facilities, even if the stream records are initially collected and/or stored using resources located at a provider network. In response to such placement requests, nodes of different functional categories for a given stream may be instantiated at respective resource sets with differing security characteristics or profiles.

[0051] The resource sets may differ from one another in various security-related characteristics in different embodiments, including for example physical location, physical security protocols being used (e.g., who has physical access to the resources), network isolation levels (e.g., the extent to which network addresses of the resources are visible to various entities), multi-tenancy versus single-tenancy, and so on. In some embodiments, clients may be able to establish isolated virtual networks (IVNs) within a provider network, with a given client being given substantial control over networking configurations of various devices included within that client's IVN. In particular, clients may be able to restrict access to the network addresses (e.g., Internet Protocol or IP addresses) assigned to various servers or compute instances within their IVNs. In such embodiments, clients may request that certain subsets of their SMS or SPS nodes be instantiated within specified IVNs. In embodiments in which provider network resources such as virtualization instance hosts (which may typically be configured as multi-tenant hosts) are being used for various categories of SMS or SPS nodes, a client may request that some set of nodes be instantiated on instance hosts that are restricted to implementing instances belonging to that client alone (i.e., some SMS or SPS nodes may be implemented at instance hosts configured as single-tenant hosts).

[0052] In some embodiments, as another security-related option, clients may request that the data records of a particular stream be encrypted before they are transmitted over a network link - e.g., before being ingested at the SMS, between the ingestion and storage subsystems, between the storage and retrieval subsystems, between the retrieval subsystems and the SPS worker nodes, and/or between the worker nodes and the SPS output destinations. Clients may specify the encryption algorithms to be used in some embodiments. In one embodiment, secure networking protocols such as TLS (Transport Layer Security) or SSL (secure sockets layer) protocols may be used for data record transmissions and/or for transmitting SPS processing results.

Data stream concepts and overview

[0053] FIG. 1 provides a simplified overview of data stream concepts, according to at least some embodiments. As shown, a stream 100 may comprise a plurality of data records (DRs) 110, such as DRs 110A, 110B, 110C, 110D and 110E. One or more data producers 120 (which may also be referred to as data sources), such as data producers 120A and 120B, may perform write operations 151 to generate the contents of data records of stream 100. A number of different types of data producers may generate streams of data in different embodiments, such as, for example, mobile phone or tablet applications, sensor arrays, social media platforms, logging applications or system logging components, monitoring agents of various kinds, and so on. One or more date consumers 130 (such as data consumers 130A and 130B) may perform read operations 152 to access the contents of the data records generated by the data producers 120. Data consumers 130 may comprise, for example, worker nodes of a stream processing stage in some embodiments.

[0054] In at least some embodiments, a given data record 110 as stored in an SMS may comprise a data portion 101 (e.g., data portions 101A, 101B, 101C, 101D and 101E of DRs 110A, 110B, 110C, 110D and 110E respectively) and a sequence number SN 102 (e.eg., SNs 102A, 102B, 102C, 102D and 102E of DRs 110A, 110B, 110C, 110D and 110E respectively). The sequence number 102 may be indicative of the order in which the DRs are received at a stream management system (or at a particular node of a stream management system) in the depicted embodiment. The data portions 101 may comprise immutable un-interpreted byte sequences in some implementations: that is, once a write operation 152 is completed, the contents of the DR generated as a result of the write may not be changed by the SMS, and in general the SMS may not be aware of the semantics of the data. In some implementations, different data records of a given stream 100 may comprise different amounts of data, while in other implementations, all the data records of a given stream may be of the same size. In at least some implementations, nodes of the SMS (e.g., ingestion subsystem nodes and/or storage subsystem nodes) may be responsible for generating the SNs 102. As described below in further detail, the sequence numbers of the data records need not always be consecutive. In one implementation, clients or data producers 120 may provide, as part of a write request, an indication of a minimum sequence number to be used for the corresponding data record. In some embodiments, data producers 120 may submit write requests that contain pointers to (or addresses of) the data portions of the data records, e.g., by providing a storage device address (such as a device name and an offset within the device) or a network address (such as a URL) from which the data portion may be obtained.

[0055] The stream management service may be responsible for receiving the data from the data producers 120, storing the data, and enabling data consumers 130 to access the data in one or more access patterns in various embodiments. In at least some embodiments, the stream 100 may be partitioned or "sharded" to distribute the workload of receiving, storing, and retrieving the data records. In such embodiments, a partition or shard may be selected for an incoming data record 110 based on one or more attributes of the data record, and the specific nodes that are to ingest, store or retrieve the data record may be identified based on the partition. In some implementations, the data producers 120 may provide explicit partitioning keys with each write operation which may serve as the partitioning attributes, and such keys may be mapped to partition identifiers. In other implementations, the SMS may infer the partition ID based on such factors as the identity of the data producer 120, the IP addresses of the data producers, or even based on contents of the data submitted. In some implementations in which data streams are partitioned, sequence numbers may be assigned on a per-partition basis – for example, although the sequence numbers may indicate the order in which data records of a particular partition are received, the sequence numbers of data records DR1 and DR2 in two different partitions may not necessarily indicate the relative order in which DR1 and DR2 were received. In other implementations, the sequence numbers may be assigned on a stream-wide rather than a per-partition basis, so that if sequence number SN1 assigned to a data record DR1 is lower than sequence number SN2 assigned to data record DR2, this would imply that DR1 was received earlier than DR2 by the SMS, regardless of the partitions to which DR1 and DR2 belong.

[0056] The retrieval or read interfaces supported by an SMS may allow data consumers 130 to access data records sequentially and/or in random order in various embodiments. In one embodiment, an iterator-based set of read application programming interfaces (APIs) may be supported. A data consumer 130 may submit a request to obtain an iterator for a data stream, with the initial position of the iterator indicated by a specified sequence number and/or a partition identifier. After the initiator is instantiated, the data consumer may submit requests to read data records in sequential order starting from that initial position within the stream or the partition. If a data consumer wishes to read data records in some random order, a new iterator may have to be instantiated for each read in such embodiments. In at least some implementations, the data records of a given partition or stream may be written to disk-based storage in sequence number order, typically using sequential write operations that avoid disk seeks. Sequential read operations may also avoid the overhead of disk seeks. Accordingly, in some embodiments, data consumers may be encouraged to perform more sequential reads than

random reads using pricing incentives: e.g., random-access read operations such as iterator instantiations may have higher associated billing rates than sequential-access read operations.

Example system environment

[0057] FIG. 2 provides an overview of the flow of data among various subcomponents of a stream management system (SMS) and a stream processing system (SPS) comprising a collection of stream processing stages, according to at least some embodiments. As shown, the SMS 280 may comprise an ingestion subsystem 204, a storage subsystem 206, a retrieval subsystem 208, and an SMS control subsystem 210. Each of the SMS subsystems may include one or more nodes or components, implemented for example using respective executable threads or processes instantiated at various resources of a provider network (or a client-owned or third-party facility) as described below. Nodes of the ingestion subsystem 204 may be configured (e.g., by nodes of the SMS control subsystem 210) to obtain data records of a particular data stream from data producers 120 (such as 120A, 120B, and 120C) based on a partitioning policy in use for the stream, and each ingestion node may pass received data records on to corresponding nodes of the storage subsystem 206. The storage subsystem nodes may save the data records on any of various types of storage devices in accordance with a persistence policy selected for the stream. Nodes of the retrieval subsystem 208 may respond to read requests from data consumers, such as worker nodes of SPS 290. Stream processing stages 215, such as stages 215A, 215B, 1215C and 215D of the SPS 290 may be established with the help of SPS control subsystem 220. Each stage 215 may include one or more worker nodes configured by the SPS control subsystem 220 to perform a set of processing operations on received data records. As shown, some stages 215 (such as 215A and 215B) may obtain data records directly from the SMS 280, while others (such as 215C and 215D) may receive their inputs from other stages. Multiple SPS stages 215 may operate in parallel in some embodiments, e.g., different processing operations may be performed concurrently on data records retrieved from the same stream at stages 215A and 215B. It is noted that respective subsystems and processing stages similar to those illustrated in FIG. 2 for a particular stream may be instantiated for other streams as well.

[0058] In at least some embodiments, at least some of the nodes of the subsystems and processing stages shown in FIG. 2 may be implemented using provider network resources. As noted earlier, networks set up by an entity such as a company or a public sector organization to provide one or more network-accessible services (such as various types of cloud-based database, computing or storage services) accessible via the Internet and/or other networks to a distributed set of clients may be termed provider networks herein. Some of the services may be used to build higher-level services: for example, computing, storage or database services may be used as

building blocks for a stream management service or a stream processing service. At least some of the core services of a provider network may be packaged for client use in service units called "instances": for example, a virtual machine instantiated by a virtualized computing service may represent a "compute instance", and a storage device such as a block-level volume instantiated by a storage service may be referred to as a "storage instance", or a database management server may be referred to as a "database instance". Computing devices such as servers at which such units of various network-accessible services of a provider network are implemented may be referred to as "instance hosts" or more simply as "hosts" herein. Nodes of the ingestion subsystem 204, the storage subsystem 206, the retrieval subsystem 208, the SMS control system 210, the processing stages 215, and/or the SPS control subsystem 220 may comprises threads or processes executing at various compute instances on a plurality of instance hosts in some embodiments. A given instance host may comprise several compute instances, and the collection of compute instances at a particular instance host may be used to implement nodes for various different streams of one or more clients. Storage instances may be used for storing the data records of various streams in some embodiments, or as destinations of the results of stream processing stages. Over time, control subsystem nodes may modify the populations of other subsystems dynamically in response to various triggering conditions, e.g., by adding or removing nodes, changing the mappings of nodes to processes or compute instances or instance hosts, or re-partitioning a given stream while still continuing to receive, store and process data records as described below with reference to FIG. 15 and FIG. 16.

[0059] In the context of embodiments in which provider network resources are used for stream-related operations, the term "client", when used as the source or destination of a given communication, may refer to any of the computing devices, processes, hardware modules or software modules that are owned by, managed by, or allocated to, an entity (such as an organization, a group with multiple users or a single user) that is capable of accessing and utilizing at least one network-accessible service of a provider network. Clients of one service may themselves be implemented using resources of another service – e.g., a stream data consumer (a client of a stream management service) may comprise a compute instance (a resource provided by a virtualized computing service).

[0060] A given provider network may include numerous data centers (which may be distributed across different geographical regions) hosting various resource pools, such as collections of physical and/or virtualized computer servers, storage servers with one or more storage devices each, networking equipment and the like, needed to implement, configure and distribute the infrastructure and services offered by the provider. A number of different hardware

and/or software components, some of which may be instantiated or executed at different data centers or in different geographical regions, may collectively be used to implement each of the services in various embodiments. Clients may interact with resources and services at the provider network from devices located at client-owned or client-managed premises or data centers external to the provider network, and/or from devices within the provider network. It is noted that although provider networks serve as one example context in which many of the stream management and processing techniques described herein may be implemented, those techniques may also be applied to other types of distributed systems than provider networks, e.g., to large-scale distributed environments operated by a single business entity for its own applications.

Programmatic interface examples

[0061] As indicated above, in at least some embodiments an SPS may utilize SMS programmatic interfaces to build higher-level functionality that can more easily be used by SPS clients to implement the desired business logic for various stream-based applications. When considering the differences between SPS and SMS functionality, an analogy may be helpful: SPS functions may in general be compared to programming language constructs in higher-level languages such as C++, while SMS functions may in general be compared to the assembly language instructions to which the programming language constructs are translated by a compiler. It may be possible to implement the same operations using the assembly language instructions directly, but programming in the higher-level language may typically be easier for many categories of customers or users. Similarly, it may be possible to implement various applications using the primitives provided by an SMS, but it may be easier to do so using SPS features. SPS processing operations (such as idempotent processing operations performed on data records) may be implemented on the data contents of the stream records, while the SMS operations are performed to acquire, store and retrieve the records themselves, usually without considering the contents of the records. FIG. 3 illustrates examples of respective sets of programmatic interfaces that may be implemented at an SMS an SPS, according to at least some embodiments. A number of different application programming interfaces (APIs) are indicated for both the SMS and the SPS by way of example. The APIs illustrated are not intended to be exhaustive lists of those supported in any given implementation, and some of the illustrated APIs may not be supported in a given implementation.

[0062] As indicated by arrow 350, SPS clients 375 may invoke SPS programmatic interfaces 305 to configure processing stages. Various types of SPS programmatic interfaces 305 may be implemented in different embodiments. For example, a createStreamProcessingStage API may enable clients to request the configuration of a new processing stage 215 for a specified input

stream, such that worker nodes of the stage are each configured to perform a set of idempotent operations specified in the interface invocation, and to distribute the results to destinations indicated by an output distribution descriptor or policy. In some versions of the createStreamProcessingStage API or its equivalent, a client may request the creation of the input stream as well, while in other versions, an input stream may have to be created before the processing stage is created. A recovery policy may be specified for the worker nodes, indicating for example whether a checkpoint-based recovery technique is to be used or a best-effort recovery technique is preferred. In some embodiments an initializeWorkerNode API may be supported to request the explicit instantiation of worker nodes at a specified stage. In embodiments in which checkpoint-based recovery is implemented, a saveCheckpoint API may be supported to allow clients to request that progress records be generated by worker nodes.

[0063] Various types of SPS output management APIs may be supported in different embodiments, such as a setOutputDistribution API by which a client may indicate one or more streams to be created using the results of the processing operations performed at a specified stage, and the particular partitioning policies to be used for the newly created streams. Some processing stages may be configured primarily for repartitioning – e.g., one partitioning function PF1 that maps data records to N1 partitions based on record attribute set A1 may be in use for an input stream S1, and a processing stage may be used to implement a different partitioning function PF2 to map those same data records to N2 partitions (using either a different attribute set A2, or the same attribute set A1). Some SPS APIs such as linkStages may be used to configure arbitrary graphs (e.g., directed acyclic graphs) comprising a plurality of stages. In some embodiments, connectors to third-party or open-source stream processing frameworks or services may be supported. In one such embodiment, an SPS stage may be used to prepare data records (e.g., by appropriately formatting results of the processing operations performed at the stage) for consumption by existing third-party or open-source systems. An API such as createThirdPartyConnector may be used to set up such connectors in the depicted embodiment, and the appropriate transformations of the results of the SPS stage into a format compatible with the third party system may be performed by one or more connector modules instantiated as a result of a createThirdPartyConnector invocation.

[0064] The SPS may invoke SMS APIs 307 to perform at least some of its functions, as indicated by arrow 352. The SMS APIs 307 may include, for example, createStream and deleteStream (to create and delete a stream, respectively) and getStreamInfo (to obtain metadata for a stream, such as the network addresses of various types of nodes responsible for a given partition) in the depicted embodiment. A putRecord interface may be used to write data records,

while the getIterator and getNextRecords interfaces may be used for non-sequential and sequential reads respectively. A repartitionStream interface may be used to request dynamic repartitioning of a specified stream in some embodiments. Clients 370 that wish to do so may invoke the SMS APIs 307 directly, as indicated by arrow 354. As indicated earlier, various other SMS and/or SPS APIs may also be implemented in other embodiments, and some of the APIs listed in FIG. 3 may not be implemented in some embodiments.

[0065] In various embodiments, programmatic interfaces other than APIs may also or instead be implemented for either the SPS or the SMS. Such interfaces may include graphical user interfaces, web pages or web sites, command-line interfaces, and the like. In some cases web-based interfaces or GUIs may use the APIs as building blocks – e.g., a web-based interaction may result in the invocation of one or more APIs at control components of the SMS or SPS. FIG. 4 illustrates an example web-based interface that may be implemented to enable SPS clients to generate graphs of stream processing stages, according to at least some embodiments. As shown, the interface comprises a web page 400 with a message area 402, a graph menu area 404 and a graph design area 403.

[0066] Users may be provided general instructions regarding the construction of stream processing graphs in message area 402, as well as links to enable used to learn more about stream concepts and primitives. A number of graphical icons may be provided as part of a stream processing graph toolset in menu area 404. For example, clients may be allowed to indicate, as inputs or outputs of various SPS processing stages, persistent streams 451, ephemeral streams 452, or connectors 453 to third-party processing environments. With respect to the SPS/SMS for which the web-based interface is implemented, a persistent stream 451 may be defined as a stream whose data records are stored on persistent storage devices such as disks, non-volatile RAMs, or SSDs, and an ephemeral stream 452 may be defined as one whose data records need not be stored at persistent storage devices. An ephemeral stream may be created, for example, from the output of an SPS stage that is expected to be consumed as input by a different SPS stage at which a best-effort recovery policy is to be implemented.

[0067] Two types of processing stages are supported in the example SPS graph construction web page 400: stages 455 in which checkpoint-based worker node recovery is used (e.g., each worker node saves progress records at intervals, and in the event of failure of a particular worker node, a replacement node refers to the failed node's progress records to determine which data records to start processing), and stages 456 in which best-effort recovery is used (e.g., replacement worker nodes do not refer to progress records, but simply start processing new data records as they are received). Details regarding the processing operations to be performed at each

stage may be entered by clicking on the corresponding icon in the graph construction area 403, as indicated by the instructions in message area 402. In addition to icons for streams, connectors, and processing stages, the menu area 404 also includes icon type 459 indicating third-party or external stream processing systems, and icon type 460 indicating nodes of a storage service that may be implemented at a provider network whose resources are being used for the processing stages.

[0068] In the example scenario shown in FIG. 4, a client has constructed a graph 405 comprising three processing stages 412, 415 and 416 within graph design area 403. Processing stage 412, which is configured to use checkpoint-based recovery, uses a persistent stream 411 as input. Output or results of the processing at stage 412 is sent to two destinations: in the form of a different persistent stream 413 that forms the input of stage 415, and in the form of an ephemeral stream 414 that forms the input of stage 416. Stages 415 and 416 both use best-effort recovery policies for their worker nodes. The output of stage 415 is sent in the form of an ephemeral stream to storage service node 419. The output of stage 415 is sent via a connector 417 to a third-party processing system 418. A "save graph" button 420 may be used to save a representation of the processing stage graph, e.g., in any appropriate format such as JSON (JavaScript Object Notation), XML (Extensible Markup Language) or YAML. Arbitrarily complex processing workflows may be constructed using tools similar to those shown in FIG. 4 in various embodiments. The workflows created using such tools may subsequently be activated, and such activations may result in invocations of SMS APIs – for example, to obtain data records for a processing stage such as stage 412 of FIG. 4, getIterator and/or getNextRecords interfaces may be invoked on stream 411.

[0069] FIG. 5 illustrates examples of programmatic record submission interfaces and record retrieval interfaces that may be implemented at an SMS, according to at least some embodiments. Data records, such as the illustrated DRs 110K and 110Q, may be submitted via various types of programmatic ingestion interfaces 510 to the SMS in the depicted embodiment. A DR 110 may comprise four types of elements in some embodiments: a stream identifier such as 501A (for stream "S1") or 501B (for stream "S2"), an indication of the data or body of the record, an optional partition key 504 (such as 504A or 504B), and an optional sequencing preference indicator 506 (such as sequencing preference indicators 506A and 506B). The data itself may be provided in-line in some data records (e.g., inline data 502 of DR 110K), while for other data records a pointer or address 503 may be provided, indicating to the SMS a network-accessible location (or an address at a local device that does not require network transfers). In some embodiments, a given stream may support both inline and by-reference (address-based)

data record submissions. In other embodiments, a given stream may require data producers to supply all the data inline or all the data by reference. In some implementations, a data record submission may include a partition identifier to be used for the record.

[0070] The incoming data records 110 may be directed to respective ingestion and/or storage nodes based on a partitioning policy in the depicted embodiment. Similarly, record retrieval may also be partition-based – e.g., one or more retrieval nodes may be designated for responding to read requests directed to records of a given partition. For some streams, data producers may be required to provide an explicit partition key with each data record write request. For other streams, the SMS may be able to distribute the data records according to a partitioning scheme that relies on metadata or attributes other than explicitly-supplied partition keys – for example, identification information pertaining to the submitting data producer may be used as a partition key, or a portion or all of the submitting data producer's IP address may be used, or a portion of the data being submitted may be used. In some implementations, for example, a hash function may be applied to a partition key to obtain an integer value of a certain size, such as a 128-bit integer. The total range of positive integers of that size (e.g., from 0 to 2^128-1) may be divided into N contiguous sub-ranges, with each sub-range representing a respective partition. Thus, in such an example, any given partition key determined or supplied for a data record would be hashed to a corresponding 128-bit integer, and the contiguous sub-range of 128-bit integers to which that integer belongs may indicate the partition to which the data record belongs. Further details about partitioning policies and their use are provided below with respect to FIG. 15.

[0071] The set of nodes responsible for ingesting or accepting the data records of the particular partition, storing the data records, and responding to read requests for the particular partition, are collectively referred to as ISR (ingestion, storage and retrieval) nodes for the partition in FIG. 5. The notation Sj-Pk is used to indicate the kth partition of stream Si. In the illustrated embodiment, ISR nodes 520A are configured for ingesting, storing and retrieving records of partition S1-P1, ISR nodes 520B are set up for records of partition S1-P2, ISR nodes 520C are set up for records of partition S1-P3, ISR nodes 520K are set up for records of partition S2-P1, and ISR nodes 520L are set up for records of partition S2-P2. In some embodiments, a give node of an ingestion subsystem, a storage subsystem, or a retrieval subsystem may be configured to handle data records of more than one partition (or more than one partition of more than one stream). In some embodiments, the records of a single partition of a given stream may be ingested, stored or retrieved by more than one node. The number of ingestion nodes designated for a given partition Sj-Pk may in at least some cases differ from the number of ingestion nodes designated for a different partition Sj-Pl, and may also differ from the number of

storage nodes designated for Sj-Pk and/or the number of retrieval nodes designated for Sj-Pk. With respect to ingestion and/or retrieval, SMS control nodes may implement APIs (such as getStreamInfo) in some embodiments to allow clients to determine which nodes are responsible for which partitions. The mappings between data records and partitions, and between partitions and ISR nodes (or control nodes) configured, may be modified over time, as described below in the discussion regarding dynamic repartitioning.

[0072] In some embodiments, several different programmatic interfaces 580 may be implemented for retrieving or reading stream data records from a given partition. As shown in FIG. 5, some retrieval interfaces 581 may be implemented for non-sequential accesses, such as getIterator (to instantiate an iterator or read cursor at or after a data record with a specified sequence number) or getRecord (to read a data record with a specified sequence number). Other retrieval interfaces 582 may be implemented for sequential retrieval, such as getNextRecords (an interface requesting that N records be read from the current position of an iterator, in order of increasing sequence number). In rotating disk-based storage systems, as mentioned earlier, sequential I/O may in many cases be much more efficient than random I/O, because the number of disk head seeks required on average per I/O may typically be much lower for sequential I/O than for random I/O. In many embodiments, the data records of a given partition may be written in sequence number order, and as a result sequential read requests based on sequence number ordering (e.g., using getNextRecords or a similar interface) may be much more efficient than random read requests. In at least some embodiments, therefore, different billing rates may be set for sequential versus non-sequential retrieval interfaces – for example, clients may be charged more for non-sequential reads.

Ingestion subsystem

[0073] FIG. 6 illustrates example elements of an ingestion subsystem 204 of an SMS, according to at least some embodiments. In the depicted embodiment, ingestion operations are logically divided into front-end and back-end functions, with the front-end functions involving interactions with data producers 120 (e.g., 120A, 120B or 120C), and back-end functions involving interactions with an SMS storage subsystem. Such a front-end/back-end split may have several advantages, such as enhancing the security of the storage subsystem and avoiding having to provide partitioning policy details to data producers. SMS client libraries 602 may be provided for installation at various data producers 120, and the data producers may invoke programmatic interfaces included in the libraries 602 to submit data for ingestion. For example, in one embodiment the data producers 120 may comprise logging or monitoring agents instantiated at hundreds or thousands of physical and/or virtual servers of a provider network. Such agents may

collect various log messages and/or metrics at their respective servers and periodically submit the collected messages or metrics to a front-end load distributor 604 endpoint instantiated by one or more ingestion control nodes 660 of the SMS. In some embodiments, one or more virtual IP addresses (VIPs) may be established for the load distributors, to which the data producers may submit the stream data. In one implementation, a round-robin DNS (Domain Name System) technique may be used for a VIP to select a particular load distributor from among several equivalently configured load distributors to which data is to be sent by data producers 120.

[0074] The received data records may be directed to any of several front-end nodes 606 (e.g., 606A, 606B or 606C) in the depicted embodiment. In at least some embodiments, the load distributor 604 may not be aware of the partitioning policy 650 in use for the data records, and the front-end node 606 may therefore be chosen for a given data record using round-robin load balancing (or some other general-purpose load balancing algorithm) rather than partition-based load balancing. The front-end nodes 606 may be aware of the partitioning policies 650 for various streams, and may interact with the ingestion control nodes 660 to obtain the identities of the specific back-end ingestion node 608 (e.g., 608A, 608B or 608C) that is configured for a given partition's data records. Thus, in the depicted embodiment, the front-end nodes 604 may each transmit data records to a plurality of back-end nodes 606, based on the respective partitions to which the data records belong. As noted earlier, the partition to which a data record belongs may be determined based on any combination of various factors, such as a partition key supplied by the data producer, one or more other attributes such as the identity or address of the data producer, or the contents of the data.

[0075] The back-end nodes 606 may each receive data records belonging to one or more partitions of one or more streams, and transmit the data records to one or more nodes of the storage subsystem. The back-end nodes may be referred to as "PUT servers" in some embodiments in which the data is submitted via HTTP (HyperText Transfer Protocol) "PUT" web service APIs. A given back-end node may determine the set of storage subsystem nodes to which its data records are to be transmitted by submitting a query to a control node 660 (which in turn may submit a corresponding query to a control node of the storage subsystem in embodiments in which control functions for the different subsystems are handled by separate sets of nodes).

[0076] In at least some embodiments, a number of different ingestion acknowledgement policies 652 may be supported, such as an at-least-once ingestion policy or a best-effort ingestion policy. In an at-least-once policy, the data producers 120 may require positive acknowledgements for each data record submitted, and may repeatedly submit the same data

record (if an acknowledgement of the first submission is not received) until an acknowledgement is eventually received. In the best-effort ingestion policy, positive acknowledgements may not be required for at least some data records submitted (although the ingestion subsystem may still provide occasional acknowledgements, or may respond to explicit requests for acknowledgements from the data producers). In some embodiments in which the ingestion subsystem 204 is required to provide acknowledgements to the data producers, the back-end ingestion node 608 responsible for a given data record may wait until the required number of replicas of the data records have been successfully created at the storage subsystem (e.g., in accordance with a persistence policy established for the stream), before generating an acknowledgement. In various embodiments, a sequence number may be generated by the ingestion subsystem for each data record received, e.g., indicative of the order in which that record was ingested relative to other records of the same partition or stream, and such a sequence number may be returned to the data producer as an acknowledgement, or as part of an acknowledgement. Further details regarding sequence numbers are provided below with reference to FIG. 13a and FIG. 13b. The acknowledgement and/or sequence number may be transmitted back to the data producer via a front-end node 606 in some implementations. In at least one implementation, the at-least-once policy may be implemented between the front-end and the back-end nodes of the ingestion subsystem itself – e.g., a given front-end node 606 may repeatedly submit a data record to the appropriate back-end node 608 until the back-end node provides an acknowledgement.

[0077] Ingestion control nodes 660 may be responsible for, among other functions, instantiating the front-end and back-end nodes, monitoring the health and workload levels of the nodes, orchestrating failovers as needed, providing responses to queries regarding which nodes are responsible for a given partition or to policy-related queries, for ingestion-related configuration operations resulting from dynamic repartitioning of streams. The number of ingestion control nodes designated for a given set of one or more streams may itself be changed over time in some embodiments, e.g., one or more master control nodes may be responsible for reconfiguring the control node pool as needed. In some embodiments in which redundancy groups are set up for ingestion front-end or back-end nodes, as described below in further detail with respect to FIG. 9 and FIG. 10, the control nodes 660 may be responsible for keeping track of which nodes are primaries and which are non-primary, for detecting the triggering conditions for failover, and for selecting replacements when failovers are required. It is noted that the multi-layered ingestion subsystem architecture illustrated in FIG. 6 may not be implemented in some embodiments, e.g., only a single set of ingestion nodes may be configured in some scenarios.

Storage subsystem

[0078] FIG. 7 illustrates example elements of a storage subsystem of an SMS, according to at least some embodiments. As shown, ingestion nodes 608 (e.g., back-end ingestion nodes in embodiments in which front-end and back-end ingestion responsibilities are handled by different sets of nodes) may transmit data records of one or more partitions of a stream to respective storage nodes 702 configured for those partitions. For example, data record 110A of partition S1-P1 is sent to storage node 702A, data record 110B of partition S2-P3 is sent to storage nodes 702B and 702C, data record 110C of partition S3-P7 is sent to storage node 702D, and data record 110D of partition S4-P5 is sent initially to storage node 702E. Storage control nodes 780 may be responsible for enforcing the persistence policies 750 that are applied to data records of the different streams, configuring and reconfiguring storage nodes as needed, monitoring storage node states, managing failovers, responding to storage configuration queries or storage policy queries, and various other administrative tasks in the depicted embodiment.

[0079] Persistence policies 750 may differ from one another in various ways in different embodiments. For example, a persistence policy P1 applied to stream Sj may differ from a policy P2 applied to stream Sk in (a) the number of replicas of each data record to be stored, (b) the type of storage device or system on which the replicas are to be stored (e.g., whether replicas are to be stored in volatile memory, non-volatile caches, rotating disk-based storage, solid-state drives (SSDs), storage appliances of various kinds, RAID (redundant arrays of inexpensive disks) of various kinds, in database management systems, at nodes of a storage service implemented by a provider network, and so forth), (c) the geographical distribution of the replicas (e.g., whether the stream data is to be made resilient to large-scale failures or certain types of disasters by placing replicas in different data centers), (d) the write acknowledgement protocol (e.g., if N replicas are to be stored, how many of the N copies have to be written successfully before an acknowledgement should be provided to the ingestion node), and/or (e) whether, in cases in which multiple replicas of data records are to be stored, the replicas should be created in parallel or sequentially. In some cases in which multiple replicas are to be stored, as in the case of data record 110D, a given storage node may transmit the data record to another storage node (e.g., storage node 702E sends data record 110D for further replication to storage node 702F, and storage node 702F sends it on to storage node 702G). In other cases in which a multiple-replica persistence policy is used, as in the case of data record 110B for which two in-memory replicas are to be stored, the ingestion node may initiate the multiple replications in parallel. In at least some embodiments, the client's chosen persistence policy may not specify the type of storage location to be used for stream data records; instead, the SMS may select the

appropriate types of storage technology and/or locations based on various criteria, such as cost, performance, proximity to data sources, durability requirements, and so on. In one embodiment, either the client or the SMS may decide to use different storage technologies or storage location types for different partitions of a given stream, or for different streams.

[0080] In the example shown in FIG. 7, the persistence policy applied to stream S1 (or at least partition S1-P1 of stream S1) is a single-replica in-memory policy, while for stream S2 a two-parallel-replica in-memory policy is applied. Accordingly, an in-memory replica 704A of data record 110A is created at storage node 702A, while two in-memory replicas 705A and 705B corresponding to data record 110B are created in parallel at storage nodes 702B and 702C. For stream S3's data record 110C, a single on-disk replica 706A is created. For stream S4, a sequential three-replica-on-disk policy is applicable, and as a result respective on-disk replicas 707A, 707B and 707C are created sequentially at storage nodes 702E, 702F and 702G. Various other types of persistence policies may be applied to data streams in different embodiments. Nodes of the retrieval subsystem may obtain the data records from the appropriate storage nodes in response to invocations of various types of retrieval APIs by data consumers.

Retrieval subsystem and processing stages

[0081] FIG. 8 illustrates example elements of a retrieval subsystem of an SMS and examples of interactions of the retrieval subsystem with an SPS, according to at least some embodiments. As shown, retrieval subsystem 206 may comprise a plurality of retrieval nodes 802, such as retrieval node 802A, 802B and 802C, as well as a collection of retrieval control nodes 880. Each of the retrieval nodes 802 may be configured to respond to stream data retrieval requests from various clients or data consumers 130, such as worker nodes 840 of an SPS as described below. A variety of programmatic retrieval interfaces 802 may be implemented by the retrieval nodes in different embodiments, such as the non-sequential and sequential retrieval interfaces described earlier. In some embodiments, web services APIs such as HTTP GET requests may be used for data record retrieval, and the retrieval nodes 802 may accordingly be referred to as GET servers. A given retrieval node 802 may be configured, e.g., by a retrieval control node 880, to obtain data records of one or more stream partitions in the depicted embodiment from the appropriate set of storage subsystem nodes 702, such as storage nodes 702A and 702B.

[0082] In the depicted embodiment, a retrieval node 802 may interact with one or more storage nodes 702, and also respond to retrieval requests received from one or more SPS worker nodes 840. For example, data records of partitions S4-P5 (e.g., data record 110K) and S5-P8 (e.g., data record 110L) are read from storage node 702A by retrieval node 802A, and provided to worker nodes 840A and 840K respectively. Data records of partition S6-P7, such as 110M, are

read by retrieval node 802B from storage node 702A and provided to worker node 840K. Data records of partition S4-P7 are read by retrieval node 802C from storage node 702B and provided to worker node 840B, and also to other data consumers 130 (e.g., data consumers that directly invoke SMS retrieval APIs instead of interacting with the SMS via an SPS).

[0083] In at least some embodiments, some or all of the retrieval nodes 802 may implement respective caches 804 (such as cache 804A at retrieval node 802A, cache 804B at retrieval node 802B, and cache 804C at retrieval node 802C) in which data records of various partitions may be retained temporarily in anticipation of future retrieval requests. Retrieval control nodes 880 may be responsible for implementing a number of retrieval policies 882, including for example caching policies (e.g., how large a cache should be configured for a given partition, how long data records should be cached), storage node selection policies (e.g., which particular storage node should be contacted first to obtain a given data record, in scenarios in which multiple replicas of data records are stored), and so on. In addition, retrieval control nodes may be responsible for instantiating and monitoring retrieval nodes 802, responding to queries regarding which retrieval nodes are responsible for which partitions, initiating or responding to re-partitioning operations, and so on.

[0084] In the illustrated example, SPS 290 comprises two processing stages, 215A and 215B. SPS control nodes 885 may be responsible for instantiating worker nodes 804 at the various processing stages 215, such as worker node 840A to process records of partition S4-P5, worker node 840B to process records of partition S4-P7, and worker node 840K to process records of partitions S5-P8 and S6-P7. The SPS control nodes 885 may implement programmatic interfaces (such as those illustrated in FIG. 3 and FIG. 4) enabling SPS clients to design processing workflows. Various checkpoint policies 850 may be implemented for different processing stages or workflows, indicating when or if worker nodes are to store progress records indicating how far along they are in processing their respective partitions, the types of storage devices to be used for the progress records, and so on. Failover/recovery policies 852 may indicate the triggering conditions or threshold that are to lead to replacing a worker node with a different node, and whether best-effort recovery is to be used or checkpoint-based recovery is to be used for a given processing stage. In at least some embodiments, the SPS control nodes 885 may interact with various types of SMS control nodes, e.g., to identify the retrieval nodes from which data records of a given stream are to be obtained, to establish new ephemeral or persistent streams that may be required for a particular processing workflow, and so on. In at least one embodiment, clients may interact with the SPS control nodes to instantiate streams – e.g., instead of utilizing SMS control interfaces, some clients may wish to invoke only higher-level SPS interfaces. It is noted

that although separate sets of control nodes are shown in FIG. 6, 7 and 8 for the SMS ingestion, storage, and retrieval subsystems, and for the SPS stages, in at least some embodiments a given control node may be used for several of the subsystems and/or the SPS.

Node redundancy groups

[0085] In at least some embodiments, redundant groups of nodes may be configured for one or more subsystems of an SMS. That is, instead of for example configuring one retrieval node for retrieving data records for a stream partition Sj-Pk, two or more nodes may be established for such retrievals, with one node being granted a "primary" or active role at a given point in time, while the other node or nodes are designated as "non-primary" nodes. The current primary node may be responsible for responding to work requests, e.g., requests received either from clients or from nodes of other subsystems. The non-primary node or nodes may remain dormant until a failover is triggered, e.g., due to a failure, loss of connectivity to the primary, or other triggering conditions, at which point a selected non-primary may be notified by a control node to take over the responsibilities of the previous primary. The primary role may thus be revoked from the current incumbent primary node during failover, and granted to a current non-primary node. In some embodiments, non-primary nodes may themselves take over as primary when a determination is made that a failover is to occur, e.g., explicit notifications may not be required. Such redundant groups of nodes may be set up for ingestion, storage, retrieval and/or control functions at an SMS in various embodiments, and a similar approach may also be taken for worker nodes at an SPS in at least some embodiments. Such groups comprising at least one primary node and at least one non-primary node for a given function may be referred to as "redundancy groups" or "replication groups" in some embodiments. It is noted that redundancy groups of storage nodes may be implemented independently of the number of physical copies of the data records that are stored – e.g., the number of replicas to be stored of a data record may be determined by a persistence policy, while the number of storage nodes that are configured for the corresponding partition may be determined based on redundancy group policies.

[0086] FIG. 9 illustrates examples of redundancy groups that may be set up for nodes of an SMS or an SPS, according to at least some embodiments. In the depicted embodiment, for a given stream partition Sj-Pk, respective redundancy groups (RGs) 905, 915, 925 and 935 are set up for ingestion nodes, storage nodes, retrieval nodes, and control nodes. A common RG 935 for control nodes is implemented in the illustrated embodiment, although separate RGs for ingestion control nodes, storage control nodes, or retrieval control nodes may be implemented in some embodiments. Each RG comprises a primary node (e.g., primary ingestion node 910A, primary storage node 920A, primary retrieval node 930A, and primary control node 940A) and at least

one non-primary node (e.g., non-primary ingestion node 910B, non-primary storage node 920B, non-primary retrieval node 920C, and non-primary retrieval node 920D). The primary role may be revoked and granted to a current non-primary in accordance with respective failover policies 912 (for ingestion nodes), 922 (for storage nodes), 932 (for retrieval nodes) and 942 (for control nodes). The failover policies may, for example, govern the triggering conditions that are to lead to a change in primary status, whether and how the health status of the primaries or non-primaries is to be monitored, the number of non-primaries that are to be configured in a given redundancy group, and so on. In at least some embodiments, a single RG may be established for multiple partitions – e.g., RG 905 may be responsible for handling ingestion of records of partition Sj-Pk as well as Sp-Pq. In some implementations, a node that is designated as primary for one partition may concurrently be designated as a non-primary for another partition. In one embodiment, multiple nodes may be designated concurrently as primary nodes within a given RG – e.g., the ingestion-related workload of a given partition may be distributed among two primary nodes, with one node designated as a non-primary in case of a failure at either primary. The number of nodes instantiated in a given RG may depend on the availability or resiliency level desired for the corresponding functions (e.g., on how many concurrent or overlapping failures the group is intended to be able to withstand). In some embodiments, in addition to or instead of being used for SMS nodes, redundancy groups may be set up for worker nodes of SPS processing stages. The members of a given RG may sometimes be distributed geographically, e.g., across several data centers, as illustrated in FIG. 10. Selected control nodes may be configured to detect failover-triggering conditions in some embodiments, e.g., using heartbeat mechanisms or other health monitoring techniques, and such control nodes may orchestrate the failover by selecting the appropriate non-primary node as the replacement for a failed primary, notifying/activating the selected replacement node, and so on.

[0087] In some embodiments a provider network may be organized into a plurality of geographical regions, and each region may include one or more availability containers, which may also be termed "availability zones" herein. An availability container in turn may comprise one or more distinct locations or data centers, engineered in such a way (e.g., with independent infrastructure components such as power-related equipment, cooling equipment, physical security components) that the resources in a given availability container are insulated from failures in other availability containers. A failure in one availability container may not be expected to result in a failure in any other availability container; thus, the availability profile of a resource instance or control server is intended to be independent of the availability profile of resource instances or control servers in a different availability container. Various types of

applications may be protected from failures at a single location by launching multiple application instances in respective availability containers, or (in the case of some SMSs and SPSs) distributing the nodes of a given redundancy group across multiple availability containers. At the same time, in some implementations, inexpensive and low latency network connectivity may be provided between resources (such as the hosts or compute instances used for SMS and SPS nodes) that reside within the same geographical region, and network transmissions between resources of the same availability container may be even faster. Some clients may wish to specify the locations at which their stream management or stream processing resources are reserved and/or instantiated, e.g., at either the region level, the availability container level, or a data center level, to maintain a desired degree of control of exactly where various components of their applications are run. Other clients may be less interested in the exact location where their resources are reserved or instantiated, as long as the resources meet the client requirements, e.g., for performance, high availability, and so on. Control nodes located in one availability container (or data center) may be able to remotely configure other SMS or SPS nodes in other availability containers (or other data centers) in some embodiments – that is, a particular availability container or data center may not need to have local control nodes to manage the SMS/SPS nodes.

[0088] FIG. 10 illustrates a provider network environment in which the nodes of a given redundancy group may be distributed among a plurality of data centers, according to at least some embodiments. Provider network 1002 comprises three availability containers 1003A, 1003B and 1003C in the depicted embodiment. Each availability container includes portions or all of one or more data centers – e.g., availability container 1003A comprises data centers 1005A and 1005B, availability container 1003B includes data center 1005C, and availability container 1003C includes data center 1005D. A number of different redundancy groups 1012 of SMS and/or SPS nodes are shown. Some RGs 1012 may be implemented entirely within a single data center, as in the case of RG 1012A located within data center 1005A. Other RGs may use resources of multiple data centers within a given availability container, such as RG 1012B, which spans data centers 1005A and 1005B of availability container 1003A. Yet other RGs may be implemented using resources spread across different availability containers. For example, RG 1012C uses resources located in data centers 1005B and 1005C of availability containers 1003A and 1003B respectively, and RG 1012D utilizes resources at data centers 1005B, 1005C and 1005D in availability containers 1003A, 1003B and 1003C respectively. In one example deployment, if RG 1012 comprises one primary and two non-primary nodes, each of the three nodes may be located in a different availability container, thus ensuring that at least one node is

highly likely to remain functional even if large-scale failure events occur at two different availability containers concurrently.

[0089] Console services 1078 and 1076, associated with the SMS and SPS respectively, may provide easy-to-use web-based interfaces for configuring stream-related settings in provider network 1002 in the depicted embodiment. A number of additional services, at least some of which may be used by the SMS and/or the SPS, may be implemented in provider network 1002 using resources spread over one or more data centers or across one or more availability containers. For example, a virtual computing service 1072 may be implemented, enabling clients to utilize selected amounts of computing power packaged as compute instances of various different capability levels, and such compute instances may be used to implement SMS and/or SPS nodes. One or more storage services 1070 may be implemented, enabling clients to store and access data objects with desired data durability levels, e.g., either via a block-device volume interface or via a web-services interface. The storage objects may be attachable to, or accessible from, the compute instances of service 1072, and may be used to implement various stream persistence policies at SMS storage subsystems in some embodiments. In one embodiment, one or more database services such as a high-performance key-value database management service 1074 or a relational database service may be implemented at the provider network 1002, and such a database service may be used for storing stream data records by SMNS storage subsystems, and/or for storing metadata of control subsystems, ingestion subsystems, storage subsystems, retrieval subsystems, or processing stages.

Stream security options

[0090] In at least some embodiments, the users of the SMS and/or the SPS may be provided a number of security-related options for data streams, enabling clients to select the security profiles of resources (e.g., virtual or physical machines) to be used for the various functional categories such as ingestion, storage, retrieval, processing and/or control. Such options may include, for example, choices regarding the types of physical locations of the resources used for various nodes (e.g., whether provider network facilities are to be used, or client-owned facilities are to be used, which may have different security characteristics than provider network facilities), choices regarding encryption of stream data, and/or network isolation choices in various parts of the stream-handling infrastructure. Some clients may be concerned about the possibility of intruders or attackers obtaining access to valuable proprietary business logic or algorithms, for example, and may wish to implement stream processing worker nodes using computing devices within client-owned promises. The types of resources to be used for implementing a set of SMS and/or SPS nodes may be referred to herein as the "placement

destination types" for those nodes. FIG. 11 illustrates a plurality of placement destination types that may be selected for nodes of an SMS or an SPS, according to at least some embodiments.

[0091] Placement destinations may be selected within provider network 1102 for some types of SMS/SPS functional categories (e.g., ingestion, storage, retrieval, control or processing), and outside provider network 1102 for other types of SMS/SPS functional categories in the depicted embodiment. Within provider network 1102, some resources such as compute instances, storage instances, or database instances may be implemented using multi-tenant instance hosts 1103. Such multi-tenant instance hosts, at each of which SMS or SPS nodes for one or more clients may be instantiated, may form a first category "A" of placement destination types. To avoid having to share physical resources with other clients, some clients may request that their SMS/SPS nodes be implemented using instance hosts restricted to a single client. Such single-tenant instance hosts may form placement category type "B". Single-tenant instance hosts may be preferable from the perspective of some clients for several reasons. As multi-tenant instance hosts may include compute instances belonging to other clients, there may be a higher probability of security attacks from another client's instances in multi-tenant instance hosts than in single-tenant instance hosts. In addition, the "noisy-neighbor" phenomenon, in which one client's compute instance CI1 running on multi-tenant host experiences a surge in workload and starts consuming a large proportion of the host's compute cycles or other resources, thus potentially impacting the performance of another client's applications running on a different compute instance CI2, may also be avoided when single-tenant instance hosts are used.

[0092] Isolated virtual networks (IVNs) 1106, such as IVN 1106A and 1106B may represent another category "C" of placement destination types in the depicted embodiment. An IVN 1106 may be created at the request of a provider network client in some embodiments as the logical equivalent of a private network, built using provider network resources but with network configuration being controlled largely by the client. For example, the client may decide the IP addresses to be used within an IVN 1106, without having to be concerned about the possibility of duplicating IP addresses that may already be in used outside the IVN. Implementing various types of SMS and SPS nodes in one or more IVNs may add an extra level of network security to the management and/or processing of a client's stream data in the depicted embodiment. In some cases, a given client may wish to place one functional category of SMS/SPS nodes in one IVN 1106, and a different functional category in a different IVN. A given IVN 1106 may comprise either single-tenant instance hosts, multi-tenant instance hosts, or both types of instance hosts in various embodiments. In some embodiments, another set of placement destination type choices (or security profile choices) using resources of the provider network, not shown in FIG. 11, may

be available to at least some clients. In embodiments in which clients can acquire and use compute instances from a provider network's virtualized computing service for stream-related operations, the compute instances may be used in one of two modes. In one mode, a client may provide, to an SPS or an SMS, the executable program or programs to be run at compute instances configured as SPS worker nodes (or at ingestion, storage or retrieval nodes), and let the SMS or SPS run the programs and manage the nodes. This first mode may be referred to as a "stream service managed" mode of using compute instances for stream operations. In the other mode, a client may wish to run the executable programs and manage the compute instances, with less support from the SPS or SMS. This second mode may be referred to as a "client-managed" mode of using compute instances for stream operations. These two modes of operation may thus represent additional choices with respect to client-selectable placement destination types or security profiles. A client may opt for the client-managed mode if, for example, the executable program is likely to require debugging (including single-stepping) that can best be performed by subject-matter experts from the client's organization, while the stream-service-managed mode may be a reasonable choice for more mature code that is not likely to require debugging. In some embodiments, different pricing policies may apply to these two modes.

[0093] A number of placement options may be supported at facilities external to the provider network in the embodiment shown in FIG. 11. For example, hosts 1160 on which SMS libraries 1171 and/or SPS libraries 1172 are installed may be used for stream management or processing from within client facilities (e.g., client-owned data centers or premises) 1110A or 1110B, with the two types of client facilities differing in their manner of connectivity to the provider network. Client facility 1110A is linked to provider network 1102 via at least some shared Internet links 1151 (i.e., the network traffic of other entities may also flow over some of the links between client facility 1110A and the provider network 1102). In contrast, some client facilities (such as 1110B) may be linked to the provider network via special unshared dedicated physical links 1106 (which may sometimes be referred to as "direct connect" links). These two different types of client premises comprise placement destination options "D" and "E" respectively in the terminology used in FIG. 11. In some embodiments, portions of the SMS and/or SPS may also be implementable at third-party facilities (e.g., data centers used but not owned or managed by clients of the SMS/SPS), and such third-party premises may be designated as placement destination type "F". In at least some of the client and/or third-party premises, the SMS and/or SPS libraries may have to be obtained from the provider network and installed on the hosts to be used for the SMS/SPS nodes. In at least one embodiment, nodes of all the different functional

categories may be implemented externally to the provider network with the help of the appropriate libraries.

The different placement destination types may differ from one another in various security-related aspects in different embodiments, such as the network isolation features implemented, intrusion detection functionality supported, physical security policies implemented, supported encryption levels, and so on. Accordingly, each of the various destination types may be considered to have a respective security profile, which may differ from the security profile of the other placement destinations in one or more ways. In some embodiments, clients of the SMS and/or SPS may select respective placement destination types for different subsystems or node sets programmatically, e.g., by sending a request to one or more control nodes of the SMS or SPS, as illustrated in FIG.12a and 12b. It is noted that in some embodiments and for certain types of stream applications, clients may wish to control placement destination types not just for security reasons, but also for performance and/or functionality reasons. For example, the noisy-neighbor phenomenon described above may be avoided by using dedicated client-premise resources or single-tenant instance hosts. In some embodiments, clients may have special-purpose or proprietary hardware and/or software that they wish to use for SPS stages or SMS nodes, where the functional capabilities or performance levels achievable using such components cannot easily be replicated at a provider network, or are simply not supported at the provider network. A client may have access at an external data center to a computer server with supercomputer-level processing capabilities, for example, which may be able to perform SPS processing at a much higher rate than would be possible using provider network resources alone. Enabling a client to select the placement destinations for various nodes may allow such special-purpose devices or software to be used.

[0094] FIG. 12a and 12b illustrate examples of security option requests that may be submitted by SPS clients and SMS clients, respectively, according to at least some embodiments. FIG. 12a illustrates an SPS security option request 1200 in which a client indicates, for one or more processing stages with identifiers 1210, the placement destinations types (PDTs) requested for control nodes of the stage (element 1212), and the PDTs requested for worker nodes (element 1214). In at least one embodiment, clients may also be able to submit requests to configure encryption settings for their stream data records or stream processing results, e.g., by requesting that data records be encrypted using a specified algorithm or protocol prior to their transmission over various network links, or that various control or administrative interactions be encrypted. For example, in FIG. 12a, the encryption settings for the stage may indicate encryption techniques to be applied to the results of the stages processing operations, and/or the encryption

used for the communications between the control nodes of the stage and the worker nodes of the stage.

[0095] Similarly, in Fig. 12b, a client's SMS security option request 1250 comprises a number of elements that indicate the client's security preferences for one or more streams with specified identifiers 1252. Placement destination type preferences for ingestion nodes, storage nodes, and retrieval nodes may be indicated in elements 1254, 1258 and 1262 respectively. PDT preferences for ingestion control nodes, storage control nodes and retrieval control nodes may be indicated by elements 1256, 1260 and 1264 respectively. Encryption preferences for data records, e.g. whether and/or how encryption is to be implemented for the data records as they are transmitted from one category of node to another, may be indicated via element 1266. Using security option requests such as those shown in FIG. 12a and 12b, clients may be able to choose the locations (e.g., within the provider network or external to the provider network) and various other security profile components for different parts of their stream management and processing environment.

[0096] It is noted that the choice of node placement destinations may be offered for other reasons than security in at least some embodiments. For example, a client may wish to have some types of SMS or SPS nodes implemented at single-tenant hosts for performance reasons (e.g., to avoid the "noisy-neighbor" problems indicated earlier rather than primarily for security reasons. Placement choices may be changed in at least some embodiments during the lifetime of a stream – e.g., a client may initially allow SMS nodes to be instantiated at multi-tenant instance hosts, but may wish to move at least some subset of the nodes to single-tenant instance hosts later. Different pricing policies may be applied to the different security-related options in at least some embodiments – e.g., it may cost more to implement SMS nodes of a particular functional category at a IVN than at multi-tenant instance hosts outside IVNs, or it may cost more to implement SMS nodes at single-tenant instance hosts than at multi-tenant instance hosts.

Sequential storage and retrieval of stream records

[0097] For many types of stream applications, data records may be received at the SMS at very high rates from a plurality of data producers 120, and data consumers may typically wish to access stored data records in the order in which the records were generated. Especially in environments in which rotating magnetic disks are used as the storage devices for stream data records, as mentioned earlier, sequential I/O access patterns (for both reads and writes) may have significant performance advantages over random I/O access patterns. In several embodiments, stream-specific or partition-specific sequence numbers may be assigned to data records as they are received by the SMS, and sequential retrieval operations based on sequence numbers may be

supported. FIG. 13a illustrates example interactions between a stream data producer and an ingestion subsystem of an SMS, according to at least some embodiments. The stream data producer may submit a data record 110 to an ingestion subsystem, and in the depicted embodiment, the ingestion subsystem may respond with a sequence number 102 that has been chosen for the submitted record. In at least some embodiments, an ingestion node may obtain a portion of the sequence number from the storage subsystem – e.g., the sequence number 102 may be determined subsequent to the storage of the received data record in accordance with the applicable persistence policy in such embodiments, and the storage subsystem may generate a numerical sequence indicator of its own for the data record and provide that indicator for inclusion in the larger sequence number assigned to the data record by the ingestion node..

[0098] Sequence numbers may be implemented in various embodiments to provide a stable, consistent ordering of data records, and to enable repeatable iteration over records by data consumers. Sequence numbers assigned to the data records of a particular partition may increase monotonically over time, although they need not be consecutive in at least some implementations. In various embodiments, sequence numbers may be assigned with at least some subset of the following semantics: (a) sequence numbers are unique within a stream, i.e., no two data records of a given stream may be assigned the same sequence number; (b) sequence numbers may serve as indexes into the stream's data records, and may be used to iterate over data records within a given stream partition; (c) for any given data producer, the order in which the data producer successfully submitted data records is reflected in the sequence numbers assigned to the data records; and (d) sequence numbering for data records with a given partition key value retain the monotonically increasing semantics across dynamic repartitioning operations – e.g., the sequence numbers assigned to data records with a partition key value K1 after a repartitioning may each be larger than any of the sequence numbers that were assigned to data records with that partition key value K1 prior to the dynamic repartitioning. (Dynamic repartitioning is described in further detail below with respect to FIG. 16.)

[0099] In some embodiments, a data producer may wish to influence the selection of the sequence number 102 selected for at least some data records. For example, a data producer 120 may wish to demarcate boundaries or separators within the assigned sequence numbers of a stream, so that it becomes easier for data consumers of that stream to submit read requests targeted at particular subsets of the stream. In some implementations, the data producer 120 may submit an indication of a minimum sequence number together with a record, and the SMS may select a sequence number in accordance with the requested minimum that also conforms to the sequence number semantics discussed above.

[00100] FIG. 13b illustrates example elements of a sequence number that may be generated for an ingested data record at an SMS, according to at least some embodiments. The sequence number may comprise four elements in the depicted embodiment: an n1-bit SMS version number 1302, an n2-bit timestamp or epoch value 1304, an n3-bit subsequence number 1306, and an n4-bit partition number 1308. In some implementations, 128 bit sequence numbers may be used, e.g., n1, n2, n3 and n4 may be 4, 44, 64 and 16 bits respectively. The version number 1302 may be used simply to avoid confusion across SMS software version rollouts, e.g., so that it is easy to tell which version of the SMS software was used to generate the sequence number. Version number 1302 may not be expected to change frequently in at least some implementations. The timestamp value 1304 may be obtained, for example, from a local clock source or a globally accessible clock source (e.g., a state management system of a provider network that implements a getCurrentEpoch or getCurrentTime API) by an ingestion subsystem node. In at least some implementations, an offset from a well-known point in time (e.g., the number of seconds that have elapsed since 00:00:00 AM UTC on January 1, 1970, which can be obtained by invoking various time-related system calls in Unix™-based operating systems) may be used for the timestamp value 1304. In some embodiments, the subsequence number 1036 may be generated by the storage subsystem and may indicate the order in which data records of a particular partition are written to a storage device. Thus, in an implementation in which numerous data records are received within a given second and the timestamp values 1304 only change at approximately one-second intervals, the subsequence numbers 1306 may serve as indicators of the record arrival (or storage) order for data records that happen to have arrived within the same second and therefore are assigned the same timestamp value. The partition number 1308 may uniquely identify a partition within a given stream in some embodiments. In at least some implementations in which the sequence number timestamps indicate (at least approximately) the clock times at which the corresponding data records were ingested, the sequence numbers may be used for an indexing mechanism for certain types of time-based retrieval requests. For example, a client may wish to retrieve stream records generated or ingested on a particular day or during a specified time range, and the sequence numbers may be used as keys of an implicit secondary index to retrieve the appropriate set of data records. Thus, in at least some embodiments, the use of sequence numbers that contain timestamps for ordered storage and retrieval may have an additional benefit of providing a temporal index into the set of stored data records.

[00101] Data records of a given partition may typically be written (e.g., to disk) in sequence number order, often using large sequential write operations. In some embodiments, as indicated

earlier, iterator-based programmatic interfaces may be implemented to allow data consumers to read data records in sequence number order. FIG. 14 illustrates examples of ordered storage and retrieval of stream data records at an SMS, according to at least some embodiments. Six data records 110A – 110F of a partition Sj-Pk (the kth partition of a stream Sj) are shown stored in sequence number order. As illustrated, the sequence numbers may not be consecutive in at least some embodiments, e.g., because the manner in which the values are assigned to the timestamp portions 1304 or the subsequence numbers 1306 discussed above may not always result in consecutive values for those elements.

[00102] In the example shown in FIG. 14, a data consumer has requested an iterator to be created, specifying a starting sequence number "865". In response to the request, the SMS has initialized Iterator1, positioned at the data record with the nearest sequence number that is higher than or equal to the requested starting sequence number. In this case, data record 110C with sequence number 870 has been selected as the iterator's starting position, as the next lower sequence (860, assigned to data record 110B) is smaller than the starting sequence number in the consumer's request. The getIterator interface may be considered the logical equivalent of a request to set a cursor at a requested position within the partition, and the getNextRecords interface may be used to then read data records starting from the cursor position, e.g., to move the cursor along the stream in sequence number order. In the illustrated example, a data consumer has invoked the getNextRecords interface with parameter "iterator" set to Iterator1 and "maxNumRecords" (the maximum number of data records to return) set to 3. Accordingly, the SMS retrieval subsystem returns the data records 110C, 110D and 110E in that order to the data consumer. The iterator Iterator1 may be moved to a new position, e.g., to data record 110F, after the getNextRecords call completes, and subsequent getNextRecord invocations for the same iterator may return data records starting with 110F. The semantics of the getIterator call may differ in some embodiments – e.g., instead of positioning the iterator at the data record with the nearest sequence number higher than or equal to the specified sequenced number, the iterator may be positioned at the nearest data record with highest sequence number equal to or lower than the requested sequence number in some embodiments. In another embodiment, clients may have to specify an existing sequence number in the getIterator call – e.g., an error may be returned if a record with the requested sequence number doesn't exist in the stream.

Partition mappings

[00103] As described earlier, the workload related to ingestion, storage, retrieval and processing of the records of a given stream may be subdivided and distributed among several nodes in various embodiments in accordance with various partitioning and repartitioning

policies. FIG. 15 illustrates an example of a stream partition mapping 1501 and corresponding configuration decisions that may be made for SMS and SPS nodes, according to at least some embodiments. When a particular data stream is created or initialized, e.g., in response to a client's invocation of a createStream API, a partitioning policy may be activated for the stream, which may be used to determine the partition of which any given data record of the stream is to be considered a member. The particular nodes of the ingestion subsystem 204, the storage subsystem 206, the retrieval subsystem 208 and any relevant SPS stages 215 that are to perform operations for a given data record may be selected on the basis of the record's partition. In one embodiment, at least a subset of the control nodes used for a given data record may be selected based on the partition as well. In at least some embodiments, dynamic repartitioning of a data stream may be supported as part of the partitioning policy, e.g., in response to triggering conditions indicated in the policy or in response to explicit requests.

[00104] In various embodiments, the partition selected for a given data record may be dependent on a partitioning key for the record, whose value may be supplied by the data producer either directly (e.g., as a parameter of a write or put request), or indirectly (e.g., the SMS may use metadata such as the identifier or name of the data producer client, an IP address of the data producer, or portions of the actual contents of the data record as a partition key). One or more mapping functions 1506 may be applied to the data record partition key or attribute 1502 to determine the data record partition identifier 1510 in the embodiment shown in FIG. 15. In one implementation, for example, a given partition identifier 1510 may represent a contiguous range over the space of 128-bit integer values, such that the union of the ranges for all the partitions of the stream may cover all possible values a 128-bit integer can assume. In such an example scenario, one simple mapping function 1506 may generate a 128-bit hash value from the partition key value(s) or selected attribute value(s) of the data record, and the partition identifier may be determined based on the particular contiguous range within which the hash value happens to lie. In some implementations, the contiguous ranges may at least initially be equal in size; in other implementations, different partitions may correspond to contiguous ranges that may differ in size from one another. Repartitioning may also result in adjustments to the range boundaries in one implementation. Other partitioning functions 106 may be used in different implementations.

[00105] If the data stream undergoes dynamic repartitioning (as discussed below in further detail), the partition to which records with a particular key are mapped may change. Thus, in at least some embodiments, SMS and/or SPS control nodes may have to keep track of several different mappings that apply to a stream during the lifetime of the stream. In some

embodiments, metadata such as a timestamp validity range 1511 or a sequence number validity range may be stored by the control nodes for each partition mapping. The timestamp validity range 1511 may, for example, indicate that a particular mapping M1 applies from the stream's creation time until time T1, that a different mapping M2 applies from T1 to T2, and so on. When responding to read requests directed at a stream, the retrieval nodes may have to first determine which mapping is to be used (depending for example on the sequence number indicated in a read request), and then use that mapping to identify the appropriate storage nodes.

[00106] The SMS and SPS control nodes may be responsible for mapping partitions to resources at several different granularities in at least some embodiments. For example, as shown in example implementations 1599 of FIG. 15, in one implementation, each ingestion, storage, retrieval or processing (worker) node may be implemented as a respective process or a respective thread of execution within a server virtual machine such as a Java™ Virtual Machine (JVM) or a compute instance, and each JVM or compute instance may be instantiated at a particular physical host. In some embodiments, multiple JVMs may be launched within a single compute instance, adding another layer of resource mapping decisions. Thus, for a given partition, one or more control nodes may select which particular resources are to be used as ingestion nodes 1515, storage nodes 1520, retrieval nodes 1525, or processing stage worker nodes 1530 (e.g., nodes 1530A or 1530B for stages PS1 or PS2 respectively). The control nodes may also determine the mappings of those nodes to servers (such as ingestion servers 1535, storage servers 1540, retrieval servers 1545, or processing servers 1550), and the mappings between servers and hosts (such as ingestion hosts 1555, storage hosts 1560, retrieval hosts 1565 or SPS hosts 1570A/1570B). In some implementations, a partition mapping may be considered to comprise identification information (e.g., resource identifiers) at each of various resource granularities (e.g., node, server and host granularities) illustrated, an indication of the data record attributes being used as input to the function or functions 1506, as well as the functions 1506 themselves. The control servers may store representations of the partition mapping in a metadata store, and in some embodiments may expose various APIs (such as getPartitionInfo APIs) or other programmatic interfaces to provide the mapping information to data producers, data consumers, or to the nodes of the SMS subsystems or the SPS.

[00107] The mappings of data records to partitions, and from the partitions to the resources, may be further complicated in some embodiments by various factors such as: (a) a given node, server or host may be designated responsible for multiple partitions in some embodiments, or (b) failures or other triggers may result in new nodes, servers or hosts being assigned to a given partition or set of partitions. In addition, as indicated above and described below, partition

mappings for a given stream may be modified dynamically over time while the stream records continue to be handled by the SMS and/or SPS nodes. As a result several versions of mapping metadata may be retained for a given stream at least temporarily in some embodiments, each corresponding to a different period of time.

Dynamic stream repartitioning

[00108] FIG. 16 illustrates an example of dynamic stream repartitioning, according to at least some embodiments. At time T1 of the timeline illustrated in FIG. 16, a stream S1 is created or initialized. A partition mapping PM1 is created for the stream S1, and remains in effect during the time interval T1 through T2. Three data records received by an SMS between T1 and T2 are shown by way of example. Data record 110A (DR110A) is submitted with a client-supplied partition key value "Alice", DR110B is submitted with a client-supplied partition key value "Bill" and DR110C is submitted with a client-supplied partition key value "Charlie". In the initial mapping PM1, all three data records 110A, 110B and 110C are mapped to the same partition with a partition identifier "P1". For P1 data records, a single node I1 is configured to handle ingestion, a single node S1 is configured to handle storage, a single node R1 is configured to handle retrieval, and a single worker node W1 is configured to handle SPS processing. The start timestamp for a validity range of the mapping PM1 is set to T1.

[00109] At time T2, stream S1 is dynamically repartitioned in the example timeline of FIG. 16. Data records continue to arrive and be handled by the SMS and the SPS in the depicted embodiment, irrespective of when the repartitioning occurs; neither the SMS nor the SPS need to be taken offline. The repartitioning may be initiated as a result of any of a number of factors – e.g., in response to a detection of an overload condition at an ingestion, storage, retrieval or processing node, in response to a detection of a skew or imbalance between workload levels at different hosts of the various subsystems, or in response to a request from a data consumer or a data producer client. In the depicted embodiment, a new mapping PM2 takes effect at time T2 (or shortly after T2), as indicated by the validity range start timestamp setting shown for PM2. In at least some implementations, a different set of data record attributes may be used for partitioning data records than were used before the repartitioning. In some cases, an additional partitioning attribute may be submitted by the data producer (e.g., at the request of the SMS), while in other cases the additional attribute may be generated by an SMS ingestion node. Such additional attributes may be referred to as "salted" attributes, and the technique of using additional attributes for repartitioning may be referred to as "salting". In one example implementation, an overloaded ingestion server may indicate to a data producer (e.g., to the SMS client library code being executed by the data producer) that, for repartitioning, a randomly

selected small integer value be provided in additional to the previously-used partition key. The combination of the original partition key and the salted additional integer may subsequently be used to distribute the ingestion workload among a different set of ingestion nodes. In some embodiments, the retrieval nodes and/or data consumers may have to be informed regarding the additional attributes being used for repartitioning. Such additional attributes may not be used for repartitioning in at least some implementations.

[00110] In the embodiment shown in FIG. 16, the new partition mapping results in different partitions being selected for at least some of the data records received after T2, relative to the partition selected for the same key before T2. DR110P is submitted after T2 with the partition key value "Alice", DR110Q is submitted after T2 with the partition key value "Bill", and DR110R is submitted after T2 with the partition key value "Charlie". Using the PM2 mapping, DR110P is designated a member of partition "P4", DR110Q is designated a member of partition "P5", while DR110R is designated a member of partition "P6" in the illustrated example scenario. In the depicted embodiment, none of the example data records shown as being received after T2 are designated as members of the previously-used partition "P1"; instead, completely new partitions may be used after the repartitioning. In some embodiments, at least some previously used partitions may continue to be used after repartitioning. For each of the new partitions P4, P5 and P6, different nodes may be designated for ingestion, storage, retrieval and/or processing. For example, nodes I4, S4, R4 and W4 may be configured for partition P4, nodes I5, S5, R5 and P5 may be configured for partition P5, and nodes I6, S6, R6 and P6 may be configured for partition P6. In some embodiments, the same storage node may be used for a record with a particular partition key or attribute after repartitioning as was used for such records before repartitioning, but a different storage location within that node (e.g., a different disk, a different disk partition, or a different SSD) may be used after the repartitioning.

[00111] During at least some time period after the dynamic repartitioning at T2, retrieval requests may continue to be retrieved for data records that were processed by the SMS ingestion and/or storage subsystems prior to the repartitioning. In at least some cases, the requested data records may have to be retrieved based on the PM1 mapping which was in effect at the time that the data records were ingested. Accordingly, as indicated in FIG. 16, for the purposes of data retrieval, both PM1 and PM2 may continue to be used for some time after T2. In at least some implementations, data records may eventually be deleted from the stream as they age, and the older partition mappings may also be discarded eventually, e.g., when all the corresponding data records have themselves been deleted. In some embodiments, instead of (or prior to) being deleted, stream records may be archived (e.g., based on client-selected archival policies) to a

different set of storage locations or devices, such that the partition mappings used by the SMS may still be usable to retrieve the records after archival. In such embodiments, partition mappings such as PM1 and PM2 may be retained for as long as they are needed to support retrieval requests directed to the archival storage. In some archival implementations, different retrieval approaches may be used that do not require the stream partition mappings to be retained (e.g., new indexes may be created for the archived data records). In some embodiments a partition such as P2 that was being used prior to a repartitioning, but to which writes are no longer directed after the repartitioning, may at some point after the repartitioning be "closed" for reads – e.g., the equivalent of an "end of partition reached" error message may be provided in response to retrieval requests.

[00112] In some implementations, a given data stream may be divided into numerous (e.g., hundreds or thousands) of partitions. Consider an example case in which a stream S1 is initially divided into 1000 partitions, P1, P2, …, P1000. In the event that an overload condition corresponding to one partition, say P7, is detected, it may be worthwhile to change the initial mapping of data records to P7, but the mapping of the other partitions need not need to be changed. In one approach, two new partitions P1001 and P1002 may be created via a repartitioning operation. Records received after the repartitioning, whose attributes would originally (i.e., on the basis of the original mapping) have resulted in their membership in P7, may be mapped to either P1001 or P1002 after the repartitioning, thus distributing the workload of P7 among two partitions. The remaining partitions, e.g., P1 – P6 and P8 – P1000, may not need to be modified. As only a small subset of partitions are affected by such a repartitioning, in at least some embodiments a combined data structure such as a directed acyclic graph of partition entries (or a tree of partition entries) may be generated and stored. Each entry may indicate a partitioning function output range, and a validity time range (the time period during which the entry's partitioning information is to be considered valid). Assume, in the example above, that the repartitioning involving P7 was performed at time T2, while the stream S1 (and its initial mapping) was created at time T1. In such a scenario, the validity time period for the entry regarding P7 would be "T1 toT2", the validity time periods for P1001 and P1002 would be "T2 onwards", and the validity time time periods for the remaining partitions would be "T1 onwards". Using such a combined data structure may lead to a substantial reduction in the amount of memory or storage used for partition mapping metadata in at least some implementations. In the above example, a split of partition P7 into two new partitions was discussed. In at least some implementations, partitions may also be merged during repartitioning – e.g., two adjacent partitions for which relatively few retrieval requests were received, or

relatively few records were submitted, may be merged into a single partition. For any given point in time, the partition to which a data record belongs may be determined unambiguously using the partitioning function and the validity time range information. Over time, the combined data structure may evolve as more splits and/or merges are performed, but the total space required for the partitioning metadata may (depending of course on how often splits occur, and how many partitions are affected by the splits on average) not increase dramatically. In contrast, in a different implementation, each time a repartitioning occurs, the entire set of unchanged metadata for a stream may be replicated and combined with entries for the partitions affected by repartitioning. The storage and memory requirements for partition mapping metadata may increase at a much faster rate in the latter implementation, especially if the older mappings may have to be retained for at least some time after repartitioning as described above.

[00113] In at least some embodiments in which sequence numbers that comprise timestamp values (such as the timestamp value 1304 shown in FIG. 13b) are used, a special type of sequence number transition may be implemented for dynamic repartitioning. Assume by way of example that a timestamp-based sequence number scheme, similar to that shown in FIG. 13b, is being used for a stream S1, in which new timestamp values are generated every second for inclusion in the sequence numbers. In at least some implementations in which dynamic repartitioning is supported, the sequence numbers assigned after the dynamic repartitioning may all use a different set of timestamp values (starting with a selected initial timestamp value corresponding to the repartition event) than were used before the dynamic repartitioning. For example, if the timestamp value in use at the time the dynamic repartitioning is committed (i.e., put into effect) was Tk, any new sequence numbers issued after the commit may be required to use timestamp values Tk+1 onwards. Since sequence number values encode the timestamp value in at least some of their higher-order bits in the scheme used in FIG. 13b, ensuring that repartition events correspond to timestamp boundaries as described may in turn simplify the bookkeeping involved in identifying the mappings to be used in response to a retrieval request. Thus, in such implementations, when a retrieval request specifying a particular sequence number is received, the timestamp value may be extracted from that sequence number, and it may be easily determined whether the post-repartitioning mapping should be used, or the pre-repartitioning mapping should be used. If the extracted timestamp value is lower than the initial timestamp selected for the repartition, the pre-repartitioning mapping may be used, and if the extracted timestamp value is equal to or higher than the initial timestamp value selected for the repartition, the post-repartitioning mapping may be used.

Methods for stream management and processing

[00114] FIG. 17 is a flow diagram illustrating aspects of operations that may be performed to support respective sets of programmatic interfaces for stream record ingestion and stream record retrieval, according to at least some embodiments. As shown in element 1701, a request to create or initialize a data stream may be received, e.g., from an SMS client or a data producer client. The initial partition mapping to be used for the stream may be determined (element 1704), e.g., the function(s) to be used to identify the partition to which a particular data record belongs, and the input parameters to be used for the function(s), may be identified based on a partitioning policy. As mentioned earlier, control components of the SMS may be responsible for receiving and responding to stream creation requests in various embodiments. The manner in which stream creation and initialization (as well as other control-plane operations) are implemented may differ from one embodiment to another. In one embodiment, for example, a redundancy group of control servers may be established, and the primary control server of that redundancy group may respond to a stream creation request by generating and storing the appropriate metadata for a new stream (e.g., the initial partition mapping, the initial sets of nodes of the ingestion, storage and retrieval, and so on) in a persistent storage location. Responses to subsequent queries regarding the stream (e.g., a request from a front-end ingestion node regarding the back-end node responsible for a given partition) may be generated by the primary control server using the stored metadata. In another implementation of the SMS control-plane functionality, stream configuration metadata may be stored in a database that is directly accessible by at least some nodes of the ingestion, storage, or retrieval subsystems. After a stream has been created and initialized, data-plane operations such as record submission, storage and retrieval may commence, and may be handled by respective components of the corresponding subsystems, typically without additional interactions with the control components.

[00115] In some embodiments, data producers may be required to submit explicit partition keys with write requests, while in other embodiments, the inputs to be used for the partitioning functions may be determined based on metadata associated with the write requests, such as the identity of the data producers, the IP addresses from which the data records are received, or from the contents of the data records themselves. In at least one implementation, clients may optionally supply partition identifiers in the data record submissions, and additional partitioning functions may not be required in such an implementation.

[00116] A number of different factors may be taken into account when determining or configuring the initial set of nodes for ingestion, storage and retrieval functions for the stream (element 1707). For example, the partition mapping itself (which may determine how many

partitions the stream is divided into, and the relative expected sizes of the partitions), information about the expected ingestion rates and/or retrieval rates if such information is available, durability/persistence requirements for the stream data records, and/or high availability requirements for the various subsystems (which may result in the setting up of redundancy groups similar to those illustrated in FIG. 9 and 10) may influence the number and placement of the nodes of the different subsystems. In addition, in embodiments in which clients may indicate placement destination type preferences for various categories of nodes (as illustrated in FIG. 11, 12a and 12b), such preferences may also play a role in determining the resources to be used for the SMS and/or SPS nodes. In at least some embodiments, respective pools of nodes capable of performing ingestion, storage and/or retrieval functions may be set up in advance, and control components may assign selected members of such pools to each new stream that is created. In other embodiments, at least in some cases new ingestion, storage or retrieval nodes may have to be instantiated when a stream is created or initialized.

[00117] At the ingestion nodes in the depicted embodiment, records may be received via any of a set of programmatic interfaces implemented for data record submission (element 1710), including for example in-line submission interfaces (in which the data is included in the submission requests) and by-reference submission interfaces (in which an address is provided in the submission requests, from which the data can be retrieved by the SMS ingestion nodes or the SMS storage nodes, e.g., using web service requests or other interfaces). Any of a number of different types of programmatic interfaces may be provided in different embodiments for each of the ways of submitting records, e.g., respective application programming interfaces (APIs) may be supported for in-line versus by-reference submission, web pages or web sites may be established, graphical user interfaces may be implemented, or command-line tools may be developed. In at least some embodiments, the SMS may assign a sequence number to each ingested record, e.g., indicative of the order in which the records are ingested or stored, and the sequence numbers may be usable for retrieval requests by data consumers. At the retrieval subsystem nodes, record retrieval requests may be received via any of a set of implemented programmatic retrieval interfaces, and contents of the requested data records may be provided in response (element 1713). For non-sequential access, the interfaces may include, for example, getIterator (requesting an iterator to be instantiated at a position selected within a partition based on a sequence number indicated in the getIterator invocation) or getRecordWithSequenceNumber (to obtain a data record with a specified sequence number). For sequential access, interfaces such as getNextRecords (requesting a number of records in order, starting from a current position of an iterator or from a specified sequence number) may be

implemented. In at least some embodiments, different retrieval interfaces may have different billing rates associated with them – e.g., the per-record billing rates for sequential retrieval may be set lower than the per-record billing rates for non-sequential retrieval. The different submission interfaces may also have different billing rates in some embodiments – e.g., by-reference submissions may cost more per record than inline submissions.

[00118] Over time, control nodes or specialized billing servers may collect usage metrics for the different programmatic interfaces implemented at the various subsystems of the stream management service (element 1716). The metrics may include, for example, invocation counts of the different programmatic interfaces, the total number of records ingested or retrieved (which may differ from invocation counts for at least some interfaces such as getNextRecords that can be used to retrieve multiple records with a single invocation), the total amount of data ingested or retrieved, and so on. Billing amounts to be charged to the clients that own the stream, or clients that produce and/or consume data from the stream, may optionally be generated based at least in part on the usage metrics and the respective billing rates associated with the programmatic interfaces (element 1719). In at least some embodiments, the billing activities may be asynchronous with respect to the stream ingestion/retrieval operations – e.g., a bill may be generated at the end of a monthly billing period based on the metrics collected during the month.

[00119] FIG. 18a is a flow diagram illustrating aspects of operations that may be performed to configure stream processing (SPS) stages, according to at least some embodiments. As shown in element 1801, programmatic interfaces may be implemented enabling clients to configure a number of processing stages for stream data records. To configure a particular stage, for example, a client may indicate the processing operation(s) to be performed on partitioned stream data records at the stage, the distribution policy for the output of the processing operations, as well as other parameters such as the identity of the input streams from which the data to be processed is to be obtained. In some embodiments, processing operations at SPS stages may be required to be idempotent. In other embodiments, non-idempotent operations may also be supported for at least some stages. If the processing to be performed at a given stage is non-idempotent, a client may still be able to obtain recovery-related benefits of idempotency in some embodiments by configuring worker nodes to periodically flush the output of the operations to some persistent external location, recording when the flush operations were performed with respect to the record retrieval sequence, and later configuring replacement worker nodes to replay the flush operations during recovery. In at least some embodiments, clients may be able to configure directed acyclic graphs (DAGs) or other graphs of processing stages, with several different states operating on stream data in parallel, and results of some stages being used as

input streams for other stages. In some embodiments, one or more ephemeral rather than persistent streams may be created between different stages, e.g., the data records output from one stage need not necessarily be stored on persistent storage devices before being fed as input to a different stage.

[00120] Any of a number of different recovery policies may be implemented for SPS stages in some embodiments, including for example a checkpoint-based recovery policy or a best-effort recovery policy. In one embodiment, a client may use a programmatic interface to select recovery policies for different SPS stages. At stages for which a checkpoint-based recovery is used, worker nodes may be configured to store progress records or checkpoints at intervals, indicating how far along in a stream partition they have reached (for example, the sequence numbers of the most recently processed records may be stored as indicators of the progress). The progress records may be used later during recovery operations after failures, as described below with reference to FIG. 19. In a best-effort recovery policy, progress records need not be stored, and replacement worker nodes configured in response to a failure may simply process new data records as they are received. Within a given SPS stage graph or workflow, in some embodiments different recovery policies may be applied to different stages.

[00121] An SPS control server may receive, e.g., via one of the programmatic interfaces indicated in element 1801, an indication of the idempotent operation Op1 to be performed at a particular stage PS1 of a stream S1 in accordance with a partitioning policy PPol1, with the results of the processing to be distributed in accordance with output distribution descriptor DDesc1 (element 1804). The number of worker nodes to be configured for state PS1, and the virtual or physical resources needed for the nodes, may be determined, e.g., based on various factors such as the Ppol1, the complexity of the idempotent operations Op1, and the performance capabilities of the resources to be used for the worker nodes (element 1807).

[00122] The worker nodes may then be instantiated and configured (element 1810), e.g., as processes or threads at selected virtual or physical machine resources. In one simple implementation, for example, one worker node may initially be assigned for each partition of S1. A given worker node may be configured to (a) receive data records from the appropriate subset of S1's retrieval nodes, (b) perform Op1 on the received data records, (c) optionally, e.g., based on the recovery policy for PS1, store progress records/checkpoints indicating which set of partition records have been processed, and (d) transmit output to destinations indicated by DDesc1 (e.g., as inputs to intermediate persistent or ephemeral streams, or directly to other processing stages or storage systems). It is noted that at least in some embodiments, the SPS processing may not necessarily generate any output that has to be transmitted elsewhere on an

ongoing basis. For example, some SPS applications may simply serve as temporary repositories of data records, and/or may implement query interfaces enabling users to view the data records. Such an application may manage its own output, e.g., output may be generated in response to received queries and not in accordance with a distribution descriptor. A logging-related SPS application may retain the last day's log records collected from a large-scale distributed system, for example, enabling clients to view logging data for debugging or analysis purposes. Accordingly, in some embodiments, output distribution descriptors need not be specified for at least some stages of an SPS, for at least some streams, or for at least some partitions. The worker nodes may then initiate retrieving and processing data records as per their respective configuration settings (element 1813). The SPS control nodes may monitor the health status (e.g., using responsiveness checks such as a heartbeat protocol) of the worker nodes, as well as various other metrics such as the resource utilization levels at the resources being used for the worker nodes (element 1816) in at least some embodiments. The information collected from the worker nodes may be used to determine whether a failover is required, e.g., if a worker node should be replaced and a recovery policy implemented as described below.

[00123] In some embodiments, an installable SPS client library may be provided to those clients that wish to implement SPS worker nodes at client-owned premises, and/or at client-selected resources of the provider network. The client library may also allow SPS clients to select the extent to which they wish to use various control-plane features of an SPS managed service, such as health monitoring functions, automated workload monitoring and balancing, security management, dynamic repartitioning and the like. FIG. 18b is a flow diagram illustrating aspects of operations that may be performed in response to invocations of components of a client library for configuration of stream processing worker nodes, according to at least some embodiments. As shown in element 1851, an SPS client library may be provided (e.g., via download from a web site of a multi-tenant SPS managed service configurable to perform the kinds of operations illustrated in FIG. 18a). The library may include a number of executable components, and/or components that can be linked to client applications. Some library components may enable clients to select, register with the SPS managed service, or specify desired properties of, various worker nodes at which stream processing operations of one or more SPS stages are to be performed. For example, one client may wish to use their own set of compute instances implemented at a virtual computing service of a provider network for the worker nodes, while another client may wish to use computing devices located at the client's own data center (such as special purpose devices not supported by the provider network) for processing stream records. Clients may bring worker nodes online on an as-needed basis at their

own premises, or using compute instances of the virtual computing service, as desired. In addition to or instead of such an on-demand instantiation of worker nodes, in some embodiments clients may preconfigure pools of potentially re-usable worker nodes that can be deployed when needed. In some implementations, a library component may be executed or invoked to allow a client to register, with the SPS managed service, a particular process or thread instantiated by the client as a worker node of a specified stage, for which subsequent control-plane operations may be handled by the SPS managed service. In one embodiment, the client may also be able to select from among different levels of control-plane responsibilities to be handled by the SPS managed service for the worker nodes – for example, one client may wish to use their own custom modules to monitor worker node health, while another client may wish to utilize the SPS managed service for monitoring worker node health and taking the appropriate actions if a failure is detected.

[00124] The SPS managed service may receive an indication that a particular client wishes to use the client library for configuring worker nodes and/or control plane operations of a particular SPS stage PS1 (element 1854). (PS1 itself may be designed using programmatic interfaces included in the library, or using programmatic interfaces exposed by the SPS managed service similar to the web-based interface illustrated in FIG. 4.) The client may also indicate the streams whose data is to be retrieved for use as input by PS1. Optionally, in at least some embodiments, the client may indicate control-plane settings for PS1, e.g., whether the client wants to use the service's health monitoring capabilities for the nodes, or is willing to use custom health monitoring tools (element 1857). Depending on the preferences indicated by the client, one or more nodes of the SMS and/or SPS to be configured for the client's use may be determined (element 1860). Network connectivity may be established between the client's worker nodes to the SMS/SPS nodes, and/or other configuration operations may be performed to enable the flow of data records and processing results as desired. Data records may be provided to SP1 worker nodes upon receiving retrieval requests, and desired control-plane operations (if any were requested by the client) may be performed as needed. It is noted that at least in some embodiments, a similar approach enabling clients to control the extent to which they wish to use the control-plane functionality of various subsystems of an SMS managed service may also or instead be implemented.

[00125] FIG. 19 is a flow diagram illustrating aspects of operations that may be performed to implement one or more recovery policies for stream processing, according to at least some embodiments. As shown in element 1901, an SPS control node may determine that triggering criteria for replacing a particular worker node have been met – e.g., the worker node may have

become unresponsive or unhealthy, the workload levels of the current node may have reached a threshold value for failover, the number of errors detected at the worker node may have exceeded a threshold, or some other unexpected state of a worker node may be identified. A replacement worker node may be identified or instantiated (element 1904). In some embodiments, a pool of available worker threads may be set up, from which one may be selected as a replacement, for example, or a new thread or process may be launched.

[00126] If a best-effort recovery policy is to be used at the SPS stage at which the particular worker node was active (as determined in element 1907), the replacement worker node may simply start processing additional data records as they become available (element 1916), e.g., no record of the replaced worker node's progress need be examined. If a checkpoint-based recovery policy is to be used, an indication of the location (e.g., a storage device address or a URL) at which the replacement worker node may access the progress records stored by the replaced worker node may be provided (element 1910). The replacement worker node may retrieve the most recent progress record stored by the replaced node, and use the progress record to determine the set of data records on which the replacement worker node should perform the idempotent operations of the stage (element 1913). In such a checkpoint-based recovery policy, depending on the duration between the last progress record and the time at which the replacement worker node is instantiated, as well as on the rate at which the replaced worker node had processed additional records subsequent to the progress record being stored, some number of data records may be processed more than once. If the operations being performed are idempotent, such repeat operations may have no negative effects in at least some embodiments. After the replacement worker node has performed the repeat recovery operations based on the earlier-stored progress record, in at least some embodiments the replacement worker thread may store its own progress record indicating that recovery is complete, and may start normal worker thread operations on newly-received data records (element 1916).

[00127] FIG. 20 is a flow diagram illustrating aspects of operations that may be performed to implement a plurality of security options for data streams, according to at least some embodiments. As shown in element 2001, one or more programmatic interfaces may be implemented that enable clients to select from a variety of security options for data stream management and processing, including for example placement destination type options for the nodes of different functional categories (e.g., ingestion, storage, retrieval, processing or control nodes). The placement destination types may differ from one another in various aspects of their security profiles. The physical location of the resources to be used for the SMS or SPS nodes may differ from one destination type to another in some embodiments. For example, resources

such as instance hosts located at provider network data centers may be used for the nodes, or resources at client-owned facilities may be used, or third-party resources may be used. The network isolation levels or other networking characteristics may differ from one destination type to another in at least some embodiments – e.g., some SMS or SPS nodes may be instantiated within isolated virtual networks, or at client-owned facilities connected to the provider network via dedicated isolated physical links. In one embodiment, clients may indicate that certain types of SMS or SPS nodes are to be established at single-tenant instance hosts of a provider network, instead of using multi-tenant instance hosts that may also be available. In at least some embodiments, various types of encryption options may also be selectable via the security-related programmatic interfaces.

[00128] A client's security profile choices or preferences regarding nodes of one or more functional categories for a stream S1 may be received via the security-related programmatic interfaces. For example, the client may select one security profile for nodes of functional category FC1 (e.g., the client may wish to implement SPS worker nodes at client-owned premises) and a different security profile for nodes of a different functional category FC2 (e.g., the client may be willing to implement SMS ingestion nodes or storage nodes at provider network data centers) (element 2004). In some cases, a client may decide to set up nodes of all the different functional categories with the same security profile. The SMS and/or the SPS may define default placement destination types for the various functional categories in some embodiments – e.g., unless a client indicates otherwise, nodes of all the functional categories may be set up within isolated virtual networks of a provider network.

[00129] The nodes of the different functional categories may then be configured based on the client's preferences for security profiles and/or locations (or based on default settings for the functional categories for which the client does not provide preferences) (element 2007). The configuration may involve, for example, selecting the appropriate physical hosts or machines, and instantiating the appropriate compute instances, virtual machines, processes and/or threads for the nodes of the different functional categories, and establishing the appropriate network connections between the nodes. In some embodiments, executable library components for the different stream management and processing functions may be provided for installation at hosts external to the provider network as part of the configuration.

[00130] According to at least some embodiments, encryption modules may be activated at one or more categories of the nodes, e.g., in accordance with the client's expressed encryption preferences or based on default encryption settings (element 2010). The nodes of the various

functional categories may then be activated, so that the stream data is ingested, stored, retrieved and/or processed as desired by the client (element 2013).

[00131] FIG. 21 is a flow diagram illustrating aspects of operations that may be performed to implement a partitioning policy for data streams, according to at least some embodiments. As shown in element 2101, a partitioning policy may be determined for a data stream. The policy may comprise, for example, an initial mapping of data records to partitions based on keys supplied by data producers or based on various attributes of the submitted data records, as well as one or more triggering criteria for repartitioning the data stream. In some embodiments, for example, a hash function may be applied to the partition key or keys, yielding a 128-bit integer hash value. The range of possible 128-bit integers may be divided into N contiguous sub-ranges, each representing one of N partitions of the stream. The number of partitions and/or the relative sizes of the sub-ranges may vary from one stream to another in some embodiments. In at least some embodiments, the client on whose behalf a stream is being configured may provide input regarding the partitioning scheme to be used, e.g., the number of partitions desired, or desired characteristics of the partitioning function to be used. In at least one embodiment, clients may provide the partition identifiers or names for some subset or all of the submitted data records.

[00132] As the data records of the stream are received, their respective partitions may be determined based on the supplied keys and/or other attributes, and the appropriate set of ingestion, storage and retrieval nodes may be selected for the identified partition (element 2104). In at least some embodiments, respective sequence numbers may be generated for the data records, e.g., indicative of the sequence in which the records of a given partition were received (element 2107). The sequence numbers may comprise a number of elements in some implementations, such as timestamp values (e.g., the number of seconds elapsed since a well-known epoch such as 00:00:00 UTC January 1 1970), subsequence values obtained from a storage subsystem, version numbers of the SMS software, and/or the partition identifiers. The sequence numbers may be provided to the data producers in some embodiments, e.g., to acknowledge the successful ingestion of the submitted data records. The sequence numbers may also be used by data consumers to retrieve the data records of a stream or a partition in ingestion order in some embodiments.

[00133] The data records may be stored in sequence number order in at least some embodiments at the storage nodes to which they are directed based on the partitioning policy (element 2110). In embodiments in which rotating magnetic disks storage devices are used, sequential writes may typically be used to save the received data records to disk, thereby avoiding disk seek latencies. In at least some implementations, non-volatile buffers may be used

as write caches prior to storing the records to disk, e.g., to further decrease the probability of disk seeks. In response to requests for reads of multiple data records ordered by sequence number (e.g., invocations of getNextRecords or similar interfaces), the data records may later be read using sequential reads from the storage devices (element 2113).

[00134] FIG. 22 is a flow diagram illustrating aspects of operations that may be performed to implement dynamic repartitioning of data streams, according to at least some embodiments. As shown in element 2201, a determination may be made (e.g., at a control component of an SMS or an SPS) that a stream is to be dynamically repartitioned. A number of different triggering conditions may lead to a decision to repartition a stream, such as a detection of overload at one or more of the ingestion, storage, retrieval, processing or control nodes, or a detection of an imbalance in the workload levels of different nodes, or a repartitioning request that may be received from a client (e.g., a data producer or data consumer). Client repartitioning requests may include specific details of the requested repartitioning in some implementations, such as various parameters of the modified mapping to be generated (e.g., the number of partitions to be added or removed, which specific partitions should be combined or split, and so on). In one implementation, a client repartitioning request may indicate a problem state (such as a load imbalance) that the client wishes to resolve, and the SMS or SPS may be responsible for translating the description of the problem state into the appropriate repartitioning operation. In some cases, instead of requesting a repartitioning or describing a problem state, a client may specify the triggering criteria to be used for repartitioning. A determination of a change to a data durability requirement of the data stream may trigger repartitioning in some embodiments, which may for example result in the selection of a different set of storage devices or a different storage technology for the stream records. A detection of a change to a usage pattern of the data stream (e.g., the rate at which data records are being produced or consumed) may also lead to repartitioning in some cases, and may also lead to the use of a different storage technique or a different set of storage devices that is more appropriate for the changed usage pattern. For example, a decision to repartition may be based on the determination that, for the rate of reads and writes expected for a given partition or an entire stream, SSDs may be a more appropriate storage technology than rotating magnetic disks. Scheduled or impending software and/or hardware version changes may trigger repartitioning in one embodiment. In some cases, pricing or billing concerns may trigger repartitioning, as when a client indicates a budget constraint that can be met more effectively using a different partitioning approach or a different approach to storage. Changed performance targets may also trigger repartitioning in at least some embodiments. In the embodiment depicted in FIG. 22, an initial timestamp value (such as an

offset in seconds from 00:00:00 UTC January 1 1970, an epoch value typically available via a system call in several operating systems) to be used for sequence numbers assigned after the repartitioning may be selected (element 2204). In some implementations, a global state manager implemented at a provider network may support a getEpochValue API, for example, enabling various components of the SMS and/or SPS to obtain consistent timestamp values to be used for sequence number generation. In other implementations, other time sources may be used – e.g., an SMS or SPS control node may be designated to provide consistently-ordered timestamp values to other components, or a local system call invocation may be used. In some embodiments, the timestamp values need not necessarily correspond to the wall clock time at any particular host – e.g., a monotonically increasing integer counter value may simply be used.

[00135] A modified partition mapping, different from the mapping in use at the time of the repartitioning decision, may be generated for the stream (element 2207). The changed mapping may map data records with a particular partition key to a different partition than data records with the same key were mapped to before the repartitioning in at least some embodiments. Some partitions (typically, heavily-used partitions) may be split, while other (typically lightly-used) partitions may be merged, depending on the triggering conditions for the repartitioning and/or on observed workload metrics. A different partitioning function may be used after the repartitioning than before the repartitioning in some embodiments - e.g., a different hash function, or a different approach to the subdivision of hash function results into partitions may be used. In some implementations, for example, in which the partitions correspond to contiguous ranges of 128-bit integers, the 128-bit integer space may be divided into a different set of sub-ranges after the repartitioning. In at least some embodiments, new sets of ingestion, storage, retrieval, processing or control nodes may be assigned to the newly-created partitions. In some implementations a space-efficient combined data structure may be used to represent both the initial mapping and the modified mapping (element 2208). For example, a directed acyclic graph or tree structure may be stored, in which each entry contains an indication of a partitioning function output range (e.g., the range of a partitioning hash function's results that correspond to a given partition) and a validity time range, so that only the records corresponding to modified partitions need to be altered as a result of a repartitioning. Entries for partitions that remain unaltered during a repartitioning may not need to be modified in the data structure. The new nodes may be configured to implement the modified partition mapping (element 2210). In at least some embodiments, since retrieval requests for data records stored on the basis of the earlier mapping may continue to be received for at least some time, the previous nodes and the previous mapping may be retained for some time. When a read request specifying a particular

sequence number or timestamp is received (element 2213), a determination may be made (e.g., at a control node or at a retrieval node) as to whether the read request is to be satisfied using the new partition mapping or the previous partition mapping. The selected mapping may then be used to identify the appropriate storage node from which the requested data is to be obtained.

[00136] FIG. 23 is a flow diagram illustrating aspects of operations that may be performed to implement an at-least-once record ingestion policy for data stream records, according to at least some embodiments. As shown in element 2301, one or more programmatic interfaces may be implemented to enable clients to select a record ingestion policy for a data stream from among several ingestion policy options, including for example (a) an at-least-once policy in accordance with which a record submitter is to submit a record one or more times until a positive acknowledgement is received or (b) a best-effort ingestion policy in accordance with which acknowledgements are not provided for at least some record submissions. Some data producing clients may not be as concerned about the potential loss of a small fraction of their records as others, and may therefore opt for the best-effort ingestion approach. In some implementations, even for streams configured for best-effort ingestion, the SMS may still provide acknowledgements for some subset of the data records, or may even attempt to provide acknowledgements for all the data records, even though the best-effort policy does not require acknowledgements for every data record.

[00137] A request may be received via one of the programmatic interfaces, indicating a particular ingestion policy to be used for a specified stream (element 2304). Ingestion nodes may be instantiated in accordance with the partitioning policy in effect for the stream (element 2307). When one or more submissions of the same data record are received at an ingestion node (element 2310), different actions may be taken dependent on the ingestion policy in effect. If the at-least-once ingestion policy is in use (as determined in element 2313), an acknowledgement may be sent to the data producer for each of the one or more submissions, but the data record may be saved only once at the storage subsystem (2316). (It is noted that in accordance with the persistence policies in effect for the stream, N replicas of a given record may be stored in some cases, but if a given data record is submitted M times, the replicas may be generated only for one of the submissions – i.e., the total number of record replicas stored would still be N, and not NxM.) If a best-effort ingestion policy were in effect (as also detected in element 2313), the data record may still be saved once at a storage device, but no acknowledgement need be sent to the data producer (element 2319). In at least some embodiments, client billing amounts may optionally be determined based at least in part on the ingestion policy selected (element 2322). As noted earlier, in some embodiments, two versions of an at-least-once ingestion policy may be

supported. In one version, similar to that illustrated in FIG. 23, the SMS may be responsible for de-duplicating data records (i.e., ensuring that data is stored at the SMS storage subsystem in response to only one of a set of two or more submissions). In a different version of at-least-once ingestion, duplication of data records by the SMS may be permitted. The latter approach may be useful for stream applications in which there are few or no negative consequences of data record duplication, and/or for stream applications that perform their own duplicate elimination.

[00138] FIG. 24 is a flow diagram illustrating aspects of operations that may be performed to implement a plurality of persistence policies for data streams, according to at least some embodiments. As shown in element 2401, one or more programmatic interfaces enabling clients to select a persistence policy for stream data records from among a plurality of persistence policies may be implemented. The persistence policies may differ from each other in any of various respects: e.g., (a) the number of replicas to be saved may differ (e.g., N-replica vs. 2-replica vs. single-replica policies may be supported) (b) storage location/device types to be used may differ (e.g., rotating magnetic disk vs. SSD vs. RAM vs. a database service or a multi-tenant storage service) and/or (c) the policies may differ in the expected extent of resilience to large-scale failures (e.g., multi-data-center vs. single-data-center policies may be supported). A request may be received indicating a client's selection of a particular persistence policy for a specified stream (element 2404). In some embodiments, the persistence policy selected by a client may result in the use of different storage location types or device types for respective partitions of a given stream. In one embodiment, the SMS rather than the client may select the storage location type or device types, either at the stream level or at the partition level. Clients may indicate data durability goals and/or performance goals (such as desired read or write throughput or latency) in some embodiments when selecting the persistence policy in some embodiments, and these goals may be used by the SMS to select the appropriate storage devices types or locations. For example, if low latencies are desired, SSDs may be used instead of rotating magnetic disks to store the data records of one or more partitions or streams.

[00139] A set of ingestion nodes may be determined or configured to receive the data records of the selected stream from data producers, and a set of storage nodes may be configured to implement the selected persistence policy (element 2407). When a data record is received at an ingestion node (element 2410), one or more copies of the data record may be stored, based on the selected persistence policy, at selected storage devices by the storage nodes responsible for the partition to which the data record belongs (element 2413). In at least some implementations, billing amounts may optionally (and/or asynchronously) be determined based on the specific persistence policies selected by the client (element 2416).

Decentralized workload management for stream processing

[00140] In some embodiments, a substantial portion or all of the control-plane functionality of an SPS may be implemented in a decentralized manner, e.g., by the worker nodes within a given SPS stage coordinating various control operations (such as partition assignment to the worker nodes, responses to dynamic repartitioning, health monitoring and/or load balancing) via a shared data structure such as a database table. A given worker node W1 may inspect entries within the shared data structure to determine, for example, which partitions of the stage's input streams (if any) are currently not being processed. If such a partition P1 is found, W1 may update an entry in the shared data structure to indicate that W1 will perform the stage's processing operations on P1's records. Other worker nodes may learn that W1 is assigned to process P1 records, and may therefore assign different partitions to themselves. Worker nodes may periodically or occasionally submit queries to the SMS control plane to determine the current partition maps in effect for the input stream, and update the shared data structure to indicate map changes (e.g., as a result of repartitioning) as necessary. Load balancing and other operations may also be coordinated via the shared data structure in various embodiments, as described below. In some such decentralized implementations, dedicated control nodes may not be required for the SPS, thereby reducing the overhead required to implement SPS workflows. Such decentralized SPS control plane implementations may be especially popular with budget-conscious customers that utilize SPS client libraries to implement various aspects of stream processing, e.g., at compute instances within the provider network that are assigned to the customers, or at locations outside the provider network. Decentralized SPS control-plane techniques may also be used in embodiments in which client libraries are not used, e.g., when all the resources used for the SMS and SPS are configured within a provider network. An SPS at which the worker nodes implement some or all of the SPS control plane functions for at least some processing stages may be referred to herein as a "decentralized-control SPS".

[00141] FIG. 25 illustrates an example of a stream processing system in which worker nodes of a processing stage coordinate their workloads using a database table, according to at least some embodiments. Within a decentralized-control SPS 2590, two stages 215A and 215B are defined, each with a respective set of worker nodes. Stage 215A comprises worker nodes 2540A and 2540B, while stage 415B comprises worker nodes 2540K and 2540L. For each stage 215A and 215B, a corresponding partition assignment (PA) table 2550 is created at a database service 2520, such as PA table 2550A for stage 215A and PA table 2550B for stage 215B. The PA table 2550 for a given stage may be created during stage initialization in some embodiments, e.g., in response to an invocation of a client library component or function. Each PA table 2550 may be

populated with an initial set of entries or rows representing unassigned partitions of the input streams of the stage (i.e., partitions to which no worker node is currently assigned). Example columns or attributes of the PA table entries are shown in FIG. 26 and described below. The worker nodes 2540 (e.g., processes or threads launched at compute instances or other servers) that are launched for the stage may be granted read/write access to the stage's PA table. Reads and writes directed to the PA tables from the worker nodes are represented in FIG. 25 by arrows 2564A, 2564B, 2564K and 2564L for worker nodes 2540A, 2540B, 2540K and 2540L respectively.

[00142] A given worker node 2540 may be configured to select, by examining the entries in the PA table, a particular partition on which to perform the processing operations of the stage. In one implementation, the worker node 2540A may scan the entries in the PA table 2550A until it finds an entry of an unassigned partition Pk, and may attempt to assign the partition Pk to itself by updating the entry, e.g., by inserting the worker node's identifier into one of the columns of the entry. Such an insertion may be considered analogous to locking the partition by the worker node. Depending on the type of database service being used, different approaches to managing potentially concurrent writes to PA table entries (e.g., by two or more worker nodes that happen to identify an unassigned partition at close to the same time) may be used.

[00143] In one embodiment, a non-relational multi-tenant database service of a provider network may be used, which supports strong consistency and conditional write operations without necessarily supporting relational database transaction semantics. A conditional write operation may be used in such a case for the updates by the worker nodes. Consider an example in which a column "worker-node-ID" is used to indicate the identifier of the particular worker node assigned to a partition in the PA table, and that the column's value is set to "null" if no worker node is assigned to the partition. In such a scenario, a worker node with identifier WID1 may request the logical equivalent of the following: "if, in the entry for partition Pk, worker-node-ID is null, then set worker-node-ID for that entry to WID1". If such a conditional write request succeeds, the worker node with identifier WID1 may assume that partition Pk is assigned to it. The worker node may then start retrieving data records of partition Pk, e.g., using record retrieval interfaces of SMS retrieval subsystem 206, as indicated by arrows 2554 (e.g., arrows 2554A, 2554B, 2554K and 2554L for worker nodes 2540A, 2540B, 2540K and 2540L respectively), and performing the processing operations on the retrieved records. If the conditional write fails, the worker node may resume a search for a different unassigned partition. In other embodiments, database services (such as relational databases) that support transactions may be used, and the transaction functionality may be used to implement the equivalent of the

conditional write operations – e.g., to ensure that only one of a plurality of concurrent (or near-concurrent) attempts to assign a partition to a worker node succeeds, and that the worker nodes involved in such concurrent attempts are reliably informed of their success or failure. Synchronization techniques that rely neither on conditional writes nor on transaction support may be used in some embodiments. In some implementations a database service may not be used; instead, a locking service may be used by the worker nodes to acquire exclusive access for updates to the entries in persistent data structures analogous to the PA tables.

[00144] Other worker nodes 2540 may examine the entries in the PA table, determine which partitions are unassigned, and may eventually succeed in assigning one or more partitions to themselves. In this way, the processing workload for the partitions of the stage's input stream or streams may eventually be distributed among themselves by the stage's worker nodes.

[00145] The initial partition mapping of any given stream may change over time, e.g., as a result of the dynamic repartitioning operations described earlier. Accordingly, in the embodiment depicted in FIG. 25, one or more of the worker nodes 2540 may occasionally (or in response to triggering conditions as described below) submit requests to the SMS control subsystem 210 of their stage's input stream(s) to obtain the current partition metadata. In some implementations, such requests may comprise invocations of SMS control plane APIs, such as the invocations of a getStreamInfo API indicated by arrows 2544A, 2544B, 2544K, and 2544L. The SMS control subsystem may, for example, respond with an up-to-date list of partitions of the stream, and/or other details such as the validity time periods of the partitions. If the partition information provided by the SMS control subsystem 210 does not match the entries in the PA table, the PA table may be modified by the worker node, e.g., by inserting or deleting entries for one or more partitions. Such requests 2554 to the SMS control subsystem may typically be much less frequent than the record retrieval requests 2554 (and/or the database read or write operations 2564) in at least some embodiments, as indicated by the label "infrequent" of arrow 2554A. For example, once it is assigned a partition, a worker node may typically keep retrieving and processing that partition's data records until the partition data is fully consumed (e.g. if the owner of the stream closes the stream, or if the partition is closed as a result of dynamic repartitioning), or until some other low-probability circumstance is encountered (e.g., if a different worker node requests a transfer of the partition due to detected load imbalance, as discussed below). Thus, the overhead associated with invoking the getStreamInfo or similar APIs may typically be quite small in various embodiments, even if a substantial amount of information is provided in response to any given invocation (as might be the case if hundreds or thousands of partitions are defined for a stage's input stream).

[00146] Some of the key workload-management operations of a decentralized-control SPS environment may thus be summarized as follows in the embodiment depicted in FIG. 25: (a) selecting, based at least in part on accessing a database table by a first worker node of a stream processing stage, a particular partition of an input data stream of the stream processing stage on which to implement a set of processing operations defined for that stage; (b) writing, into a particular entry stored in the table, an indicator of an assignment of the particular partition to the first worker node; (c) retrieving, by the first worker node, records of the particular partition using programmatic record retrieval interfaces implemented at a multi-tenant stream management service; (d) implementing, by the first worker node, the set of processing operations on the records of the particular partition; (e) determining, by a second worker node, based at least in part on the particular entry in the particular database table, that the first worker node is assigned to perform the set of processing operations on the particular partition; and (f) selecting, by the second worker node a different partition on which to perform the set of processing operations. If and when a worker node determines that no more records remain in a partition assigned to it, the worker node may request metadata on the input stream from the SMS control subsystem, and may update the PA table if the metadata indicates a discrepancy.

[00147] FIG. 26 illustrates example entries that may be stored in a partition assignment table 2550 used for workload coordination, according to at least some embodiments. As shown, table 2550 may comprise four columns: partition identifier column 2614, assigned worker node identifier column 2618, a worker node health indicator column 2620, and a workload level indicator column 2622. Other column sets may be implemented in other implementations – for example, a column that indicates a partition creation time or a partitioning function output value range may be used in some embodiments, or the workload level indicator column may not be used.

[00148] It is noted that the partition list 2650 maintained by the SMS control subsystem (e.g., as part of the partition entry tree, graph or other combined data structure described earlier) may, at least at some points in time, include more partitions than are included in the PA table 2550 in some embodiments. In the depicted example, the partition list 2650 includes partitions P1, P2, P3, P4 and P5, of which P1 and P4 are shown in a closed state as a result of repartitioning, while P2, P3 and P5 are shown as active (i.e., partitions whose data records are currently being retrieved and processed). The PA table 2650 includes entries for the active partitions in the depicted embodiment, and does not include entries for the closed partitions (which may have been deleted by worker nodes when they obtained responses to getStreamInfo invocations after the repartitioning took place, for example). At least in some implementations, not all the

currently open partitions of the stream may necessarily have respective entries in the PA table at a given point in time; instead, for example, only a subset of those partitions that are currently assigned or being processed may be represented.

[00149] In the example scenario illustrated in FIG. 26, partitions P1 and P2 are assigned to worker nodes with identifiers W7 and W3 respectively, while P5 is currently unassigned. The health indicator column 2620 may store different types of values in different implementations. In some implementations, the worker nodes may be responsible for periodically (e.g., once every N seconds, or according to a schedule based on some set of heuristics) updating the contents of the health indicator columns in the PA entries of their assigned partitions to indicate that the worker nodes are active and able to continue their retrieval and processing operations. In FIG. 26, an indication of the most recent time that the worker node for that entry updated the health indicator column ("last-modified-time") may be stored – e.g., worker W7 is shown as having modified the entry at 02:24:54 and 53 seconds on December 1 2013. Other worker nodes may use the last-modified time value to determine whether the assigned worker node is healthy or not in some embodiments – e.g., if X seconds or minutes have elapsed, as defined in a failover policy for the stage, the assigned worker node may be assumed to be unhealthy or inaccessible and the partition may be reassigned. In other implementations, a counter may be used as a health indicator (e.g., if the counter value has not changed in Y seconds, the assigned worker node may be deemed a candidate for failover), or a "last-read-time" value indicating when the assigned worker node last read the entry may be used.

[00150] In at least some embodiments, a workload level indicator value 2622 may be stored in the entry, e.g., by the assigned worker node, such as the number of records processed during some recent time interval (e.g., in the five minutes prior to the last-modified-time), recent performance-related metrics of the worker node such as CPU utilization, memory utilization, storage utilization and the like. Such workload level indicator values may be used in some embodiments by the worker nodes to determine whether load imbalances exist, as described below with respect to FIG. 29, and to take actions in response to detected imbalances. For example, a worker node Wk may determine that its workload level is above the average workload level, and may un-assign one of its partitions, or may request a dynamic repartitioning; alternatively, the worker node Wk may determine that its workload is too low relative to that of other worker nodes or partitions, and may assign additional partitions to itself. Thus, using the columns of the PA table indicated in FIG. 26, worker nodes may perform some of the same types of control-plane functions in the depicted embodiment that may typically be performed by dedicated SPS control nodes in centralized-control SPS implementations

[00151] FIG. 27 illustrates aspects of operations that may be performed by worker nodes of a stream processing stage to select partitions on which to perform processing operations, according to at least some embodiments. As shown in element 2701, a PA table PAT1 may be initialized at a database service for a decentralized-control SPS processing stage SP1. The table may be created, for example, when an SPS client library component is invoked, e.g., from a host at a client facility or from a compute instance at a provider network data center. The client library may be used for various purposes: for example, to provide an executable component such as a JAR (Java™ archive) file for the particular processing operations to be implemented at the SPS stage, to indicate a label (such as a program name, a process name or a compute instance name) that can be used to identify the worker nodes, to indicate the stream to be used as the input for the stage, to indicate the output destinations (if any) of the stage, and so on. PAT1 may initially be populated in some embodiments with entries or rows for at least a subset of the partitions {P1, P2, ...} defined for the input stream(s) of the stage. In some implementations, the table may be left empty initially, and one or more of the worker nodes may populate the table with rows for unassigned partitions, e.g., as a result of obtaining partition metadata from an SMS control subsystem. A initial set of worker nodes {W1, W2, ...} may be started up, e.g., at various compute instances within a provider network or at client-owned computing devices (element 2704). The worker nodes may be granted read and write access to PAT1 in the depicted embodiment.

[00152] As the worker nodes come online, they may each access PAT1 to try to find partitions that are unassigned. For example, worker node W1 may examine PAT1 and find that partition P1 is unassigned (element 2707). W1 may then update P1's entry in PAT1, e.g., using a conditional write request or a transactional update request depending on the type of database service being used, to indicate that P1 is assigned to W1 (element 2710). Having updated the table, W1 may initiate retrieval of data records of P1 using SMS retrieval subsystem interfaces (element 2713), and may perform the processing operations of the stage PS1 on the retrieved records.

[00153] Meanwhile, at some point in time, a different worker node W2 may access PAT1 in its own attempt to find unassigned partitions (element 2716). W2 may determine, based on W1's earlier update, that P1 is already assigned, but that a different partition P2 is not assigned. In some embodiments, a determination by W2 that the current assignee worker node of P2 is unhealthy or inactive (e.g., based on the health indicator column in P2's entry) may also lead W2 to select P2. Thus, in at least some embodiments, either an unassigned state, or a determination of an unhealthy state of a current worker node may be used to select a given partition for reassignment (or initial assignment). W2 may then attempt to update PAT1 to assign P2 to itself

(element 2719). If the update succeeds, W2 may start retrieving P2 records using SMS retrieval interfaces (element 2722) and performing the appropriate processing operations defined for the stage.

[00154] As mentioned earlier, the worker nodes in a decentralized-control SPS may (typically infrequently) obtain partition mapping information from the SMS, and use such information to update the PA table if necessary. FIG. 28 illustrates aspects of operations that may be performed by worker nodes of a stream processing stage to update a partition assignment table based on information obtained from a stream management service control subsystem, according to at least some embodiments. As shown in element 2801, during worker node initialization or in response to various triggering conditions such as the closing of one of the partitions assigned to it, a worker node W1 may submit a request to the SMS control subsystem to obtain the latest or current partition list, or the active partition list. In some implementations, a getStreamInfo or similar API may be invoked for this purpose. Other triggering conditions may be used in some embodiments: e.g., the worker nodes may each be configured to obtain fresh partition lists after random amounts of time, or in response to unexpected drops or increases in workload levels. The partition list returned by the SMS may be compared with the entries in the PA table for the partition (element 2807). If a discrepancy is found (e.g., if there is some partition in the freshly-obtained partition list that is not in the PA table, or if there is an entry in the PA table that is not in the SMS's list), the worker node may insert or delete entries in the PA table to resolve the discrepancy in the depicted embodiment (element 2810). (Additional coordination may be required if an entry that is targeted for deletion currently has an assigned worker node in some implementations – e.g., the assigned worker node may be notified, either directly or via the PA table itself.)

[00155] After the discrepancy is rectified, or if no discrepancy was detected, the worker node W1 may select a set of partitions on which it should perform the stage's processing operations (element 2813), and may update the PA table accordingly. In some cases, depending on the triggering condition that led to the partition list being retrieved, W1 may already have one or more partitions assigned to it, and may not need to make changes to its assignments or update the PA table. W1 may then proceed to retrieve the data records of its assigned partition or partitions, and process the records, without having to interact with the SMS control subsystem or changing the number of entries in the PA table (element 2816). Eventually, when a triggering condition is detected (e.g., when the equivalent of an "end of partition reached" response is received to a retrieval request, indicating that the a partition is closed), W1 may again send a request to the

SMS control subsystem for fresh partition information, and the operations of elements 2801 onwards may be repeated.

[00156] FIG. 29 illustrates aspects of load balancing operations that may be performed by worker nodes of a stream processing stage, according to at least some embodiments. As shown in element 2901, a worker node W1 may determine that a load balancing analysis is to be performed on its stage upon a detection of any of a variety of triggering conditions, such as a detection of a high resource utilization level, or based on a configurable schedule. W1 may examine the entries in the PA table (element 2904) to determine various workload metrics for the stage. Such metrics may include the average number of partitions assigned to worker nodes, the average workload level of the worker nodes or of different partitions (in embodiments in which workload level indicators are saved in the table), a range or distribution of the per-worker-node workload, and so on.

[00157] W1 may then compare its own workload (based for example on the number of partitions assigned to W1, and/or the per-partition workload level indicators) to some or all of the metrics. In general, any of three types of conclusions may be drawn: that W1 is overloaded, that W1 is under-loaded, or that W1's workload is neither too high nor too low. Workload levels that are "too high" or "too low" may be defined by policies selected by the clients on whose behalf the stage is configured in some embodiments, or using some default set of heuristics in other embodiments. If W1 determines that its workload is too low (element 2907), e.g., below some minimum-load threshold T1, a busier or more highly-loaded worker node Wk may be identified (element 2910). W1 may then initiate a process of transferring one or more partitions Pm from Wk to itself (element 2913), e.g., by attempting to modify the Pm entry in the PA table, requesting such a modification (which may result in a notification being generated for Wk), or by requesting Wk directly.

[00158] If W1 determines that its workload is too high (element 2916), e.g., above a maximum-threshold T2, it may identify one or more of its assigned partitions Pn to relinquish (i.e., to release for assignment by other worker nodes) (element 2919). W1 may then modify the appropriate entries in the PA table, e.g., by removing its identifier from the assignee column of the entry for Pn (element 2922). If W1's workload was neither too high nor too low, or after W1 has taken the kinds of actions described above to increase or decrease its workload, W1 may resume processing records of the partitions to which it is assigned (element 2925). Operations corresponding to elements 2901 onwards may be repeated when and if conditions triggering another load balancing analysis are met. It is noted that in the operations illustrated in FIG. 29, W1 is shown as initiating workload changes only when it detects an imbalance with respect to its

own workload. In other embodiments, W1 may initiate rebalancing actions if it detects imbalances among other worker nodes than itself – e.g., if it determines that W2 has a much lower workload level than W3. In some implementations, W1 may request or initiate dynamic repartitioning (e.g., by invoking a repartitionStream SMS API such as that shown in FIG. 3, or its equivalent) if and when it detects workload imbalances. In some embodiments, the kinds of operations illustrated in FIG. 29 may be performed by a newly-configured worker node – e.g., when new nodes are added to a stage after the stage has already been in operation for some time, the new nodes may indirectly notify the existing nodes of their presence by requesting reassignment of partitions from heavily loaded existing nodes. In some embodiments, decentralized control techniques similar to those described above for SPS worker nodes may also or instead be used at one or more SMS subsystems, e.g., the nodes of the ingestion, storage or retrieval subsystems may coordinate their workloads using shared data structures similar to the PA tables.

[00159] It is noted that in various embodiments, operations other than those illustrated in the flow diagrams of FIG. 17 – FIG. 24 and FIG. 27 - 29 may be used to implement the stream management service and/or the stream processing functionality described above. Some of the operations shown may not be implemented in some embodiments or may be implemented in a different order, or in parallel rather than sequentially. It is also noted that, with respect to each of the SMS and SPS functions for which programmatic interfaces are supported in various embodiments, any combination of one or more techniques may be used for implementing the interfaces, including the use of web pages, web sites, web-services APIs, other APIs, command-line tools, graphical user interfaces, mobile applications (apps), tablet apps, and the like.

Use cases

[00160] The techniques described above, of establishing scalable partitioning-based, dynamically configurable managed multi-tenant services for collection, storage, retrieval and staged processing of stream data records may be useful in a number of scenarios. For example, large provider networks may comprise thousands of instance hosts implementing service instances of a number of different multi-tenant or single-tenant services for tens of thousands of clients simultaneously. Monitoring and/or billing agents installed on the various instances and hosts may rapidly generate thousands of metric records, which may need to be stored and analyzed to produce accurate billing records, to determine effective provisioning plans for the data centers of the provider network, to detect network attacks, and the like. The monitoring records may form an input stream to an SMS for scalable ingestion and storage, and SPS techniques described may be implemented for the analysis of the collected metrics. Similarly,

applications to collect and analyze large numbers of log records from numerous log sources (e.g., application logs from the nodes of a distributed application, or system logs from the hosts or compute instances at a data center) may also be able to utilize SMS and SPS functionality. In at least some environments, the SPS processing operations may comprise a real-time ETL (Extract-Transform-Load) processing operation (i.e., an operation that transforms received data records in real time for loading into a destination, instead of doing the transformation offline), or a transformation of data records for insertion into a data warehouse. Using an SMS/SPS combination for loading data into a data warehouse in real time may avoid the delays that are typically required to clean and curate data from one or more data sources, before the data can be inserted into a warehouse for analysis.

[00161] A number of different "big data" applications may also be built using the SMS and SPS techniques. For example, the analysis of trends in various forms of social media interactions may be performed efficiently using streams. Data collected from mobile phones or tablet computers, such as location information of the users, may be managed as stream records. Audio or video information, collected for example from a fleet of monitoring cameras may represent another category of streaming data set that could be collected and processed in a scalable manner, potentially helping prevent attacks of various kinds. Scientific applications that require analysis of ever-growing data sets, collected for example from weather satellites, ocean-based sensors, forest-based sensors, astronomical telescopes, may also benefit from the stream management and processing capabilities described herein. The flexible policy-based configuration options and pricing options may help different types of users customize the streaming functionality to suit their specific budgets and data durability/availability requirements.

[00162] Embodiments of the disclosure can be described in view of the following clauses:

1. A system, comprising:

one or more computing devices configured to:

implement a first set of programmatic interfaces enabling a client of a multi-tenant stream management service to select, for a particular data stream, a data ingestion policy from among a plurality of data ingestion policies, wherein the plurality of data ingestion policies includes an at-least-once ingestion policy in accordance with which a record submitter transmits an indication of a data record one or more times to the stream management service until a positive acknowledgement is received;

implement a second set of programmatic interfaces enabling the client to select, for the particular data stream, a data persistence policy from among a plurality of data persistence policies, wherein the plurality of data persistence policies comprises a multiple-replica persistence policy, in accordance with which multiple copies of the data record are to be stored at respective storage locations by the stream management service;

receive, at the stream management service via respective programmatic interfaces of the first and second set, a first indication that the client has selected the at-least-once ingestion policy for the particular data stream and a second indication that the client has selected the multiple-replica persistence policy for the particular data stream;

in response to a plurality of transmissions indicating a particular data record to the stream management service,

send at least one positive acknowledgement corresponding to the plurality of transmissions in accordance with the at-least-once ingestion policy; and

store, in response to a particular transmission of the plurality of transmissions, copies of the particular data record at a plurality of storage locations in accordance with the multiple-replica persistence policy.

2. The system as recited in clause 1, wherein the plurality of data ingestion policies comprises a best-effort ingestion policy, in accordance with which the stream management service is to accept and store at least some data records of a specified stream without providing corresponding positive acknowledgements to a record submitter.

3. The system as recited in clause 1, wherein the multiple-replica persistence policy selected by the client comprises an indication of a type of storage location to be used for storing copies of data records, wherein the type of storage location comprises one of: (a) magnetic disk-based storage, (b) solid state drive (SSD), (c) volatile RAM (random access memory), (d) non-volatile RAM, (e) a database management system, or (f) a storage node of a network-accessible storage service implemented by a provider network.

4. The system as recited in clause 1, wherein the multiple-replica persistence policy selected by the client comprises an indication of a requested data durability level, wherein the one or more computing devices are further configured to:

select the plurality of storage locations based at least in part on the requested data durability level.

5. The system as recited in clause 1, wherein the one or more computing devices are further configured to:

determine a billing amount that a particular client is to be charged for stream management operations based at least in part on one or more of: (a) a data ingestion policy selected by the particular client, and (b) a data persistence policy selected by the particular client.

6. A method, comprising:

performing, by one or more computing devices:

implementing a set of programmatic interfaces enabling a client of a stream management service to select, for a particular data stream, a data ingestion policy from among a plurality of data ingestion policies, wherein the plurality of data ingestion policies includes an at-least-once ingestion policy in accordance with which a record submitter is to transmit an indication of a data record one or more times to the stream management service until a positive acknowledgement is received;

receiving a request via a programmatic interface of the set, indicating that the client has selected the at-least-once ingestion policy for the particular data stream;

in response to receiving a plurality of transmissions indicating a particular data record at the stream management service,

sending respective positive acknowledgements corresponding to each transmission of the plurality of transmissions in accordance with the at-least-once ingestion policy; and

storing, in response to receiving a particular transmission of the plurality of transmissions, copies of the particular data record at a one or more storage locations in accordance with the a data persistence policy selected for the particular data stream.

7. The method as recited in clause 6, wherein the plurality of data ingestion policies comprises a best-effort ingestion policy, in accordance with which the stream management service is to accept and store at least some data records of a specified stream without providing corresponding positive acknowledgements to a record submitter.

8. The method as recited in clause 6, wherein in accordance with the at-least-once ingestion policy, the stream management service is configured to eliminate duplicated data records.

9. The method as recited in clause 6, wherein in accordance with the at-least-once ingestion policy, the stream management service is configured to store, corresponding to the plurality of transmissions, more than one copy of the particular data record.

10. The method as recited in clause 6, further comprising performing, by the one or more computing devices:

implementing a second set of programmatic interfaces enabling the client to select, for the particular data stream, the data persistence policy from among a plurality of data persistence policies.

11. The method as recited in clause 10, wherein the plurality of data persistence policies include a multiple-replica persistence policy and a single-replica persistence policy.

12. The method as recited in clause 10, wherein the data persistence policy comprises an indication of a type of storage location to be used for storing data records, wherein the type of storage location comprises one of: (a) magnetic disk-based storage, (b) solid state drive (SSD), (c) volatile RAM (random access memory), (d) non-volatile RAM, (e) a database management system, or (f) a storage node of a network-accessible storage service implemented by a provider network.

13. The method as recited in clause 10, wherein the data persistence policy comprises an indication of a first type of storage location to be used for a first partition of the particular data stream, and an indication of a different type of storage location to be used for a second partition of the particular data stream.

14. The method as recited in clause 10, wherein the data persistence policy selected by the client does not include an indication of a type of storage location to be used for storing data records, further comprising performing, by the one or more computing devices:

selecting, by one or more components of the stream management service, a type of storage location to be used for storing data records.

15. The method as recited in clause 14, wherein said selecting a type of storage location comprises selecting a first type of storage location for a first partition of the particular data stream, and selecting a different type of storage location for a second partition of the particular data stream.

16. The method as recited in clause 10, wherein the data persistence policy includes an indication of a target latency, further comprising performing, by the one or more computing devices:

selecting, by one or more components of the stream management service, a type of storage location to be used for storing data records based at least in part on the target latency.

17. The method as recited in clause 10, wherein the data persistence policy selected by the client comprises an indication of a requested data durability level, further comprising performing, by the one or more computing devices:

selecting a plurality of storage locations at which data records of the stream are to be stored based at least in part on the requested data durability level.

18. The method as recited in clause 6, further comprising performing, by the one or more computing devices:

determining a billing amount that a particular client is to be charged for stream management operations based at least in part on one or more of: (a) a data ingestion policy selected by the particular client, and (b) a data persistence policy selected by the particular client.

19. The method as recited in clause 6, further comprising performing, by the one or more computing devices:

determining a number of data ingestion nodes to be configured for the particular data stream based at least in part on a partitioning policy in accordance with which (a) the particular data record is designated as a member of a particular partition based on one or more attributes of the particular data record and (b) a data ingestion node of the number of data ingestion nodes is selected to ingest data records of the particular partition.

20. The method as recited in clause 6, further comprising performing, by the one or more computing devices:

determining a number of data storage nodes to be configured for the particular data stream based at least in part on a partitioning policy in accordance with which (a) the particular data record is designated as a member of a particular partition based on one or more attributes of the particular data record and (b) a data storage node of the number of data storage nodes is selected to store data records of the particular partition.

21. The method as recited in clause 6, further comprising performing, by the one or more computing devices:

storing, corresponding to each data record of a plurality of data records of the particular data stream, including the particular data record, a respective sequence number usable to respond to read requests for ordered sets of data records;

receiving, from a submitter of a different data record of the data stream, an indication of a minimum sequence number to be stored corresponding to the different data record; and

storing, corresponding to the different data record, a particular sequence number that is greater than or equal to the minimum sequence number.

22. A non-transitory computer-accessible storage medium storing program instructions that when executed on one or more processors:

implement a set of programmatic interfaces enabling a client of a stream management service to select, for a particular data stream whose data records are ingested into the stream based on a selected data ingestion policy, a data persistence policy from among a plurality of data persistence policies, wherein the plurality of data persistence policies includes (a) a multiple-replica persistence policy in accordance with which multiple copies of a data record of the particular data stream are to be stored at respective storage locations and (b) a single-replica persistence policy in which a single copy of a data record of the particular data stream is to be stored;

receive a request via a programmatic interface of the set, indicating that the client has selected the multiple-replica persistence policy for the particular data stream; and

configure a plurality of storage nodes to implement the multiple-replica persistence policy for data records of the particular data stream.

23. The non-transitory computer-accessible storage medium as recited in clause 22, wherein at least one data persistence policy of the plurality of data persistence policies comprises an indication of a type of storage location to be used for storing data records, wherein the type of storage location comprises one of: (a) magnetic disk-based storage, (b) solid state drive (SSD), (c) volatile RAM (random access memory), (d) non-volatile RAM, (e) a database management system, or (f) a storage node of a network-accessible storage service implemented by a provider network.

24. The non-transitory computer-accessible storage medium as recited in clause 22, wherein the multiple-replica persistence policy selected by the client comprises an indication of a

requested data durability level, wherein the instructions when executed on the one or more processors:

determine a number of storage nodes to be configured based at least in part on the requested data durability level.

25. The non-transitory computer-accessible storage medium as recited in clause 22, further comprising performing, by the one or more computing devices:

determining a billing amount that a particular client is to be charged for stream management operations based at least in part on a data persistence policy selected by the particular client.

26. The non-transitory computer-accessible storage medium as recited in clause 22, further comprising performing, by the one or more computing devices:

determining a number of data storage nodes to be configured for the particular data stream based at least in part on a partitioning policy in accordance with which (a) the particular data record is designated as a member of a particular partition based on one or more attributes of the particular data record and (b) a data storage node of the number of data storage nodes is selected to store data records of the particular partition.

27. A system, comprising:

one or more computing devices configured to:

determine a partitioning policy to be applied to distribute data records of a data stream among a plurality of nodes of a multi-tenant stream management service, wherein the partitioning policy comprises an initial mapping of data records to a plurality of partitions based at least in part on one or more attribute values associated with the data records;

identify, using the initial mapping, a first partition of which a particular data record of the data stream is to be designated a member, based at least in part on a particular attribute value;

generate, corresponding to the particular data record, a sequence number indicative of a position of the particular data record within a record acquisition sequence at an ingestion node of the stream management service, wherein the ingestion node is selected based at least in part on the initial mapping;

store a plurality of data records of the first partition at a data storage location of the stream management service in an order based at least in part on

respective sequence numbers of the plurality of data records, wherein the data storage location is selected based at least in part on the initial mapping; and

in response to a determination that a triggering criterion for repartitioning the data stream has been met,

generate a modified mapping of data records to partitions,

initiate usage of the modified mapping without scheduling a pause in data record acquisitions of the data stream; and

select, for another data record with the particular attribute value, wherein the other data record is received subsequent to an initiation of usage of the modified mapping, at least one of: (a) a different ingestion node of the stream management service or (b) a different data storage location of the stream management service.

28. The system as recited in clause 27, wherein the sequence number comprises an indication of (a) a timestamp associated with ingestion of the particular data record, and (b) an additional subsequence value.

29. The system as recited in clause 28, wherein the one or more computing devices are further configured to:

select an initial timestamp value to be used for sequence numbers of data records to be mapped using the modified mapping;

in response to a data record retrieval request indicating a particular sequence number,

in response to a determination that a value of a particular timestamp indicated by the particular sequence number is lower than the initial timestamp value, utilize the initial mapping to retrieve one or more data records; and

in response to a determination that the value of the particular timestamp is not lower than the initial timestamp value, utilize the modified mapping to retrieve one or more data records.

30. The system as recited in clause 27, wherein the triggering criterion comprises one or more of: (a) a detection of an overload condition, (b) a detection of a workload imbalance, (c) a client request for repartitioning, (d) a determination of a change to a data durability requirement of the data stream, (e) a determination of a schedule of a software version change, (f) a detection of a change to a usage pattern of the data stream, (g) a determination of a pricing impact of repartitioning the data stream, or (h) a determination of a performance target associated with the data stream.

31. The system as recited in clause 27, wherein the one or more computing devices are further configured to:

receive a client request indicating one or more partitioning criteria to be used for the data stream; and

generate the initial mapping based at least in part on the client request.

32. A method, comprising:

performing, by one or more computing devices of a stream management service:

determining an initial mapping of data records of a data stream to a plurality of partitions based at least in part on one or more attribute values of the data records;

identifying, using the initial mapping, a first partition of which a particular data record of the data stream is to be designated a member, based at least in part on a particular attribute value;

storing the particular data record at a storage location selected based at least in part on the initial mapping; and

in response to determining that a triggering criterion has been met,

generating a modified mapping of data records to partitions, and

selecting, for another data record with the particular attribute value, wherein the other data record is received subsequent to an initiation of usage of the modified mapping, a different storage location.

33. The method as recited in clause 32, further comprising performing, by the one or more computing devices:

executing, on the particular data record prior to said determining that the triggering criterion has been met, a processing operation at a worker node selected based at least in part on the initial mapping; and

executing, on a different data record with the particular attribute value, subsequent to said determining that the triggering criterion has been met, the processing operation at a different worker node selected based at least in part on the modified mapping.

34. The method as recited in clause 32, further comprising performing, by the one or more computing devices:

generating, corresponding to the particular data record, a sequence number indicative of a position of the particular data record within a record acquisition sequence at an

ingestion node of the stream management service, wherein the ingestion node is selected based at least in part on the initial mapping; and

storing data records of the first partition in an order corresponding to their sequence numbers.

35. The method as recited in clause 34, wherein the sequence number comprises an indication of (a) a timestamp associated with ingestion of the particular data record, and (b) an additional subsequence value.

36. The method as recited in clause 35, wherein the timestamp is indicative of a clock time at which the particular data record was ingested, wherein the method further comprises performing, by the one or more computing devices:

in response to a retrieval request requesting one or more data records to be retrieved based at least in part on a specified record ingestion time range, using sequence numbers associated with the one or more data records as index keys to retrieve the one or more data records.

37. The method as recited in clause 35, further comprising performing, by the one or more computing devices:

selecting an initial timestamp value to be used for sequence numbers of data records to be mapped using the modified mapping;

in response to receiving a data record retrieval request indicating a particular sequence number,

in response to determining that a value of a particular timestamp indicated by the particular sequence number is lower than the initial timestamp value, utilizing the initial mapping to retrieve one or more data records; and

in response to determining that the value of the particular timestamp is not lower than the initial timestamp value, utilizing the modified mapping to retrieve one or more data records.

38. The method as recited in clause 32, wherein said modified mapping uses at least one additional attribute value to determine the partition of a data record.

39. The method as recited in clause 32, further comprising performing, by the one or more computing devices:

receiving a client request indicating one or more partitioning criteria to be used for the data stream; and

generating the initial mapping based at least in part on the client request.

40. The method as recited in clause 32, further comprising performing, by the one or more computing devices:

receiving a client request indicating the triggering criterion.

41. The method as recited in clause 32, further comprising performing by the one or more computing devices:

receiving a client request to re-partition the data stream, wherein the client request indicates one or more parameters of the modified mapping.

42. The method as recited in clause 32, further comprising performing by the one or more computing devices:

receiving a client request indicating a problem state targeted for potential resolution via re-partitioning.

43. The method as recited in clause 32, further comprising performing, by the one or more computing devices:

applying a hash function to at least a portion of contents of the particular data record to obtain a hash result expressed as a binary value comprising a selected number of bits;

determining a particular sub-range, of a range of binary values expressible using the selected number of bits, to which the hash result belongs; and

identifying the first partition based at least in part on the sub-range.

44. The method as recited in clause 32, wherein the one or more attribute values include at least one of: (a) a partition key supplied by a source of a data record, (b) an identification of a source of a data record, (c) at least a portion of contents of a data record, or (d) a network address of a source of the data record.

45. The method as recited in clause 32, further comprising performing, by the one or more computing devices:

configuring, subsequent to the generation of the modified mapping, a different number of nodes of the stream management system to perform one or more of (a) data record ingestion (b) data record storage, or (c) data record retrieval for the data stream than were configured prior to the generation of the modified mapping.

46. The method as recited in clause 32, further comprising performing, by the one or more computing devices:

storing a combined data structure representing the initial mapping and the modified mapping, wherein the combined data structure comprises (a) a first entry indicating a first partition to which a particular data record attribute is mapped in

accordance with the initial mapping, and a time range for which the initial mapping is applicable for the first partition, and (b) a second entry indicating a different partition to which the particular data record attribute is mapped in accordance with the modified mapping, and a different time range for which the modified mapping is applicable for the different partition.

47. The method as recited in clause 46, wherein the combined data structure comprises one of: (a) a tree, or (b) a directed acyclic graph.

48. The method as recited in clause 32, wherein the modified mapping comprises an indication of a merger of a pair of partitions indicated by the initial mapping.

49. A non-transitory computer-accessible storage medium storing program instructions that when executed on one or more processors:

determine a partitioning policy to be applied to distribute data records of a data stream among a plurality of nodes of a stream management service, wherein the partitioning policy comprises an indication of an initial mapping of data records to a plurality of partitions;

configure a first set of ingestion nodes of the stream management service to receive data records of the stream in accordance with the initial mapping, and a first set of data storage nodes of the stream management service to store data records in accordance with the initial mapping;

in response to a determination that a triggering criterion for dynamically repartitioning the data stream has been met,

generate a modified mapping of data records to a different plurality of partitions;

configure a different set of ingestion nodes and a different set of data storage nodes for data records received subsequent to a generation of the modified mapping; and

retain, for at least a particular time period during which arriving data records are stored in accordance with the modified mapping, data records stored at the first set of data nodes in accordance with the initial mapping.

50. The non-transitory computer-accessible storage medium as recited in clause 49, wherein the instructions when executed on the one or more processors:

configure an initial set of data retrieval nodes for the data stream in accordance with the initial mapping; and

in response to the determination that the triggering criterion has been met, configure a different set of data retrieval nodes for the data stream.

51. The non-transitory computer-accessible storage medium as recited in clause 49, wherein the triggering criterion comprises one or more of: (a) a detection of an overload condition, (b) a detection of a workload imbalance, (c) a client request for repartitioning, (d) a determination of a change to a data durability requirement of the data stream, (e) a determination of a schedule of a software version change, (f) a detection of a change to a usage pattern of the data stream, (g) a determination of a pricing impact of repartitioning the data stream, or (h) a determination of a performance target associated with the data stream.

52. The non-transitory computer-accessible storage medium as recited in clause 49, wherein the instructions when executed on the one or more processors:

receive a client request indicating one or more partitioning criteria to be used for the data stream; and

generate the initial mapping based at least in part on the client request.

53. The non-transitory computer-accessible storage medium as recited in clause 49, wherein the instructions when executed on the one or more processors:

receive a client request indicating the triggering criterion for re-partitioning the data stream.

Illustrative computer system

[00163] In at least some embodiments, a server that implements a portion or all of one or more of the technologies described herein, including the techniques to implement the components of the SMS subsystems (e.g., the ingestion, storage, retrieval and control subsystems), as well as the SPS worker and control nodes, may include a general-purpose computer system that includes or is configured to access one or more computer-accessible media. FIG. 30 illustrates such a general-purpose computing device 9000. In the illustrated embodiment, computing device 9000 includes one or more processors 9010 coupled to a system memory 9020 via an input/output (I/O) interface 9030. Computing device 9000 further includes a network interface 9040 coupled to I/O interface 9030.

[00164] In various embodiments, computing device 9000 may be a uniprocessor system including one processor 9010, or a multiprocessor system including several processors 9010 (e.g., two, four, eight, or another suitable number). Processors 9010 may be any suitable processors capable of executing instructions. For example, in various embodiments, processors 9010 may be general-purpose or embedded processors implementing any of a variety of instruction set architectures (ISAs), such as the x86, PowerPC, SPARC, or MIPS ISAs, or any other suitable ISA. In multiprocessor systems, each of processors 9010 may commonly, but not

necessarily, implement the same ISA. In some implementations, graphics processing units (GPUs) may be used instead of, or in addition to, conventional processors.

[00165] System memory 9020 may be configured to store instructions and data accessible by processor(s) 9010. In various embodiments, system memory 9020 may be implemented using any suitable memory technology, such as static random access memory (SRAM), synchronous dynamic RAM (SDRAM), nonvolatile/Flash-type memory, or any other type of memory. In the illustrated embodiment, program instructions and data implementing one or more desired functions, such as those methods, techniques, and data described above, are shown stored within system memory 9020 as code 9025 and data 9026.

[00166] In one embodiment, I/O interface 9030 may be configured to coordinate I/O traffic between processor 9010, system memory 9020, and any peripheral devices in the device, including network interface 9040 or other peripheral interfaces such as various types of persistent and/or volatile storage devices used to store physical replicas of data object partitions. In some embodiments, I/O interface 9030 may perform any necessary protocol, timing or other data transformations to convert data signals from one component (e.g., system memory 9020) into a format suitable for use by another component (e.g., processor 9010). In some embodiments, I/O interface 9030 may include support for devices attached through various types of peripheral buses, such as a variant of the Peripheral Component Interconnect (PCI) bus standard or the Universal Serial Bus (USB) standard, for example. In some embodiments, the function of I/O interface 9030 may be split into two or more separate components, such as a north bridge and a south bridge, for example. Also, in some embodiments some or all of the functionality of I/O interface 9030, such as an interface to system memory 9020, may be incorporated directly into processor 9010.

[00167] Network interface 9040 may be configured to allow data to be exchanged between computing device 9000 and other devices 9060 attached to a network or networks 9050, such as other computer systems or devices as illustrated in FIG. 1 through FIG. 29, for example. In various embodiments, network interface 9040 may support communication via any suitable wired or wireless general data networks, such as types of Ethernet network, for example. Additionally, network interface 9040 may support communication via telecommunications/telephony networks such as analog voice networks or digital fiber communications networks, via storage area networks such as Fibre Channel SANs, or via any other suitable type of network and/or protocol.

[00168] In some embodiments, system memory 9020 may be one embodiment of a computer-accessible medium configured to store program instructions and data as described above for FIG.

1 through FIG. 29 for implementing embodiments of the corresponding methods and apparatus. However, in other embodiments, program instructions and/or data may be received, sent or stored upon different types of computer-accessible media. Generally speaking, a computer-accessible medium may include non-transitory storage media or memory media such as magnetic or optical media, e.g., disk or DVD/CD coupled to computing device 9000 via I/O interface 9030. A non-transitory computer-accessible storage medium may also include any volatile or non-volatile media such as RAM (e.g. SDRAM, DDR SDRAM, RDRAM, SRAM, etc.), ROM, etc., that may be included in some embodiments of computing device 9000 as system memory 9020 or another type of memory. Further, a computer-accessible medium may include transmission media or signals such as electrical, electromagnetic, or digital signals, conveyed via a communication medium such as a network and/or a wireless link, such as may be implemented via network interface 9040. Portions or all of multiple computing devices such as that illustrated in FIG. 30 may be used to implement the described functionality in various embodiments; for example, software components running on a variety of different devices and servers may collaborate to provide the functionality. In some embodiments, portions of the described functionality may be implemented using storage devices, network devices, or special-purpose computer systems, in addition to or instead of being implemented using general-purpose computer systems. The term "computing device", as used herein, refers to at least all these types of devices, and is not limited to these types of devices.

Conclusion

[00169] Various embodiments may further include receiving, sending or storing instructions and/or data implemented in accordance with the foregoing description upon a computer-accessible medium. Generally speaking, a computer-accessible medium may include storage media or memory media such as magnetic or optical media, e.g., disk or DVD/CD-ROM, volatile or non-volatile media such as RAM (e.g. SDRAM, DDR, RDRAM, SRAM, etc.), ROM, etc., as well as transmission media or signals such as electrical, electromagnetic, or digital signals, conveyed via a communication medium such as network and/or a wireless link.

[00170] The various methods as illustrated in the Figures and described herein represent exemplary embodiments of methods. The methods may be implemented in software, hardware, or a combination thereof. The order of method may be changed, and various elements may be added, reordered, combined, omitted, modified, etc.

[00171] Various modifications and changes may be made as would be obvious to a person skilled in the art having the benefit of this disclosure. It is intended to embrace all such

modifications and changes and, accordingly, the above description to be regarded in an illustrative rather than a restrictive sense.

WHAT IS CLAIMED IS:

1. A system, comprising:

one or more computing devices configured to:

implement a first set of programmatic interfaces enabling a client of a multi-tenant stream management service to select, for a particular data stream, a data ingestion policy from among a plurality of data ingestion policies, wherein the plurality of data ingestion policies includes an at-least-once ingestion policy in accordance with which a record submitter transmits an indication of a data record one or more times to the stream management service until a positive acknowledgement is received;

implement a second set of programmatic interfaces enabling the client to select, for the particular data stream, a data persistence policy from among a plurality of data persistence policies, wherein the plurality of data persistence policies comprises a multiple-replica persistence policy, in accordance with which multiple copies of the data record are to be stored at respective storage locations by the stream management service;

receive, at the stream management service via respective programmatic interfaces of the first and second set, a first indication that the client has selected the at-least-once ingestion policy for the particular data stream and a second indication that the client has selected the multiple-replica persistence policy for the particular data stream;

in response to receiving at the stream management service a plurality of transmissions indicating a particular data record to the stream management service,

send at least one positive acknowledgement corresponding to the plurality of transmissions in accordance with the at-least-once ingestion policy; and

store, in response to a particular transmission of the plurality of transmissions, copies of the particular data record at a plurality of

storage locations in accordance with the multiple-replica persistence policy.

2. The system as recited in claim 1, wherein the plurality of data ingestion policies comprises a best-effort ingestion policy, in accordance with which the stream management service is to accept and store at least some data records of a specified stream without providing corresponding positive acknowledgements to a record submitter.

3. The system as recited in claim 1, wherein the multiple-replica persistence policy selected by the client comprises an indication of a type of storage location to be used for storing copies of data records, wherein the type of storage location comprises one of: (a) magnetic disk-based storage, (b) solid state drive (SSD), (c) volatile RAM (random access memory), (d) non-volatile RAM, (e) a database management system, or (f) a storage node of a network-accessible storage service implemented by a provider network.

4. The system as recited in claim 1, wherein the multiple-replica persistence policy selected by the client comprises an indication of a requested data durability level, wherein the one or more computing devices are further configured to:

select the plurality of storage locations based at least in part on the requested data durability level.

5. The system as recited in claim 1, wherein the one or more computing devices are further configured to:

determine a billing amount that a particular client is to be charged for stream management operations based at least in part on one or more of: (a) a data ingestion policy selected by the particular client, and (b) a data persistence policy selected by the particular client.

6. A method, comprising:

performing, by one or more computing devices:

implementing a first set of programmatic interfaces enabling a client of a stream management service to select, for a particular data stream, a data ingestion policy from among a plurality of data ingestion policies, wherein the plurality of data ingestion policies includes an at-least-once ingestion policy in accordance with which a record submitter is to transmit an indication of a data record one or more times to the stream management service until a positive acknowledgement is received;

implementing a second set of programmatic interfaces enabling the client to select, for the particular data stream, a data persistence policy from among a plurality of data persistence policies, wherein the plurality of data persistence policies comprises a multiple-replica persistence policy, in accordance with which multiple copies of the data record are to be stored at respective storage locations by the stream management service;

receiving, via respective programmatic interfaces of the first and second set, a first indication that the client has selected the at-least-once ingestion policy for the particular data stream and a second indication that the client has selected the multiple-replica persistence policy for the particular data stream;

in response to receiving a plurality of transmissions indicating a particular data record at the stream management service,

sending respective positive acknowledgements corresponding to each transmission of the plurality of transmissions in accordance with the at-least-once ingestion policy; and

storing, in response to receiving a particular transmission of the plurality of transmissions, copies of the particular data record at one or more storage locations in accordance with the multiple-replica persistence policy selected for the particular data stream.

7. The method as recited in claim 6, wherein the plurality of data ingestion policies comprises a best-effort ingestion policy, in accordance with which the stream management

service is to accept and store at least some data records of a specified stream without providing corresponding positive acknowledgements to a record submitter.

8. The method as recited in claim 6, wherein in accordance with the at-least-once ingestion policy, the stream management service is configured to eliminate duplicated data records.

9. The method as recited in claim 6, wherein in accordance with the at-least-once ingestion policy, the stream management service is configured to store, corresponding to the plurality of transmissions, more than one copy of the particular data record.

10. The method as recited in claim 6, wherein the plurality of data persistence policies include a single-replica persistence policy.

11. The method as recited in claim 6, wherein the data persistence policy comprises an indication of a type of storage location to be used for storing data records, wherein the type of storage location comprises one of: (a) magnetic disk-based storage, (b) solid state drive (SSD), (c) volatile RAM (random access memory), (d) non-volatile RAM, (e) a database management system, or (f) a storage node of a network-accessible storage service implemented by a provider network.

12. The method as recited in claim 6, wherein the data persistence policy comprises an indication of a first type of storage location to be used for a first partition of the particular data stream, and an indication of a different type of storage location to be used for a second partition of the particular data stream.

13. The method as recited in claim 6, wherein the data persistence policy selected by the client does not include an indication of a type of storage location to be used for storing data records, further comprising performing, by the one or more computing devices:

selecting, by one or more components of the stream management service, a type of storage location to be used for storing data records.

14. The method as recited in claim 13, wherein said selecting a type of storage location comprises selecting a first type of storage location for a first partition of the particular data stream, and selecting a different type of storage location for a second partition of the particular data stream.

15. The method as recited in claim 6, wherein the data persistence policy includes an indication of a target latency, further comprising performing, by the one or more computing devices:

selecting, by one or more components of the stream management service, a type of storage location to be used for storing data records based at least in part on the target latency.

16. The method as recited in claim 6, wherein the data persistence policy selected by the client comprises an indication of a requested data durability level, further comprising performing, by the one or more computing devices:

selecting a plurality of storage locations at which data records of the particular data stream are to be stored based at least in part on the requested data durability level.

17. The method as recited in claim 6, further comprising performing, by the one or more computing devices:

determining a billing amount that a particular client is to be charged for stream management operations based at least in part on one or more of: (a) a data ingestion policy selected by the particular client, and (b) a data persistence policy selected by the particular client.

18. The method as recited in claim 6, further comprising performing, by the one or more computing devices:

determining a number of data ingestion nodes to be configured for the particular data stream based at least in part on a partitioning policy in accordance with which (a) the particular data record is designated as a member of a particular partition based on one or more attributes of the particular data record and (b) a data ingestion node of the number of data ingestion nodes is selected to ingest data records of the particular partition.

19. The method as recited in claim 6, further comprising performing, by the one or more computing devices:

determining a number of data storage nodes to be configured for the particular data stream based at least in part on a partitioning policy in accordance with which (a) the particular data record is designated as a member of a particular partition based on one or more attributes of the particular data record and (b) a data storage node of the number of data storage nodes is selected to store data records of the particular partition.

20. The method as recited in claim 6, further comprising performing, by the one or more computing devices:

storing, corresponding to each data record of a plurality of data records of the particular data stream, including the particular data record, a respective sequence number usable to respond to read requests for ordered sets of data records;

receiving, from a submitter of a different data record of the data stream, an indication of a minimum sequence number to be stored corresponding to the different data record; and

storing, corresponding to the different data record, a particular sequence number that is greater than or equal to the minimum sequence number.

21. A non-transitory computer-accessible storage medium storing program instructions that, when executed on one or more processors, cause the one or more processors to:

implement a first set of programmatic interfaces enabling a client of a network-based stream management service to select, for a particular data stream, a data ingestion policy from among a plurality of data ingestion policies, wherein the plurality of data ingestion policies includes an at-least-once ingestion policy in accordance with which a record submitter is to transmit an indication of a data record one or more times to the network-based stream management service until a positive acknowledgement is received;

implement a second set of programmatic interfaces enabling the client of the network-based data stream management service to select, for the particular data stream, a data persistence policy from among a plurality of data persistence policies, wherein the plurality of data persistence policies includes (a) a multiple-replica persistence policy in accordance with which multiple copies of a data record of the particular data stream are to be stored at respective storage locations and (b) a single-replica persistence policy in which a single copy of a data record of the particular data stream is to be stored, wherein data records for the particular data stream are ingested into the particular data stream based on a selected data ingestion policy;

receive, via respective programmatic interfaces of the first and second set, a first indication that the client has selected the at-least once ingestion policy for the particular data stream and a second indication that the client has selected the multiple-replica persistence policy for the particular data stream; and

in response to receiving at the stream management service a plurality of transmissions indicating a particular data record to the stream management service,

send at least one positive acknowledgement corresponding to the plurality of transmissions in accordance with the at-least-once ingestion policy; and

store, in response to a particular transmission of the plurality of transmissions, copies of the particular data record at a plurality of storage locations in accordance with the multiple-replica persistence policy.

22. The non-transitory computer-accessible storage medium as recited in claim 21, wherein at least one data persistence policy of the plurality of data persistence policies comprises an indication of a type of storage location to be used for storing data records, wherein the type of storage location comprises one of: (a) magnetic disk-based storage, (b) solid state drive (SSD), (c) volatile RAM (random access memory), (d) non-volatile RAM, (e) a database management system, or (f) a storage node of a network-accessible storage service implemented by a provider network.

23. The non-transitory computer-accessible storage medium as recited in claim 21, wherein the multiple-replica persistence policy selected by the client comprises an indication of a requested data durability level, wherein the instructions, when executed on the one or more processors, cause the one or more processors to:

determine a number of storage nodes to be configured based at least in part on the requested data durability level.

24. The non-transitory computer-accessible storage medium as recited in claim 21, wherein the instructions, when executed on the one or more processors, cause the one or more processors to:

determine a number of data storage nodes to be configured for the particular data stream based at least in part on a partitioning policy in accordance with which (a) a particular data record is designated as a member of a particular partition based on one or more attributes of the particular data record and (b) a data storage node of the number of data storage nodes is selected to store data records of the particular partition.

25. The non-transitory computer-accessible storage medium as recited in claim 21, wherein the instructions, when executed on the one or more processors, cause the one or more processors to:

implement a different set of programmatic interfaces enabling a different client of the network-based stream service to select, for a different data stream, a data persistence policy from among the plurality of data persistence policies;

receive, via a different programmatic interface of the set, a request indicating that a different client of the network-based stream service has selected a different data persistence policy for a different data stream; and

configure the plurality of storage nodes to implement the different data persistence policy for data records of the different data stream.

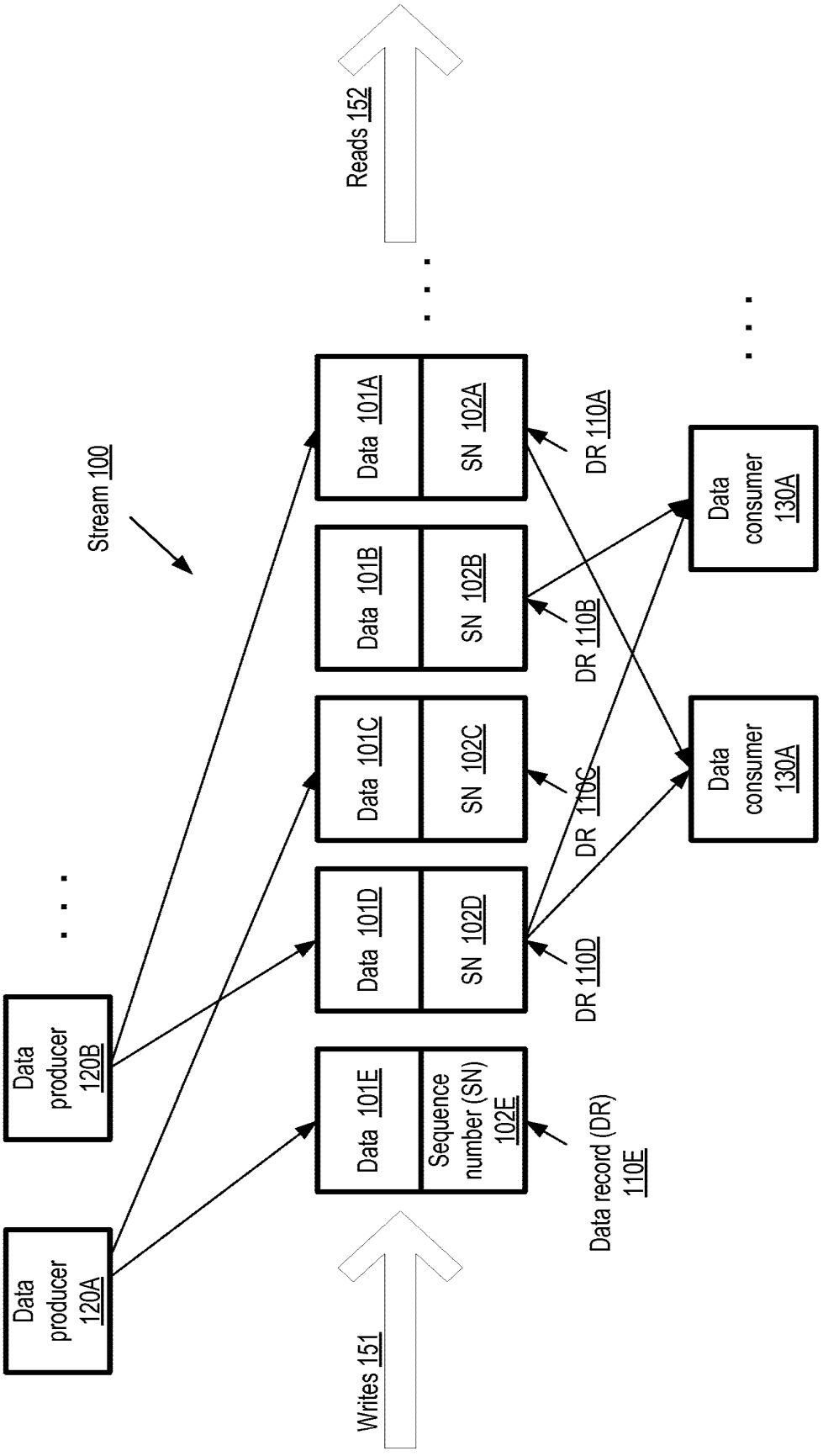

Reads 152
Stream 100
Data 101A
SN 102A
DR 110A
Data 101B
SN 102B
DR 110B
Data 101C
SN 102C
DR 110C
Data 101D
SN 102D
DR 110D
Data 101E
Sequence number (SN) 102E
Data record (DR) 110E
Data producer 120B
Data producer 120A
Data consumer 130A
Data consumer 130A
Writes 151

FIG. 1

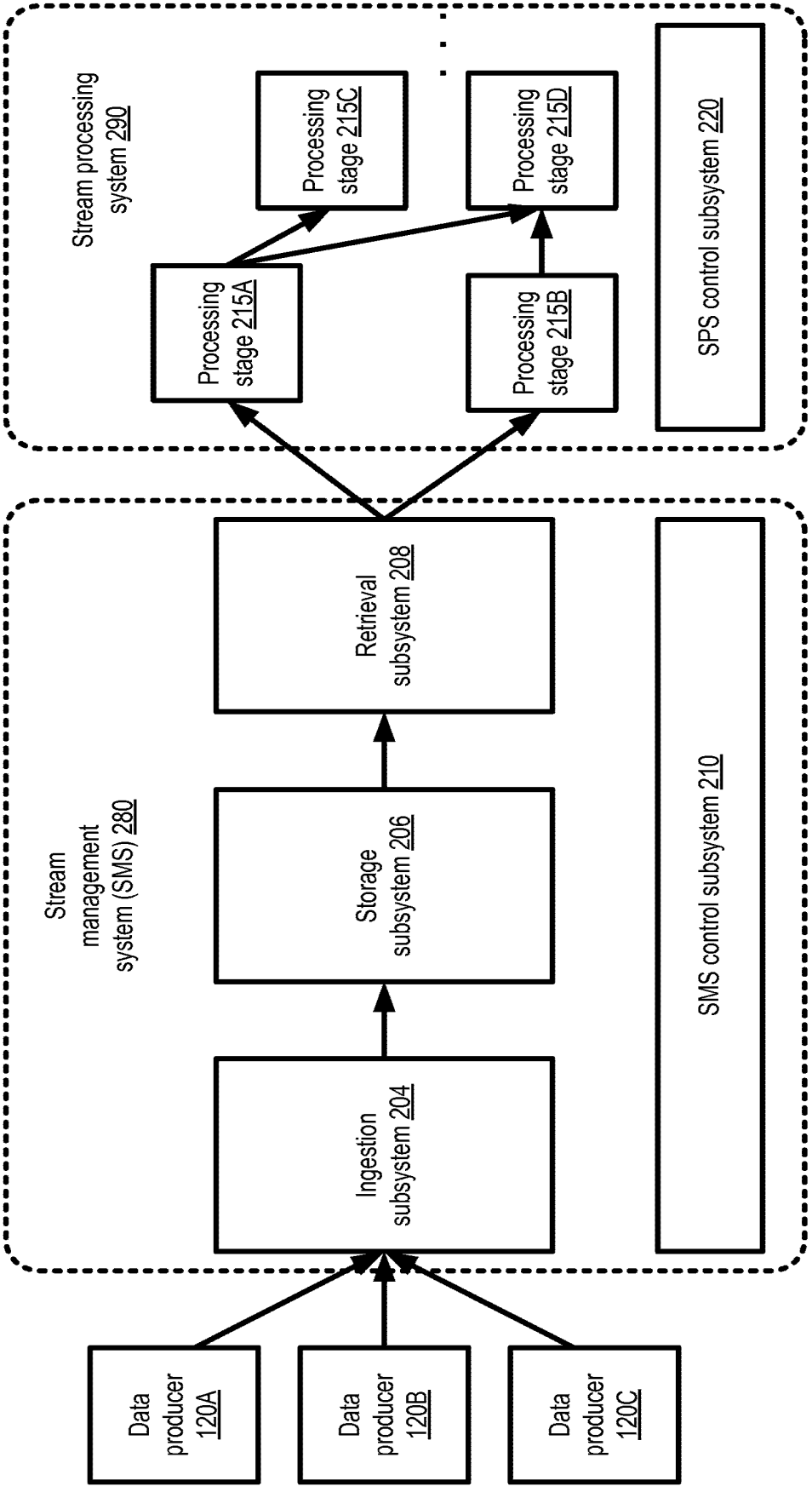

Stream processing system 290
Processing stage 215A
Processing stage 215C
Processing stage 215B
Processing stage 215D
SPS control subsystem 220
Stream management system (SMS) 280
Retrieval subsystem 208
Storage subsystem 206
Ingestion subsystem 204
SMS control subsystem 210
Data producer 120A
Data producer 120B
Data producer 120C

FIG. 2

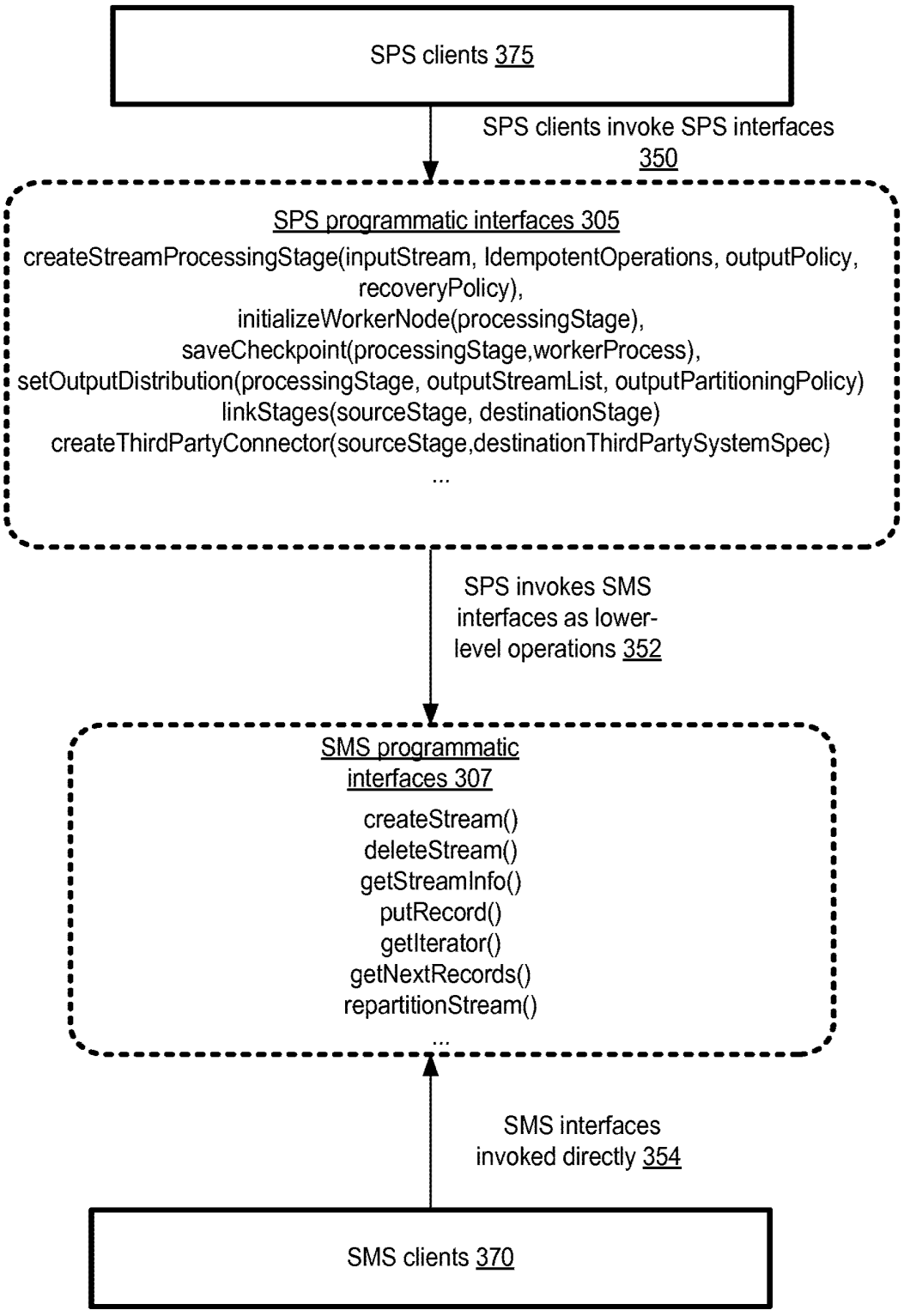

SPS clients 375
SPS clients invoke SPS interfaces 350
SPS programmatic interfaces 305
createStreamProcessingStage(inputStream, IdempotentOperations, outputPolicy, recoveryPolicy),
initializeWorkerNode(processingStage),
saveCheckpoint(processingStage,workerProcess),
setOutputDistribution(processingStage, outputStreamList, outputPartitioningPolicy)
linkStages(sourceStage, destinationStage)
createThirdPartyConnector(sourceStage,destinationThirdPartySystemSpec)
...
SPS invokes SMS interfaces as lower-level operations 352
SMS programmatic interfaces 307
createStream()
deleteStream()
getStreamInfo()
putRecord()
getIterator()
getNextRecords()
repartitionStream()
...
SMS interfaces invoked directly 354
SMS clients 370

FIG. 3

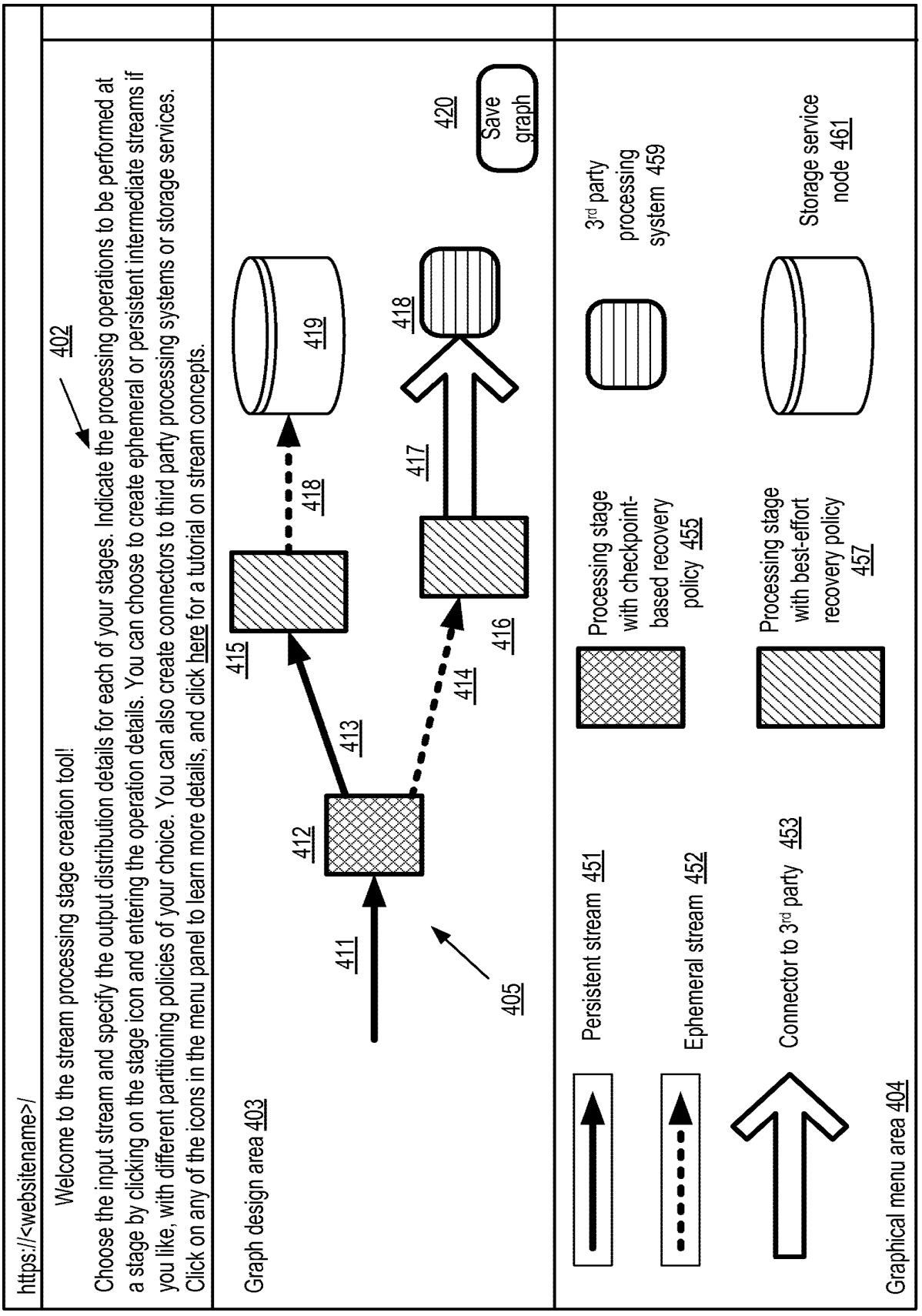

https://<websitename>/
Welcome to the stream processing stage creation tool!
402
Choose the input stream and specify the output distribution details for each of your stages. Indicate the processing operations to be performed at a stage by clicking on the stage icon and entering the operation details. You can choose to create ephemeral or persistent intermediate streams if you like, with different partitioning policies of your choice. You can also create connectors to third party processing systems or storage services. Click on any of the icons in the menu panel to learn more details, and click here for a tutorial on stream concepts.
Graph design area 403
411
412
413
414
415
416
417
418
419
420
Save graph
405
Graphical menu area 404
Persistent stream 451
Ephemeral stream 452
Connector to 3rd party 453
Processing stage with checkpoint-based recovery policy 455
Processing stage with best-effort recovery policy 457
3rd party processing system 459
Storage service node 461
400

FIG. 4

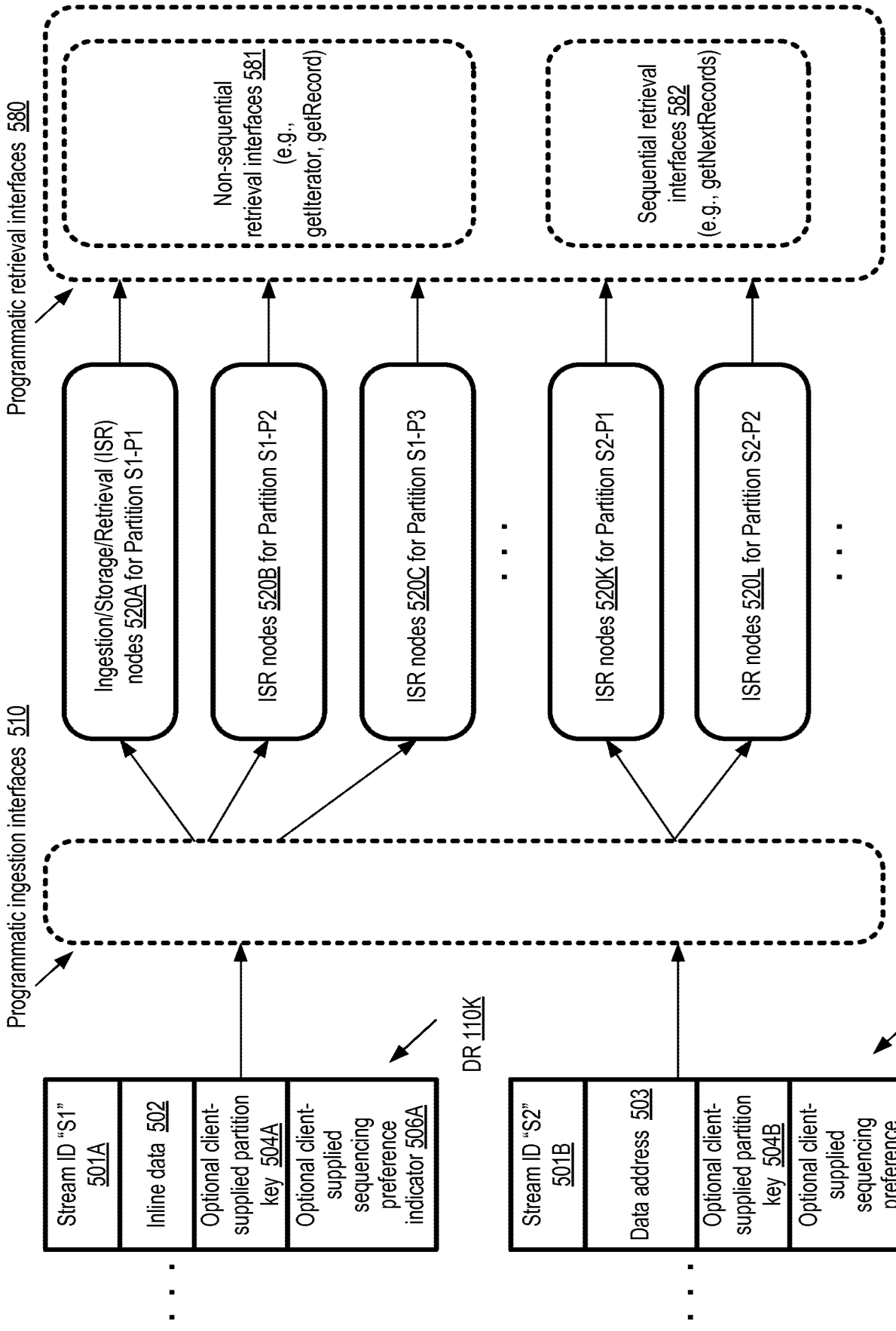

Programmatic retrieval interfaces 580
Non-sequential retrieval interfaces 581 (e.g., getIterator, getRecord)
Sequential retrieval interfaces 582 (e.g., getNextRecords)
Programmatic ingestion interfaces 510
Ingestion/Storage/Retrieval (ISR) nodes 520A for Partition S1-P1
ISR nodes 520B for Partition S1-P2
ISR nodes 520C for Partition S1-P3
ISR nodes 520K for Partition S2-P1
ISR nodes 520L for Partition S2-P2
DR 110K
DR 110Q
Stream ID "S1" 501A
Inline data 502
Optional client-supplied partition key 504A
Optional client-supplied sequencing preference indicator 506A
Stream ID "S2" 501B
Data address 503
Optional client-supplied partition key 504B
Optional client-supplied sequencing preference indicator 506B

FIG. 5

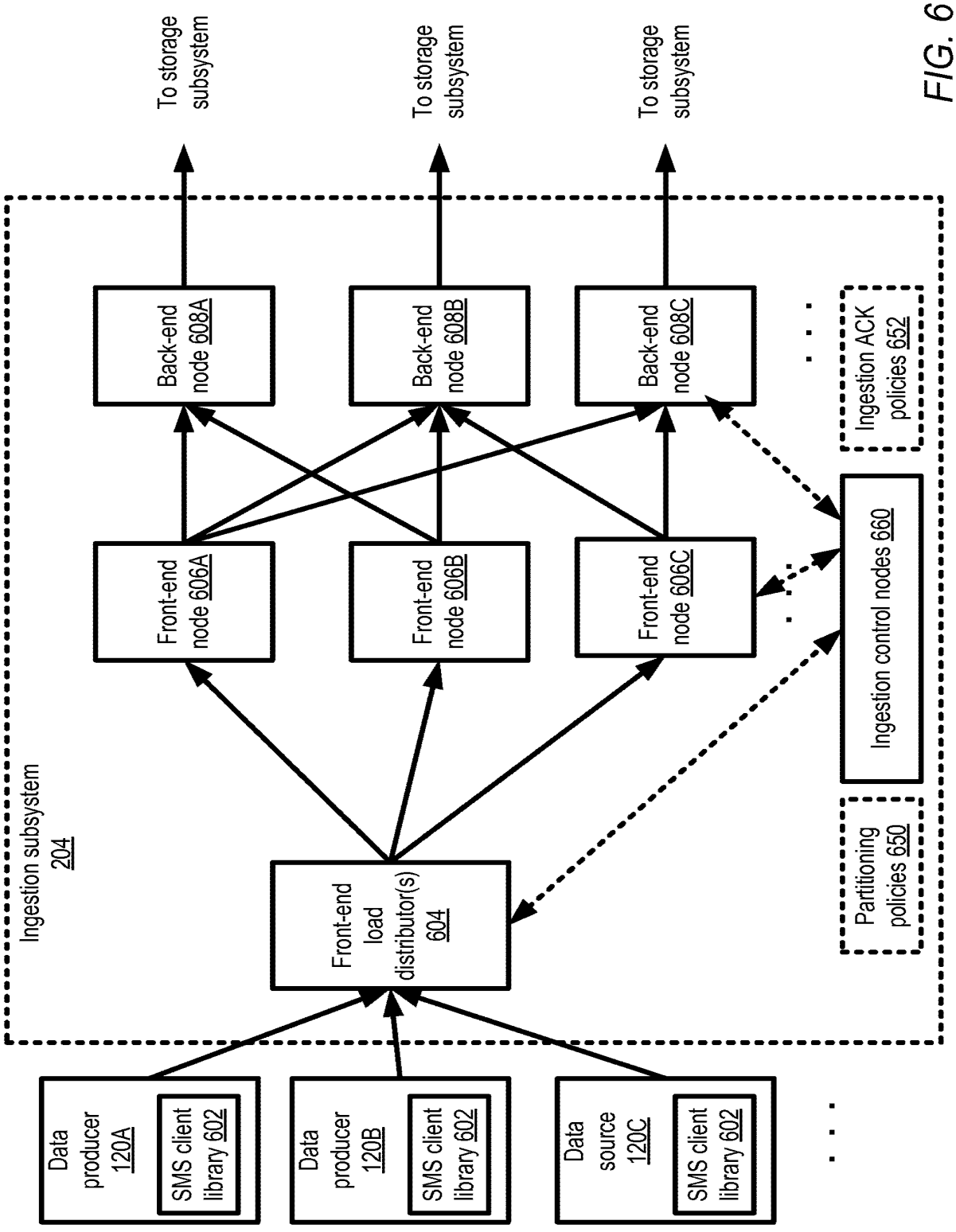

To storage subsystem
To storage subsystem
To storage subsystem
Back-end node 608A
Back-end node 608B
Back-end node 608C
Front-end node 606A
Front-end node 606B
Front-end node 606C
Ingestion ACK policies 652
Ingestion control nodes 660
Partitioning policies 650
Front-end load distributor(s) 604
Ingestion subsystem 204
Data producer 120A
SMS client library 602
Data producer 120B
SMS client library 602
Data source 120C
SMS client library 602

FIG. 6

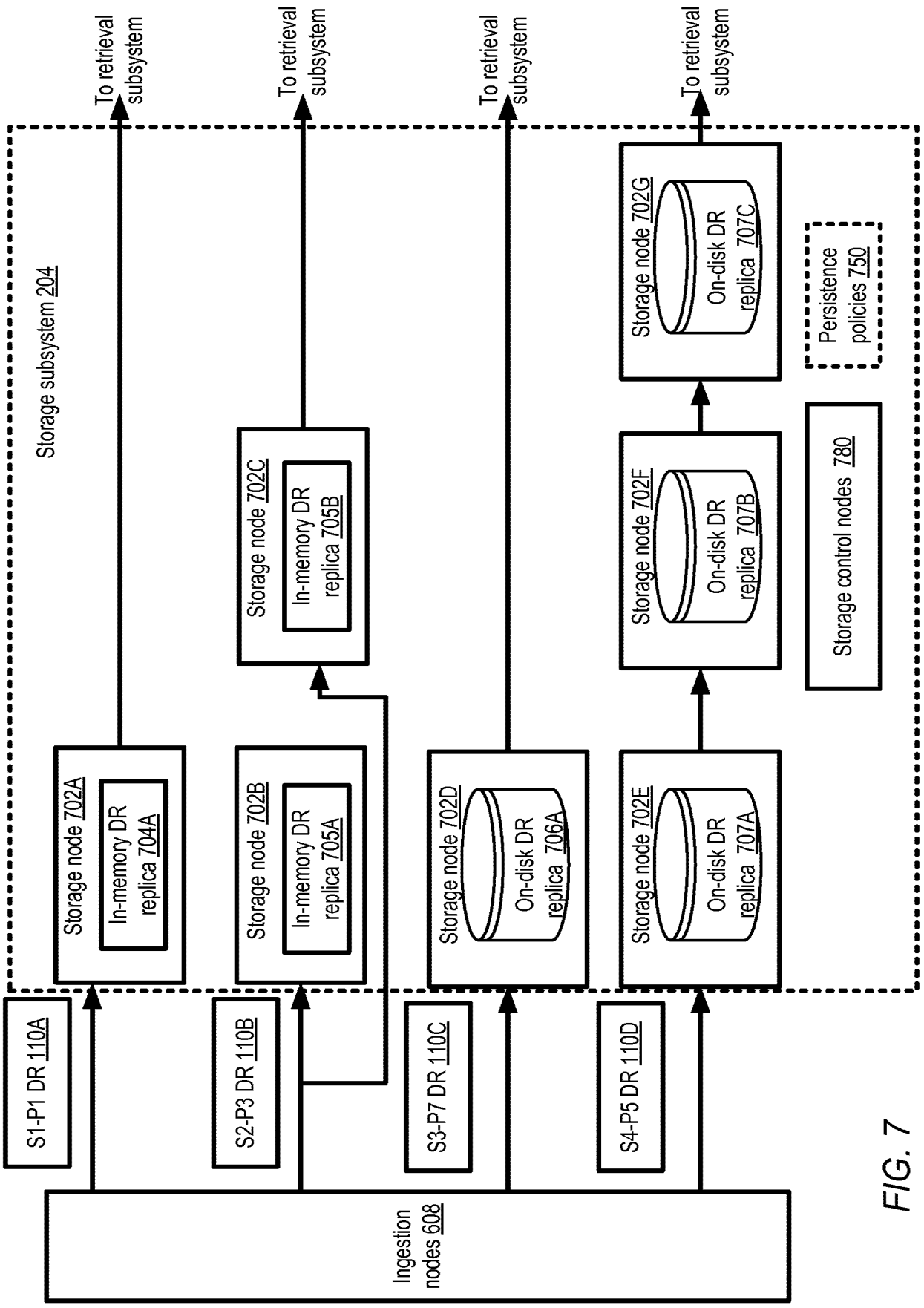

To retrieval subsystem
To retrieval subsystem
To retrieval subsystem
To retrieval subsystem
Storage subsystem 204
Storage node 702A
In-memory DR replica 704A
Storage node 702B
In-memory DR replica 705A
Storage node 702C
In-memory DR replica 705B
Storage node 702D
On-disk DR replica 706A
Storage node 702E
On-disk DR replica 707A
Storage node 702F
On-disk DR replica 707B
Storage node 702G
On-disk DR replica 707C
Persistence policies 750
Storage control nodes 780
S1-P1 DR 110A
S2-P3 DR 110B
S3-P7 DR 110C
S4-P5 DR 110D
Ingestion nodes 608

FIG. 7

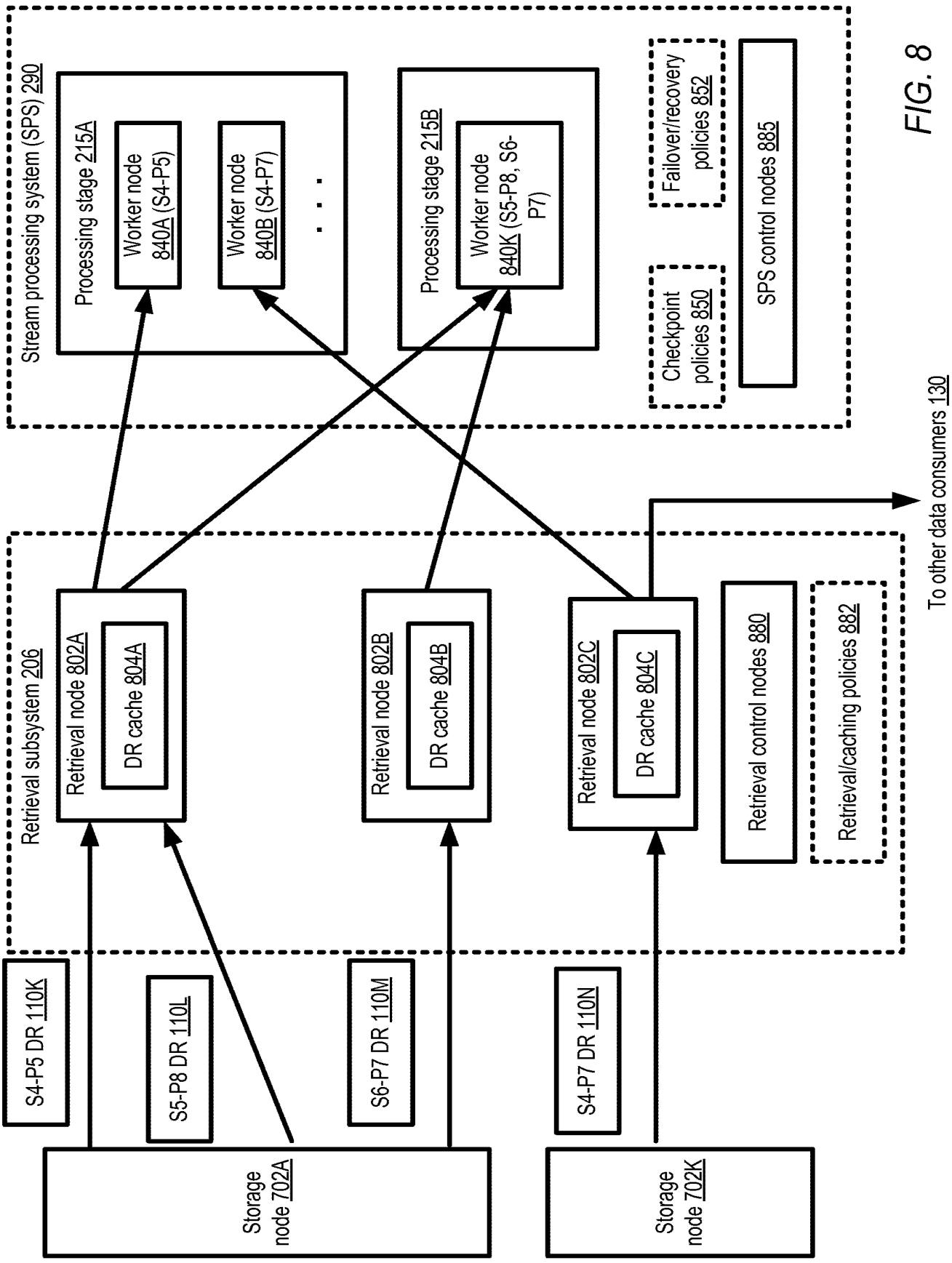

Stream processing system (SPS) 290
Processing stage 215A
Worker node 840A (S4-P5)
Worker node 840B (S4-P7)
Processing stage 215B
Worker node 840K (S5-P8, S6-P7)
Failover/recovery policies 852
SPS control nodes 885
Checkpoint policies 850
Retrieval subsystem 206
Retrieval node 802A
DR cache 804A
Retrieval node 802B
DR cache 804B
Retrieval node 802C
DR cache 804C
Retrieval control nodes 880
Retrieval/caching policies 882
To other data consumers 130
S4-P5 DR 110K
S5-P8 DR 110L
S6-P7 DR 110M
S4-P7 DR 110N
Storage node 702A
Storage node 702K

FIG. 8

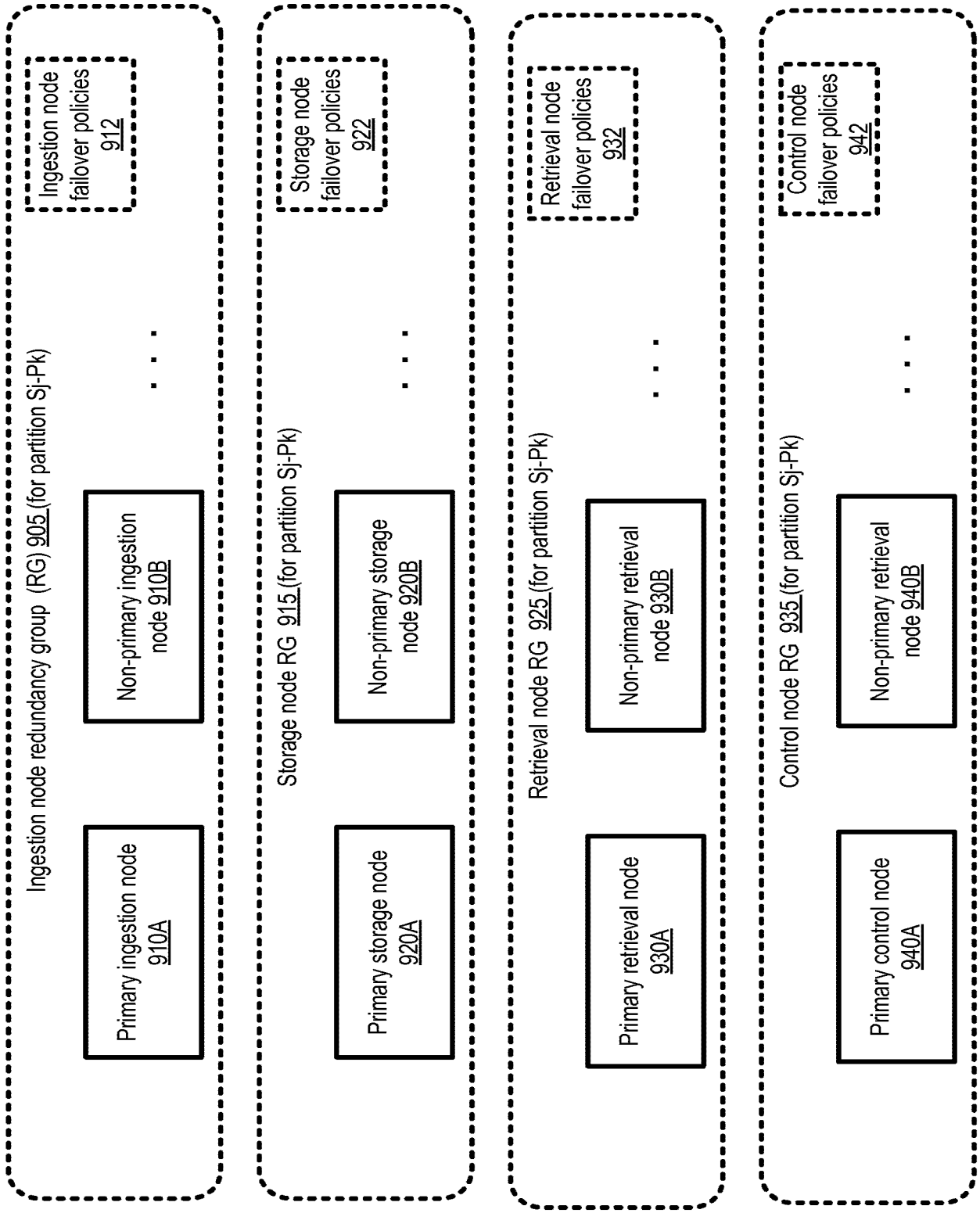

Ingestion node redundancy group (RG) 905 (for partition Sj-Pk)
Ingestion node failover policies 912
Primary ingestion node 910A
Non-primary ingestion node 910B
Storage node RG 915 (for partition Sj-Pk)
Storage node failover policies 922
Primary storage node 920A
Non-primary storage node 920B
Retrieval node RG 925 (for partition Sj-Pk)
Retrieval node failover policies 932
Primary retrieval node 930A
Non-primary retrieval node 930B
Control node RG 935 (for partition Sj-Pk)
Control node failover policies 942
Primary control node 940A
Non-primary retrieval node 940B

FIG. 9

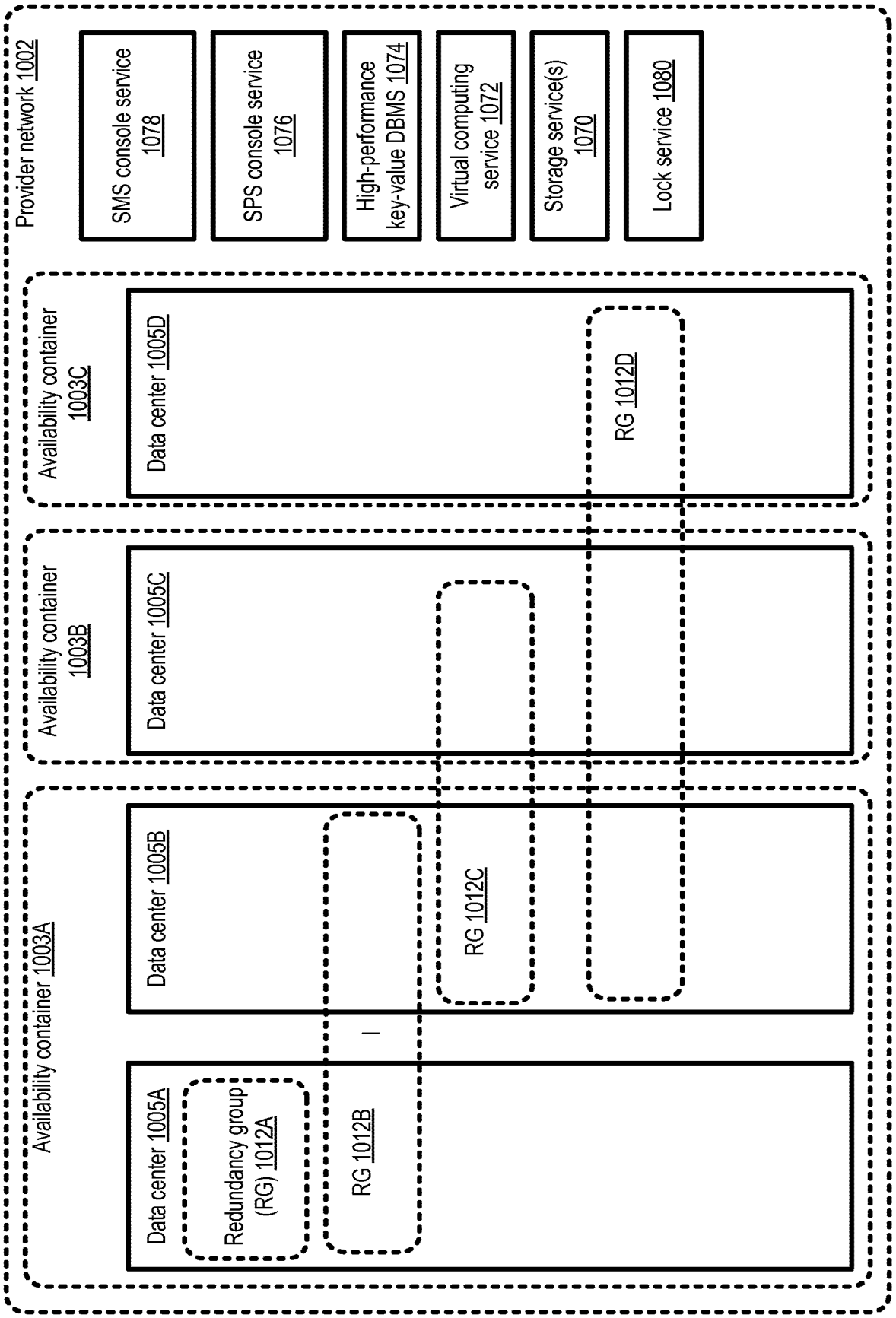

Provider network 1002
SMS console service 1078
SPS console service 1076
High-performance key-value DBMS 1074
Virtual computing service 1072
Storage service(s) 1070
Lock service 1080
Availability container 1003C
Data center 1005D
RG 1012D
Availability container 1003B
Data center 1005C
RG 1012C
Availability container 1003A
Data center 1005B
Data center 1005A
Redundancy group (RG) 1012A
RG 1012B

FIG. 10

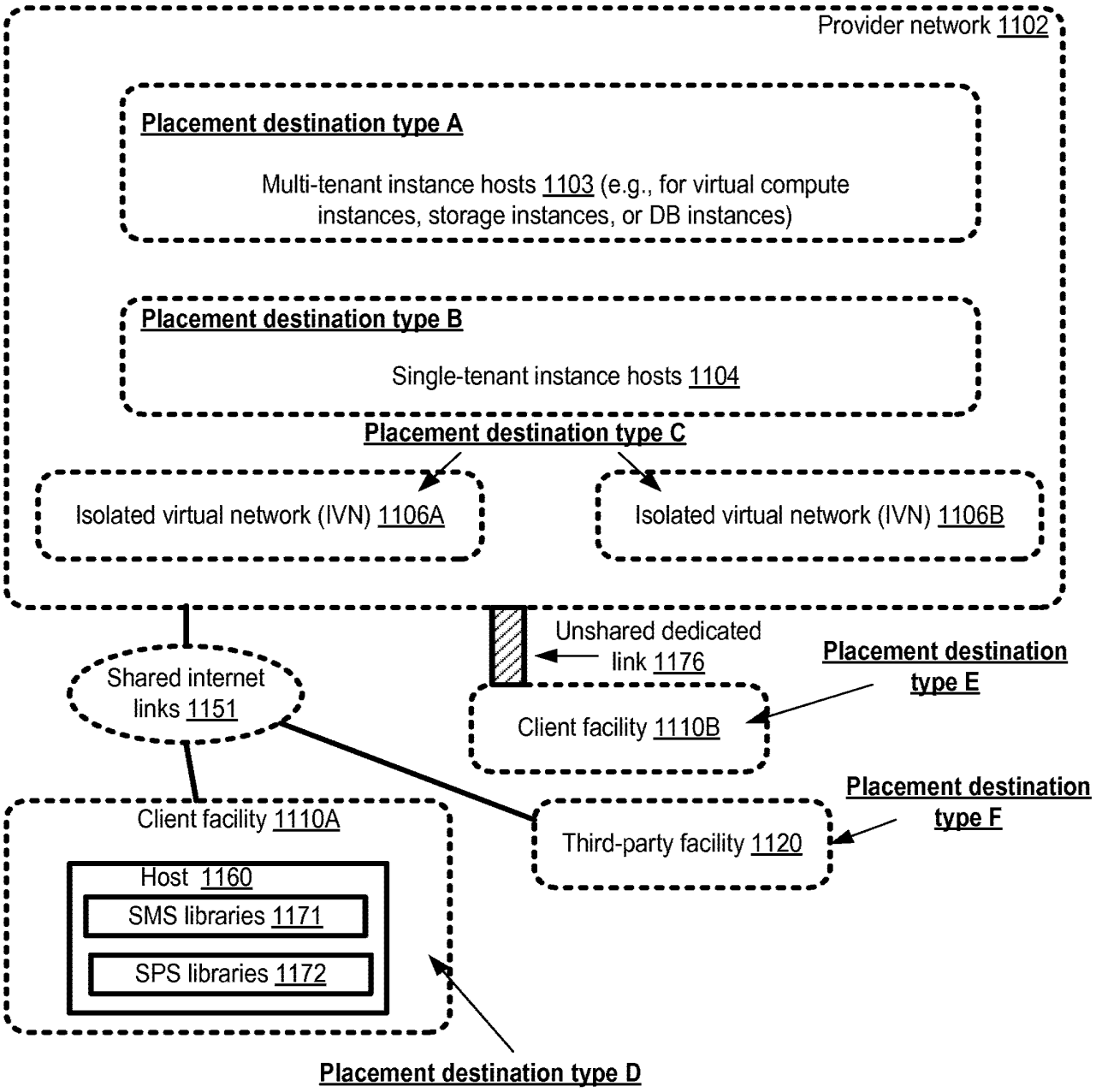

Provider network 1102
Placement destination type A
Multi-tenant instance hosts 1103 (e.g., for virtual compute instances, storage instances, or DB instances)
Placement destination type B
Single-tenant instance hosts 1104
Placement destination type C
Isolated virtual network (IVN) 1106A
Isolated virtual network (IVN) 1106B
Unshared dedicated link 1176
Placement destination type E
Shared internet links 1151
Client facility 1110B
Placement destination type F
Client facility 1110A
Third-party facility 1120
Host 1160
SMS libraries 1171
SPS libraries 1172
Placement destination type D

FIG. 11

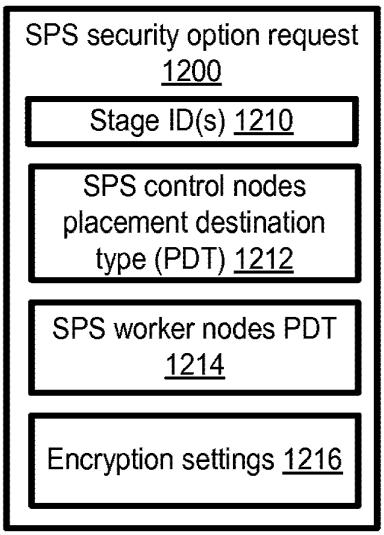


FIG. 12a

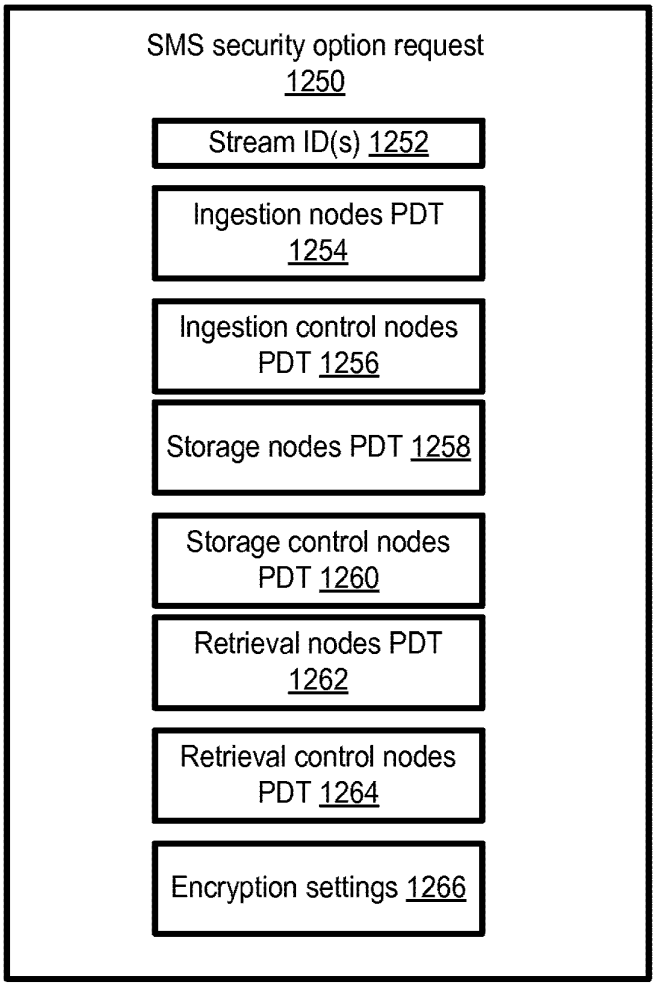


FIG. 12b

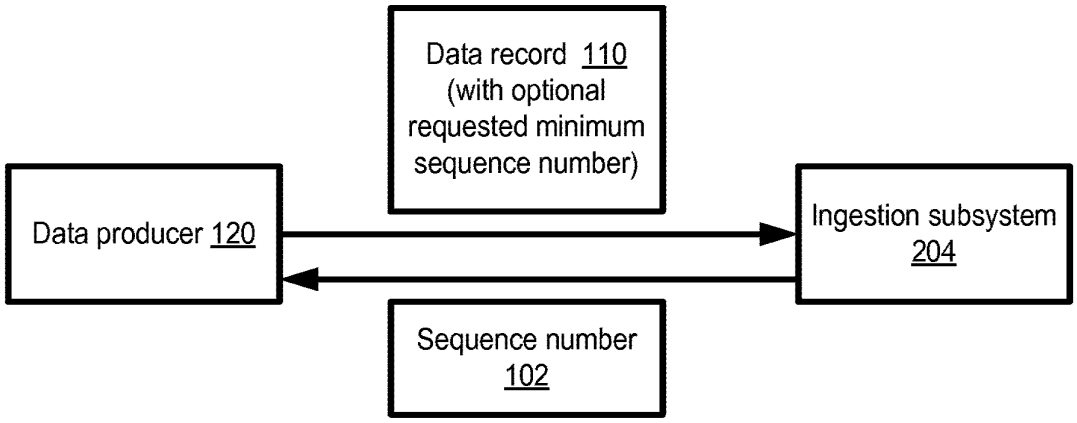


FIG. 13a

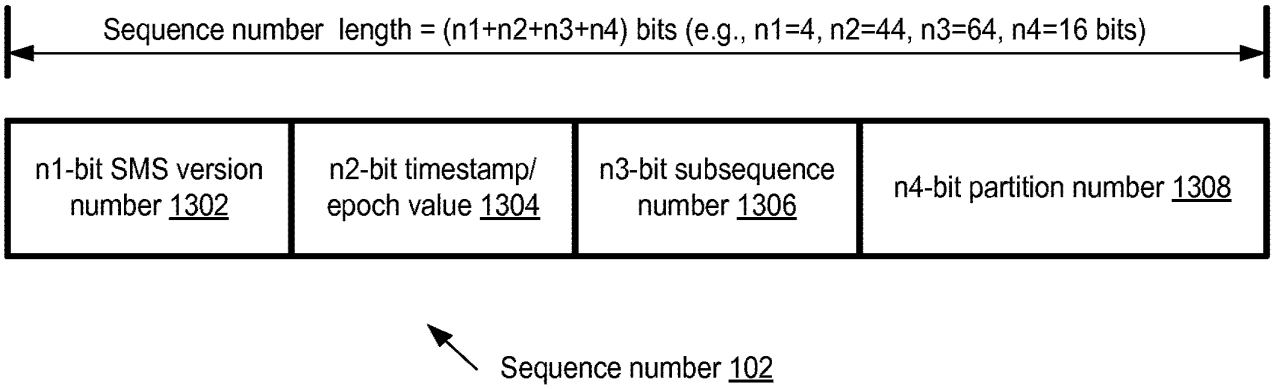


FIG. 13b

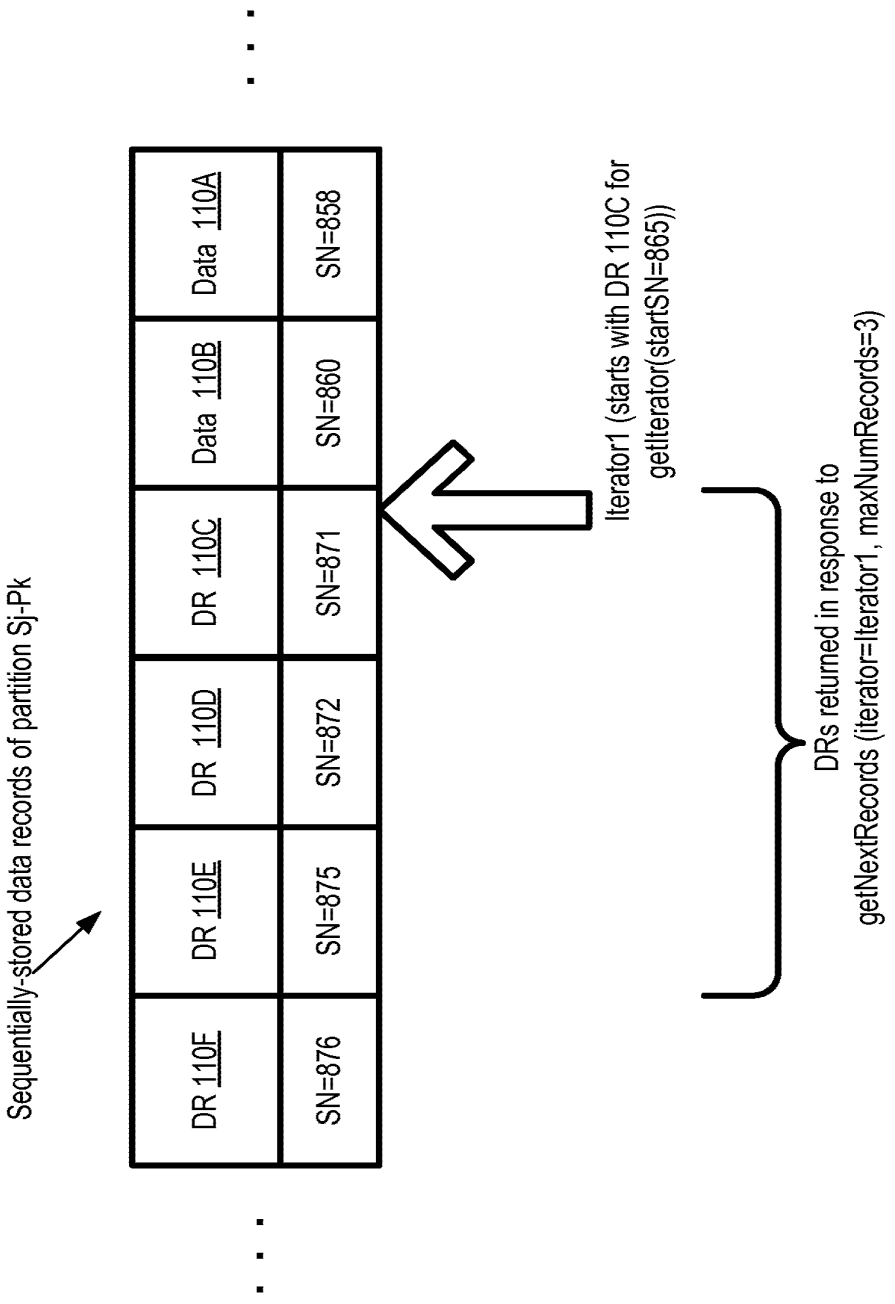

Sequentially-stored data records of partition Sj-Pk
DR 110F
DR 110E
DR 110D
DR 110C
Data 110B
Data 110A
SN=876
SN=875
SN=872
SN=871
SN=860
SN=858
Iterator1 (starts with DR 110C for getIterator(startSN=865))
DRs returned in response to getNextRecords (iterator=Iterator1, maxNumRecords=3)

FIG. 14

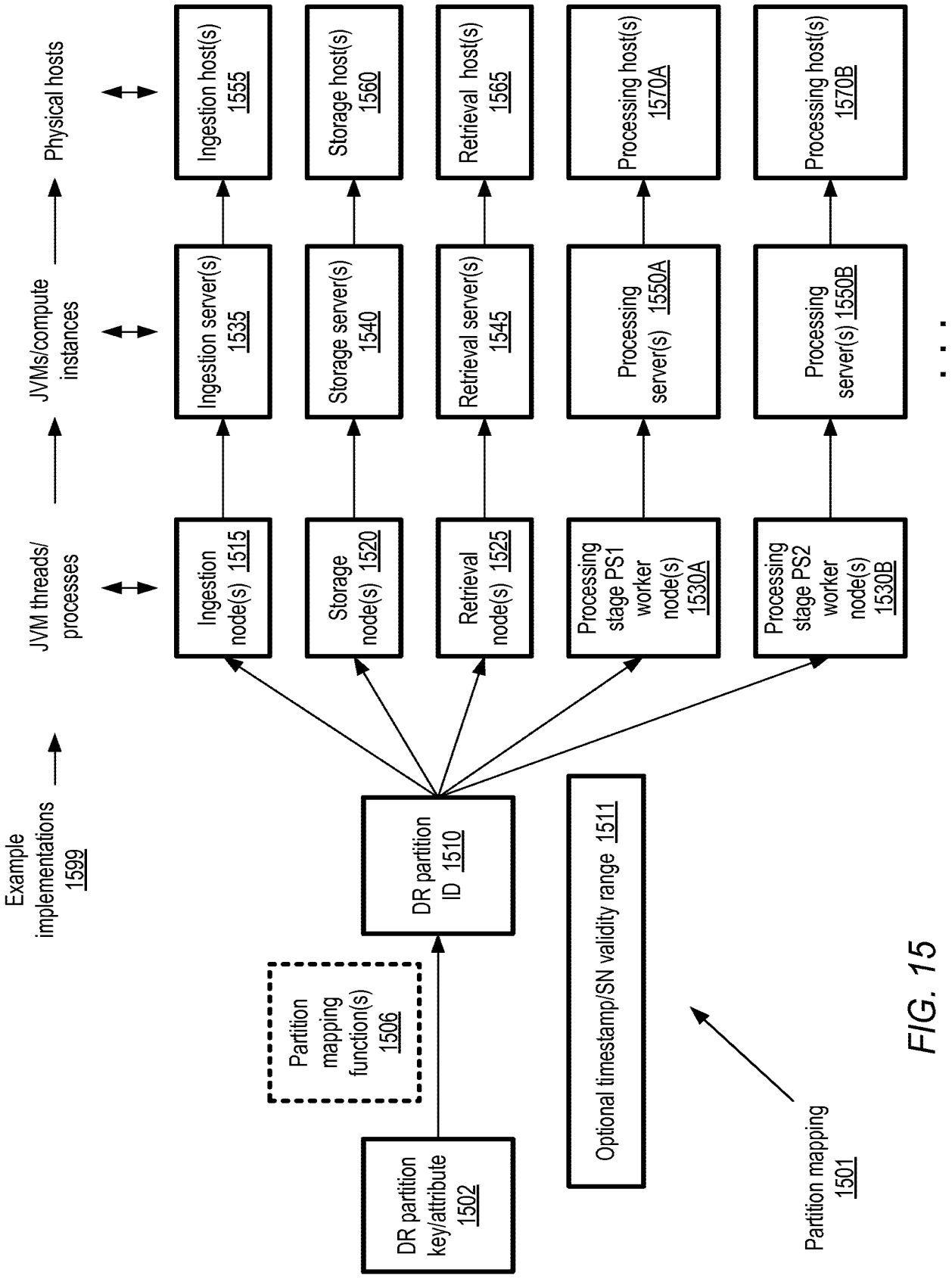

Example implementations 1599
JVM threads/ processes
JVMs/compute instances
Physical hosts
Ingestion node(s) 1515
Storage node(s) 1520
Retrieval node(s) 1525
Processing stage PS1 worker node(s) 1530A
Processing stage PS2 worker node(s) 1530B
Ingestion server(s) 1535
Storage server(s) 1540
Retrieval server(s) 1545
Processing server(s) 1550A
Processing server(s) 1550B
Ingestion host(s) 1555
Storage host(s) 1560
Retrieval host(s) 1565
Processing host(s) 1570A
Processing host(s) 1570B
DR partition ID 1510
Partition mapping function(s) 1506
DR partition key/attribute 1502
Optional timestamp/SN validity range 1511
Partition mapping 1501

FIG. 15

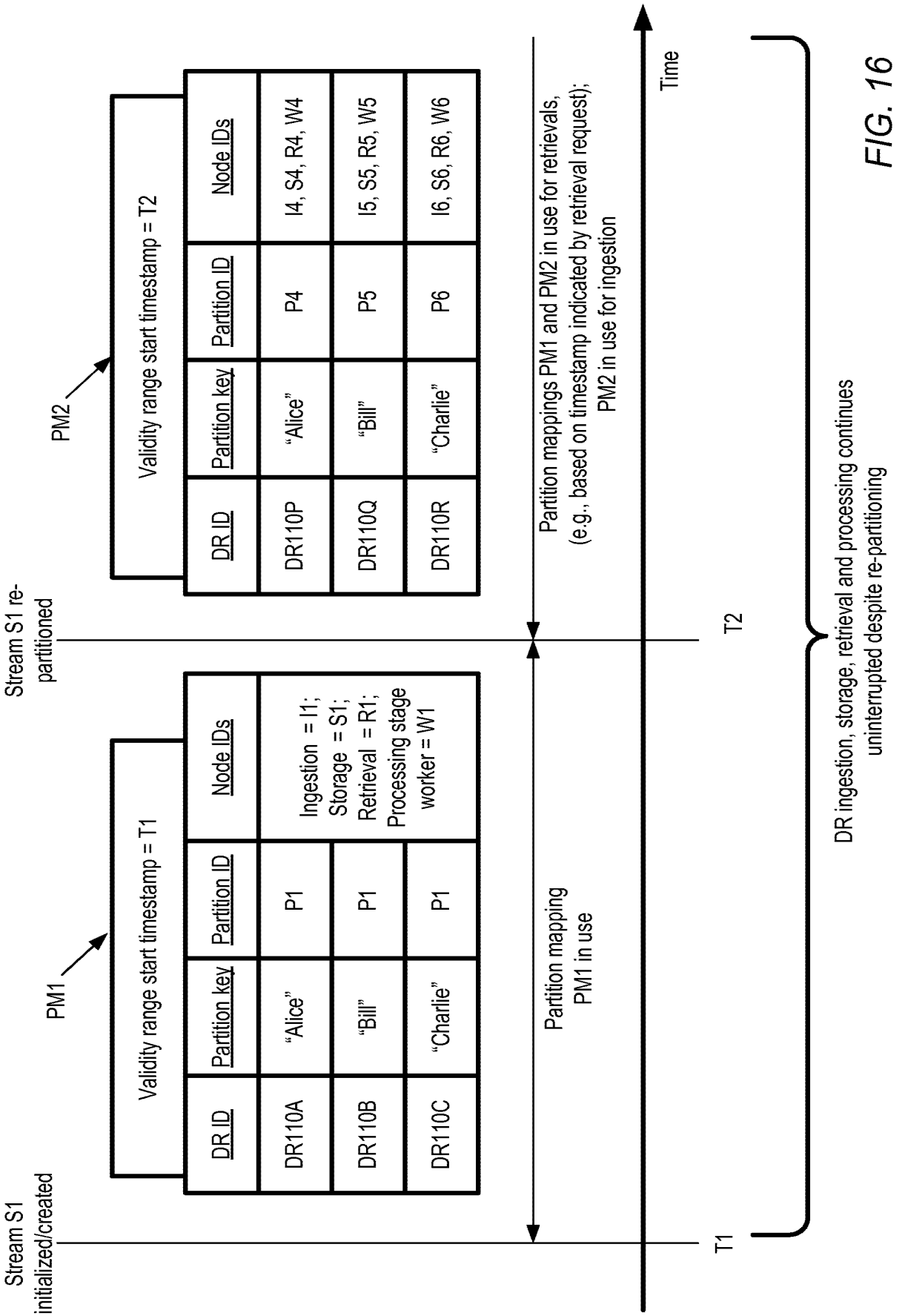

Stream S1 initialized/created
PM1
Validity range start timestamp = T1
DR ID
Partition key
Partition ID
Node IDs
DR110A
"Alice"
P1
Ingestion = I1; Storage = S1; Retrieval = R1; Processing stage worker = W1
DR110B
"Bill"
P1
DR110C
"Charlie"
P1
Partition mapping PM1 in use
T1
Stream S1 re-partitioned
PM2
Validity range start timestamp = T2
DR ID
Partition key
Partition ID
Node IDs
DR110P
"Alice"
P4
I4, S4, R4, W4
DR110Q
"Bill"
P5
I5, S5, R5, W5
DR110R
"Charlie"
P6
I6, S6, R6, W6
Partition mappings PM1 and PM2 in use for retrievals, (e.g., based on timestamp indicated by retrieval request); PM2 in use for ingestion
T2
Time
DR ingestion, storage, retrieval and processing continues uninterrupted despite re-partitioning

FIG. 16

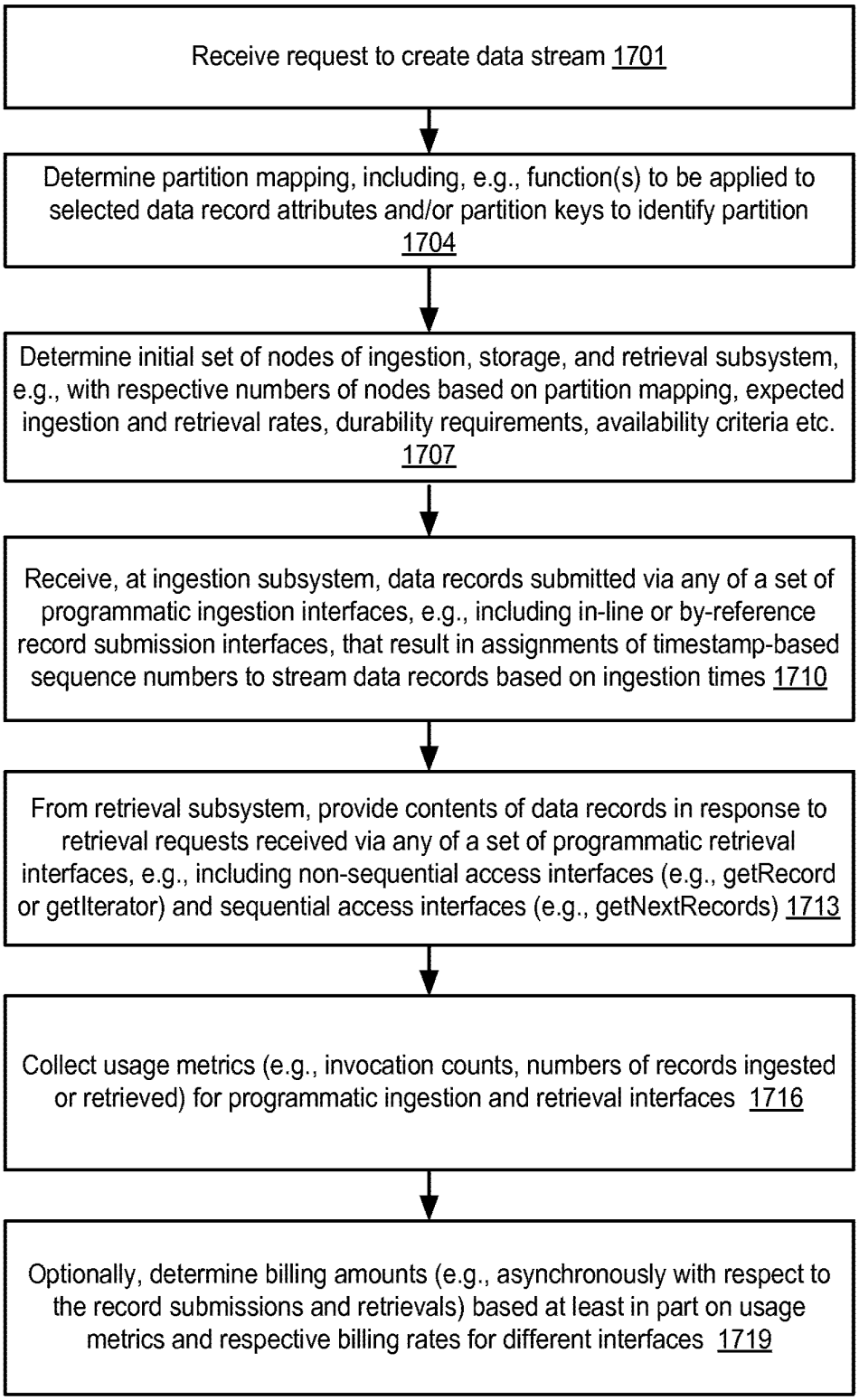

Receive request to create data stream 1701
Determine partition mapping, including, e.g., function(s) to be applied to selected data record attributes and/or partition keys to identify partition 1704
Determine initial set of nodes of ingestion, storage, and retrieval subsystem, e.g., with respective numbers of nodes based on partition mapping, expected ingestion and retrieval rates, durability requirements, availability criteria etc. 1707
Receive, at ingestion subsystem, data records submitted via any of a set of programmatic ingestion interfaces, e.g., including in-line or by-reference record submission interfaces, that result in assignments of timestamp-based sequence numbers to stream data records based on ingestion times 1710
From retrieval subsystem, provide contents of data records in response to retrieval requests received via any of a set of programmatic retrieval interfaces, e.g., including non-sequential access interfaces (e.g., getRecord or getIterator) and sequential access interfaces (e.g., getNextRecords) 1713
Collect usage metrics (e.g., invocation counts, numbers of records ingested or retrieved) for programmatic ingestion and retrieval interfaces 1716
Optionally, determine billing amounts (e.g., asynchronously with respect to the record submissions and retrievals) based at least in part on usage metrics and respective billing rates for different interfaces 1719

FIG. 17

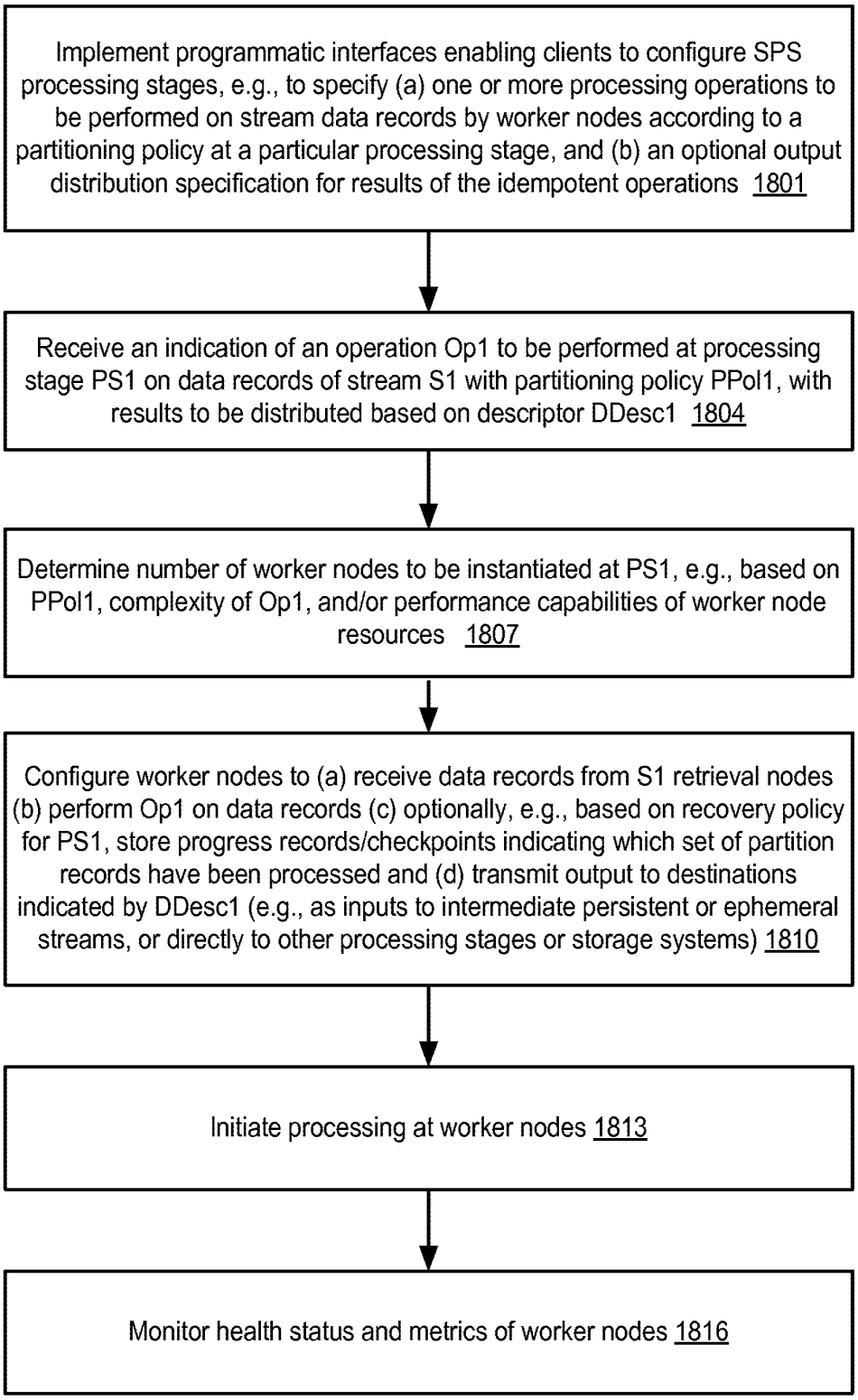

Implement programmatic interfaces enabling clients to configure SPS processing stages, e.g., to specify (a) one or more processing operations to be performed on stream data records by worker nodes according to a partitioning policy at a particular processing stage, and (b) an optional output distribution specification for results of the idempotent operations 1801
Receive an indication of an operation Op1 to be performed at processing stage PS1 on data records of stream S1 with partitioning policy PPol1, with results to be distributed based on descriptor DDesc1 1804
Determine number of worker nodes to be instantiated at PS1, e.g., based on PPol1, complexity of Op1, and/or performance capabilities of worker node resources 1807
Configure worker nodes to (a) receive data records from S1 retrieval nodes (b) perform Op1 on data records (c) optionally, e.g., based on recovery policy for PS1, store progress records/checkpoints indicating which set of partition records have been processed and (d) transmit output to destinations indicated by DDesc1 (e.g., as inputs to intermediate persistent or ephemeral streams, or directly to other processing stages or storage systems) 1810
Initiate processing at worker nodes 1813
Monitor health status and metrics of worker nodes 1816

FIG. 18a

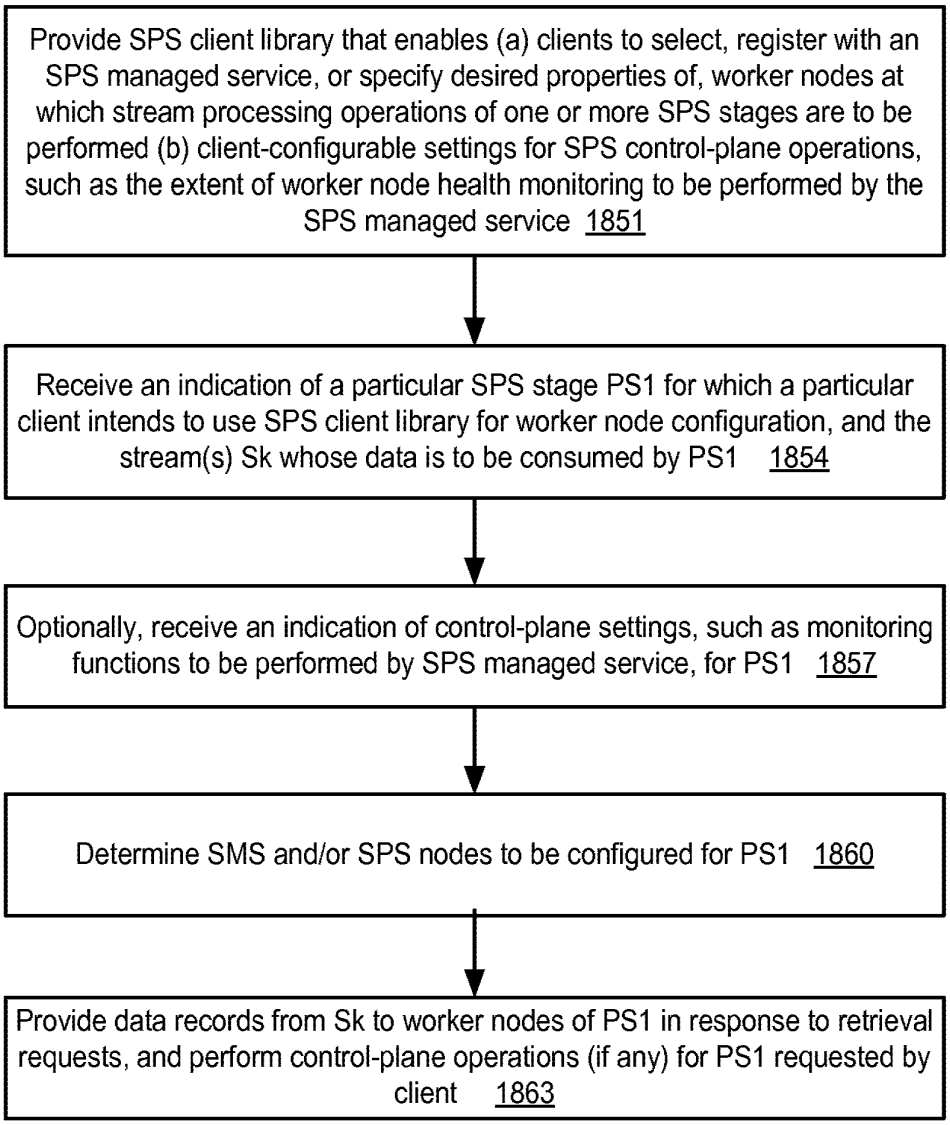

Provide SPS client library that enables (a) clients to select, register with an SPS managed service, or specify desired properties of, worker nodes at which stream processing operations of one or more SPS stages are to be performed (b) client-configurable settings for SPS control-plane operations, such as the extent of worker node health monitoring to be performed by the SPS managed service 1851
Receive an indication of a particular SPS stage PS1 for which a particular client intends to use SPS client library for worker node configuration, and the stream(s) Sk whose data is to be consumed by PS1 1854
Optionally, receive an indication of control-plane settings, such as monitoring functions to be performed by SPS managed service, for PS1 1857
Determine SMS and/or SPS nodes to be configured for PS1 1860
Provide data records from Sk to worker nodes of PS1 in response to retrieval requests, and perform control-plane operations (if any) for PS1 requested by client 1863

FIG. 18b

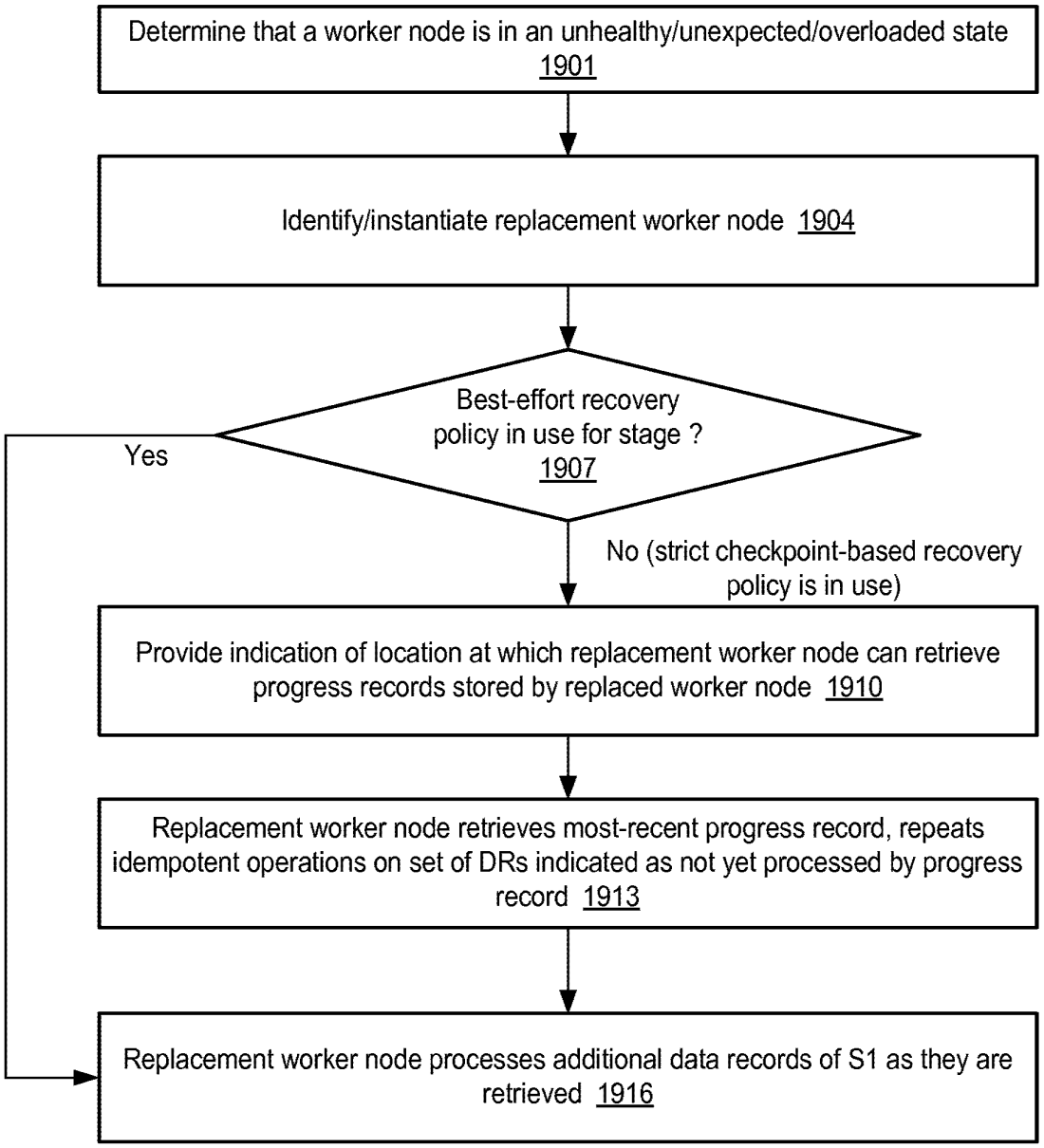

Determine that a worker node is in an unhealthy/unexpected/overloaded state 1901
Identify/instantiate replacement worker node 1904
Best-effort recovery policy in use for stage ? 1907
Yes
No (strict checkpoint-based recovery policy is in use)
Provide indication of location at which replacement worker node can retrieve progress records stored by replaced worker node 1910
Replacement worker node retrieves most-recent progress record, repeats idempotent operations on set of DRs indicated as not yet processed by progress record 1913
Replacement worker node processes additional data records of S1 as they are retrieved 1916

FIG. 19

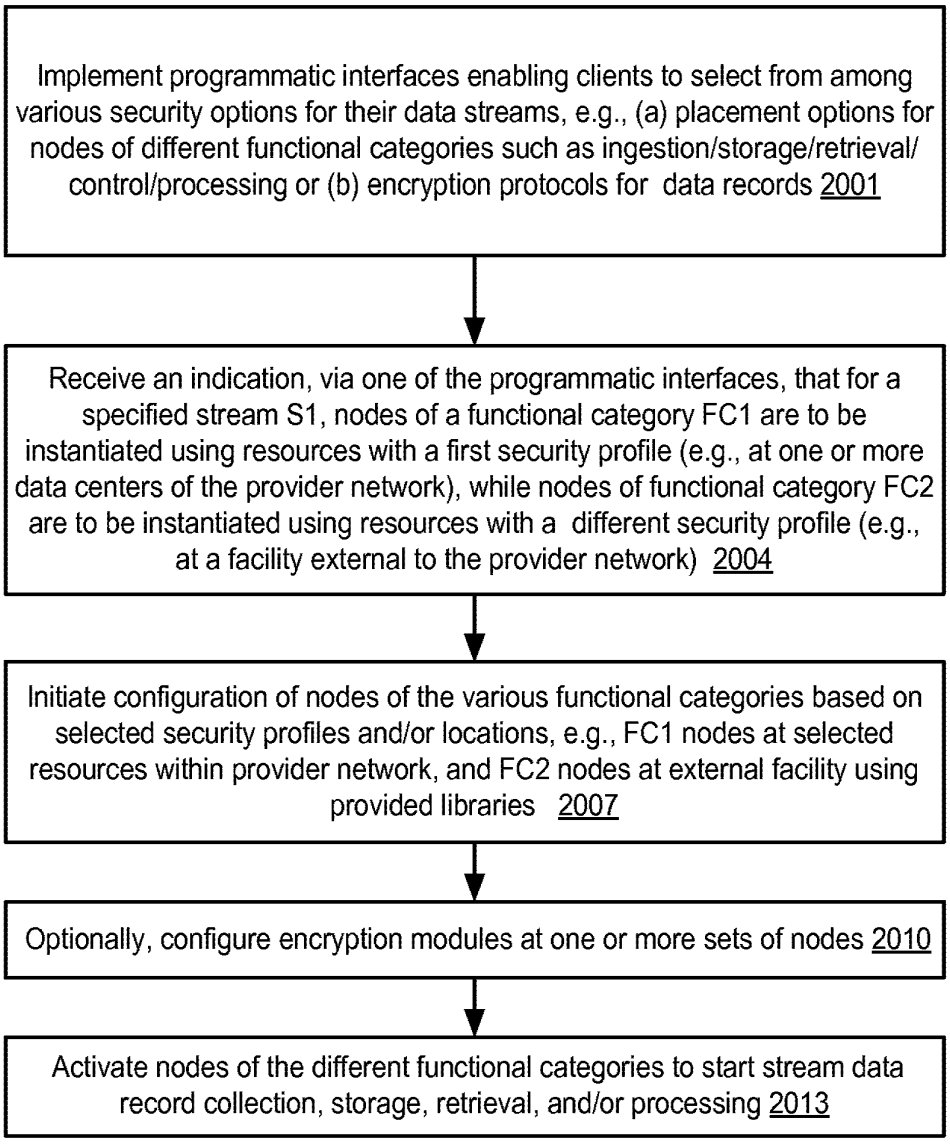

Implement programmatic interfaces enabling clients to select from among various security options for their data streams, e.g., (a) placement options for nodes of different functional categories such as ingestion/storage/retrieval/ control/processing or (b) encryption protocols for data records 2001
Receive an indication, via one of the programmatic interfaces, that for a specified stream S1, nodes of a functional category FC1 are to be instantiated using resources with a first security profile (e.g., at one or more data centers of the provider network), while nodes of functional category FC2 are to be instantiated using resources with a different security profile (e.g., at a facility external to the provider network) 2004
Initiate configuration of nodes of the various functional categories based on selected security profiles and/or locations, e.g., FC1 nodes at selected resources within provider network, and FC2 nodes at external facility using provided libraries 2007
Optionally, configure encryption modules at one or more sets of nodes 2010
Activate nodes of the different functional categories to start stream data record collection, storage, retrieval, and/or processing 2013

FIG. 20

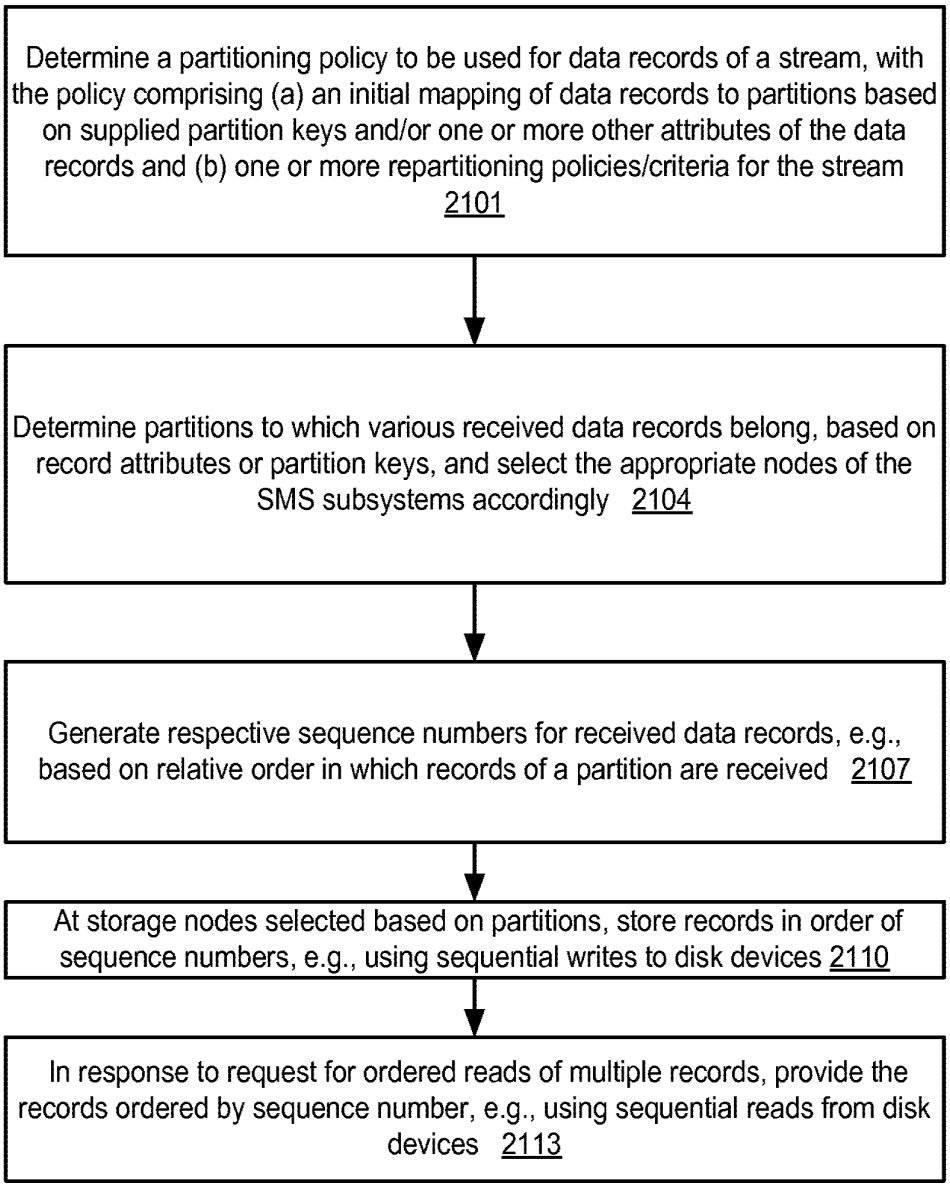

Determine a partitioning policy to be used for data records of a stream, with the policy comprising (a) an initial mapping of data records to partitions based on supplied partition keys and/or one or more other attributes of the data records and (b) one or more repartitioning policies/criteria for the stream 2101
Determine partitions to which various received data records belong, based on record attributes or partition keys, and select the appropriate nodes of the SMS subsystems accordingly 2104
Generate respective sequence numbers for received data records, e.g., based on relative order in which records of a partition are received 2107
At storage nodes selected based on partitions, store records in order of sequence numbers, e.g., using sequential writes to disk devices 2110
In response to request for ordered reads of multiple records, provide the records ordered by sequence number, e.g., using sequential reads from disk devices 2113

FIG. 21

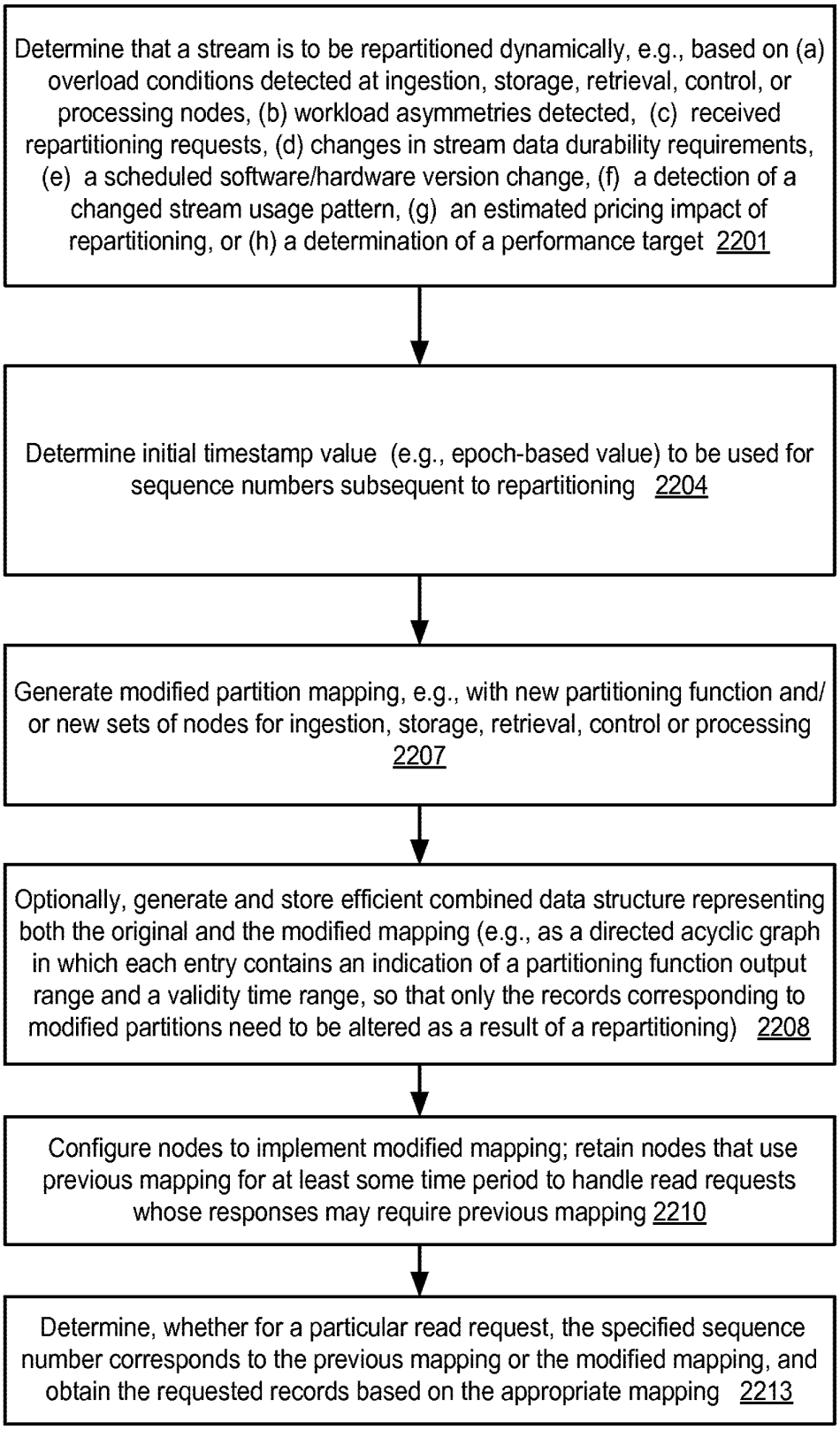

Determine that a stream is to be repartitioned dynamically, e.g., based on (a) overload conditions detected at ingestion, storage, retrieval, control, or processing nodes, (b) workload asymmetries detected, (c) received repartitioning requests, (d) changes in stream data durability requirements, (e) a scheduled software/hardware version change, (f) a detection of a changed stream usage pattern, (g) an estimated pricing impact of repartitioning, or (h) a determination of a performance target 2201
Determine initial timestamp value (e.g., epoch-based value) to be used for sequence numbers subsequent to repartitioning 2204
Generate modified partition mapping, e.g., with new partitioning function and/or new sets of nodes for ingestion, storage, retrieval, control or processing 2207
Optionally, generate and store efficient combined data structure representing both the original and the modified mapping (e.g., as a directed acyclic graph in which each entry contains an indication of a partitioning function output range and a validity time range, so that only the records corresponding to modified partitions need to be altered as a result of a repartitioning) 2208
Configure nodes to implement modified mapping; retain nodes that use previous mapping for at least some time period to handle read requests whose responses may require previous mapping 2210
Determine, whether for a particular read request, the specified sequence number corresponds to the previous mapping or the modified mapping, and obtain the requested records based on the appropriate mapping 2213

FIG. 22

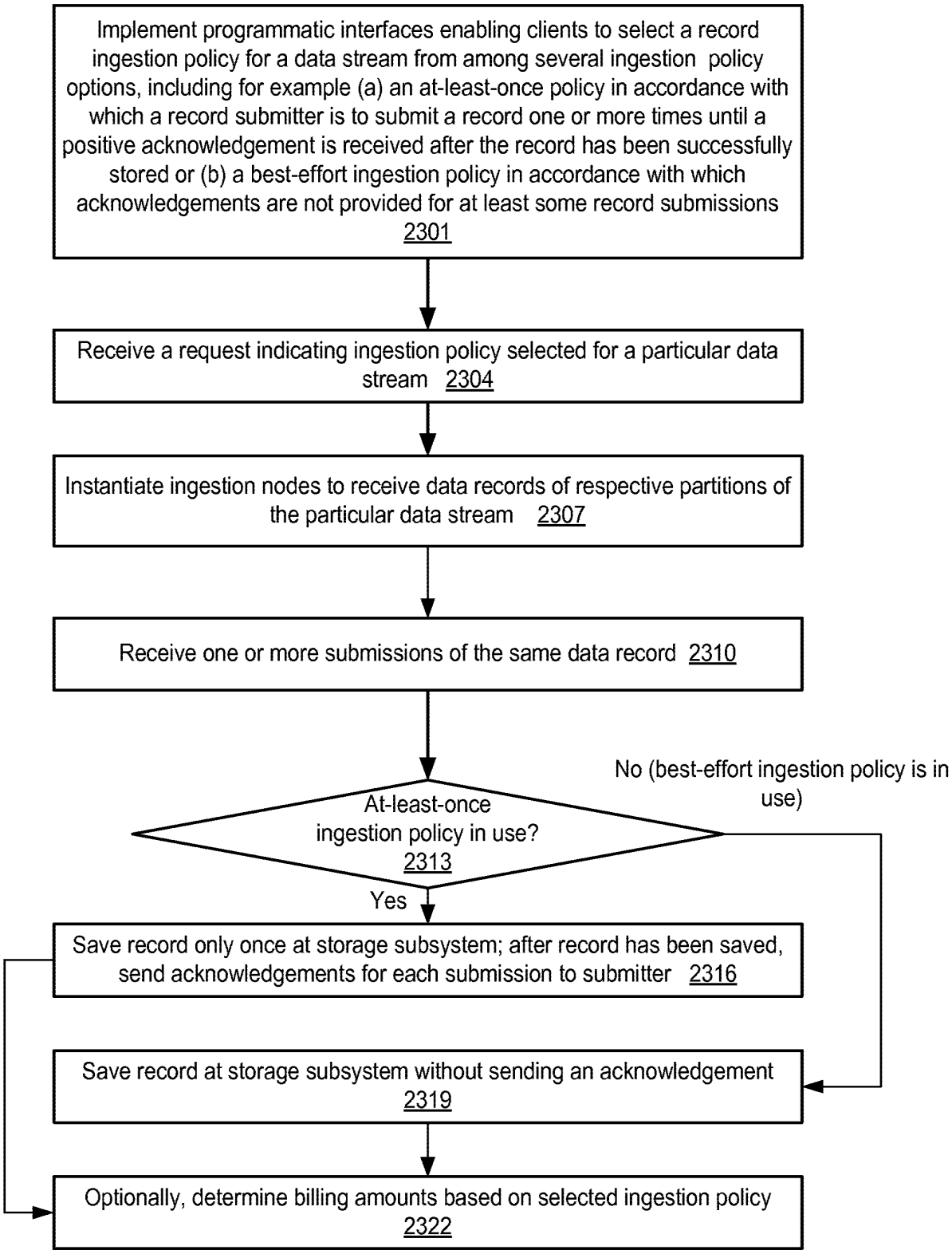

Implement programmatic interfaces enabling clients to select a record ingestion policy for a data stream from among several ingestion policy options, including for example (a) an at-least-once policy in accordance with which a record submitter is to submit a record one or more times until a positive acknowledgement is received after the record has been successfully stored or (b) a best-effort ingestion policy in accordance with which acknowledgements are not provided for at least some record submissions 2301
Receive a request indicating ingestion policy selected for a particular data stream 2304
Instantiate ingestion nodes to receive data records of respective partitions of the particular data stream 2307
Receive one or more submissions of the same data record 2310
At-least-once ingestion policy in use? 2313
No (best-effort ingestion policy is in use)
Yes
Save record only once at storage subsystem; after record has been saved, send acknowledgements for each submission to submitter 2316
Save record at storage subsystem without sending an acknowledgement 2319
Optionally, determine billing amounts based on selected ingestion policy 2322

FIG. 23

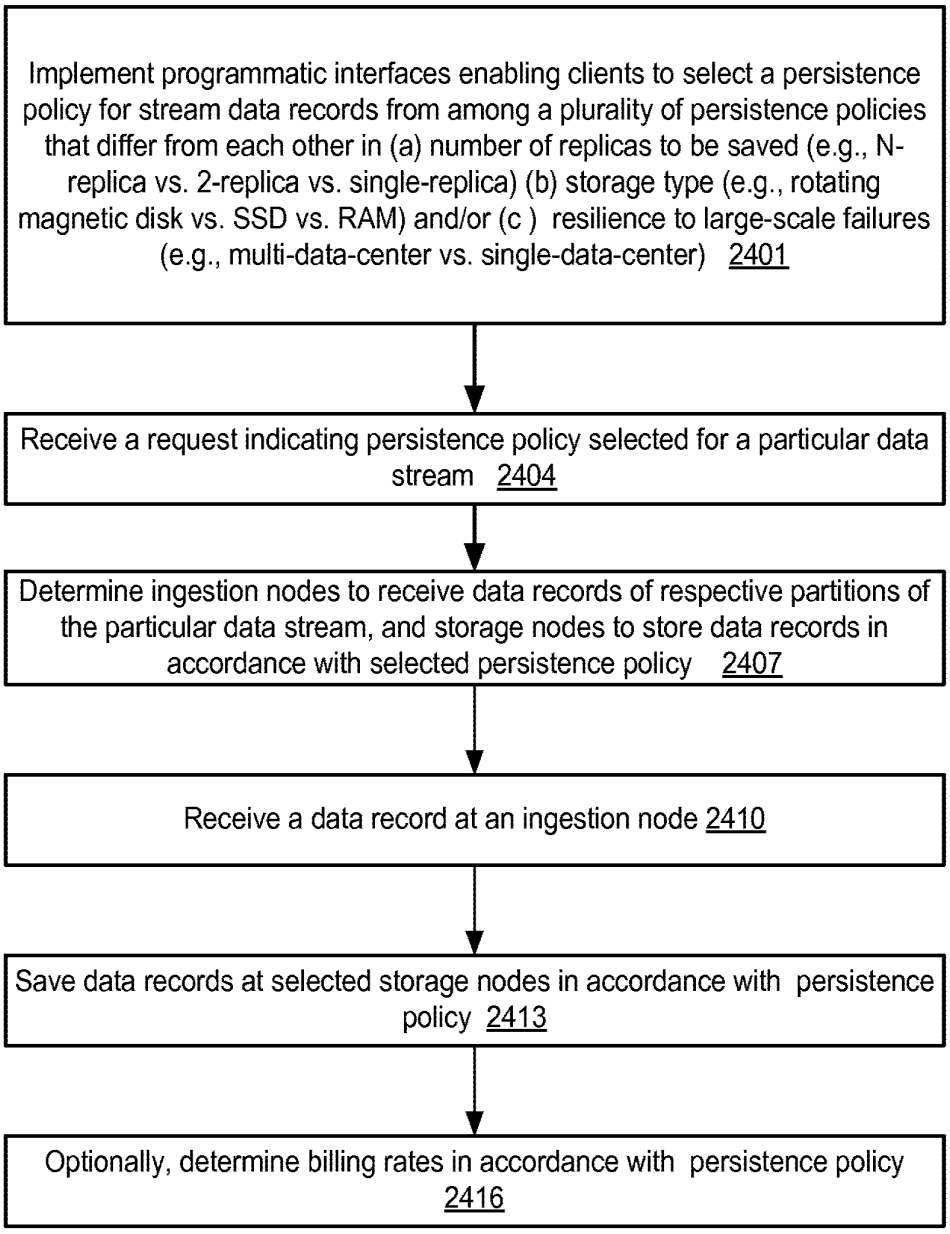

Implement programmatic interfaces enabling clients to select a persistence policy for stream data records from among a plurality of persistence policies that differ from each other in (a) number of replicas to be saved (e.g., N-replica vs. 2-replica vs. single-replica) (b) storage type (e.g., rotating magnetic disk vs. SSD vs. RAM) and/or (c) resilience to large-scale failures (e.g., multi-data-center vs. single-data-center) 2401
Receive a request indicating persistence policy selected for a particular data stream 2404
Determine ingestion nodes to receive data records of respective partitions of the particular data stream, and storage nodes to store data records in accordance with selected persistence policy 2407
Receive a data record at an ingestion node 2410
Save data records at selected storage nodes in accordance with persistence policy 2413
Optionally, determine billing rates in accordance with persistence policy 2416

FIG. 24

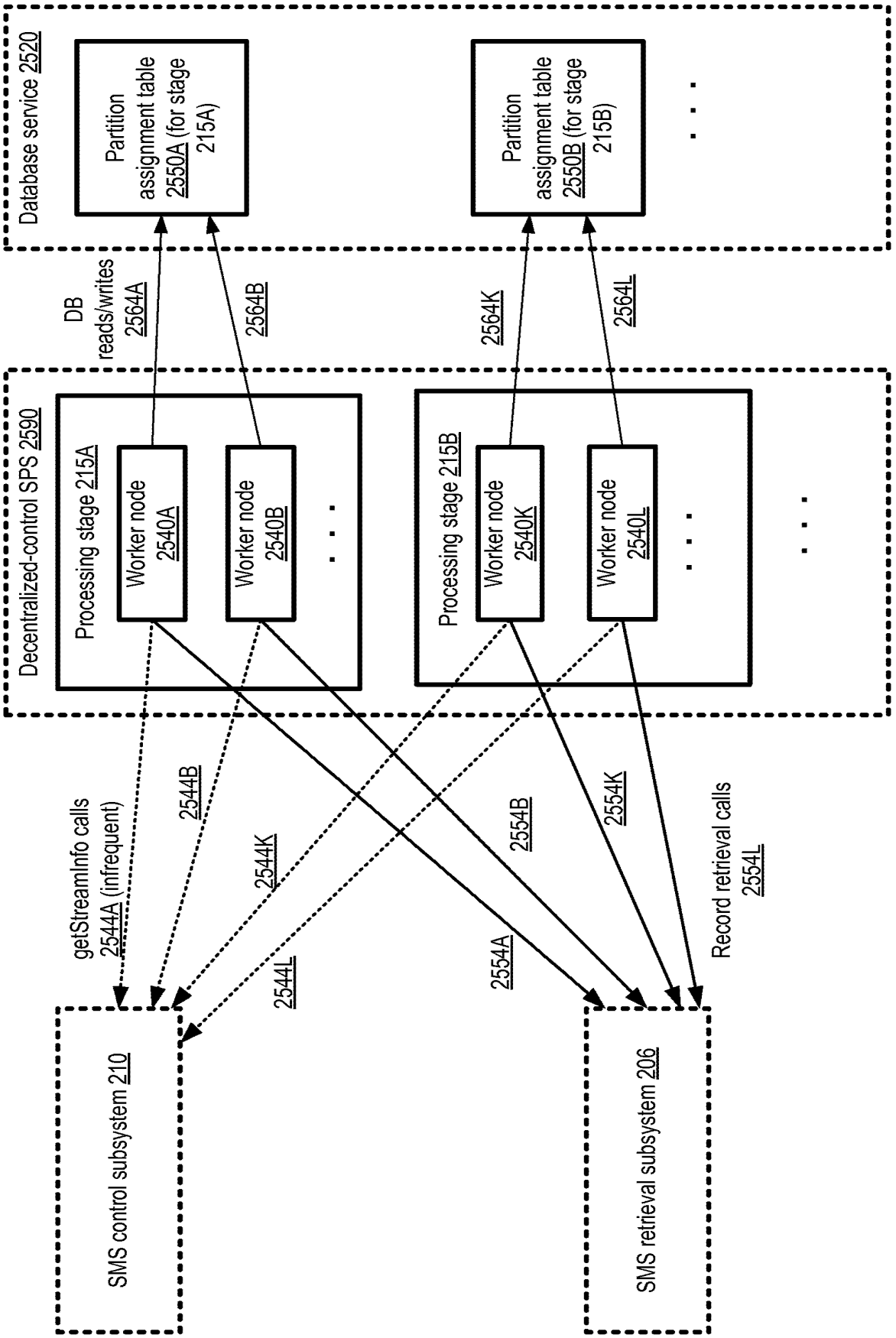

Database service 2520
Partition assignment table 2550A (for stage 215A)
Partition assignment table 2550B (for stage 215B)
DB reads/writes 2564A
2564B
2564K
2564L
Decentralized-control SPS 2590
Processing stage 215A
Worker node 2540A
Worker node 2540B
Processing stage 215B
Worker node 2540K
Worker node 2540L
getStreamInfo calls 2544A (infrequent)
2544B
2544K
2544L
2554A
2554B
2554K
Record retrieval calls 2554L
SMS control subsystem 210
SMS retrieval subsystem 206

FIG. 25

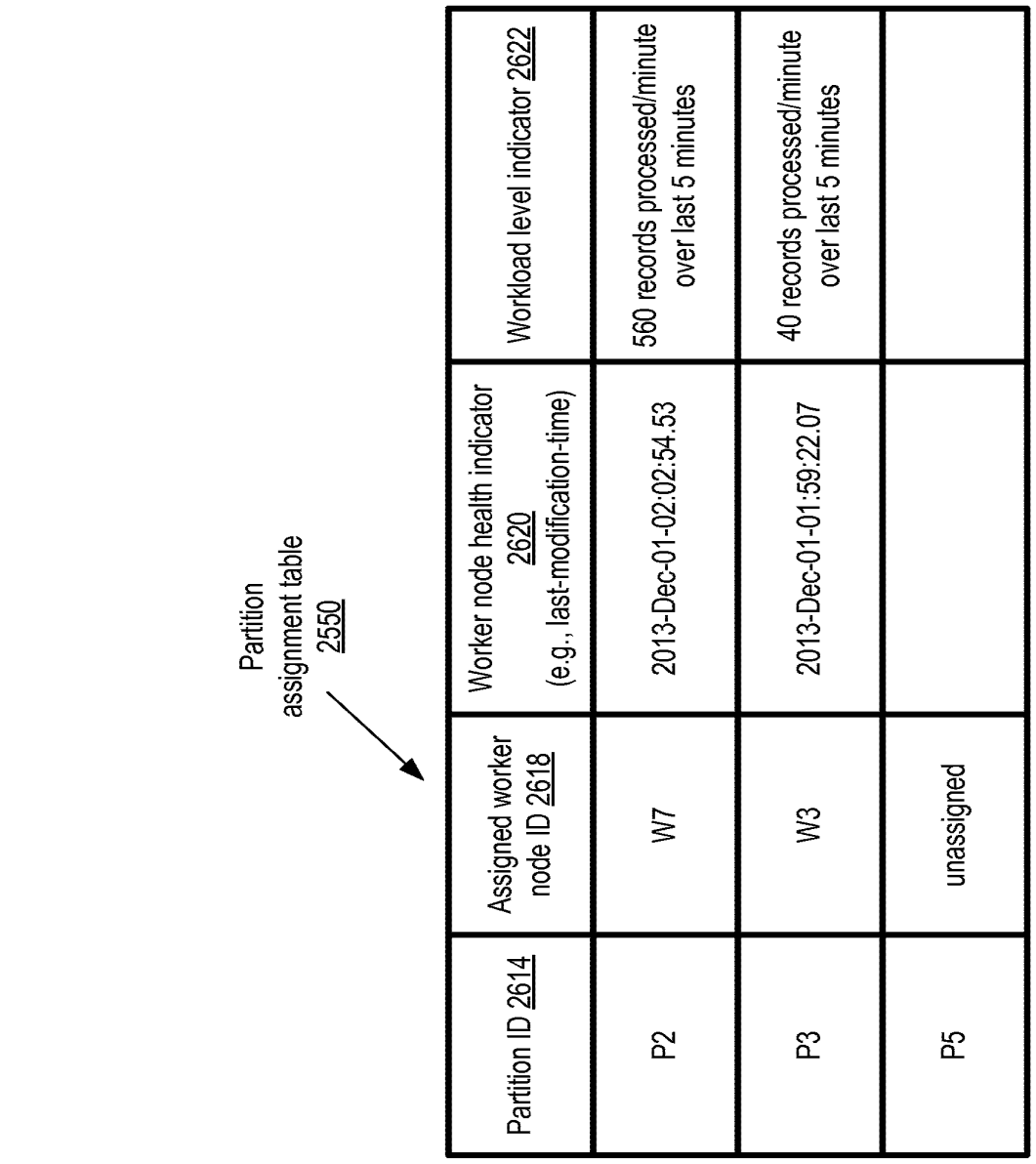

Partition assignment table 2550
Partition ID 2614
Assigned worker node ID 2618
Worker node health indicator 2620 (e.g., last-modification-time)
Workload level indicator 2622
P2
W7
2013-Dec-01-02:02:54.53
560 records processed/minute over last 5 minutes
P3
W3
2013-Dec-01-01:59:22.07
40 records processed/minute over last 5 minutes
P5
unassigned
Partition list at SMS control subsystem 2650
P1 (closed due to repartitioning)
P2 (active)
P3 (active)
P4 (closed due to repartitioning)
P5 (active)

FIG. 26

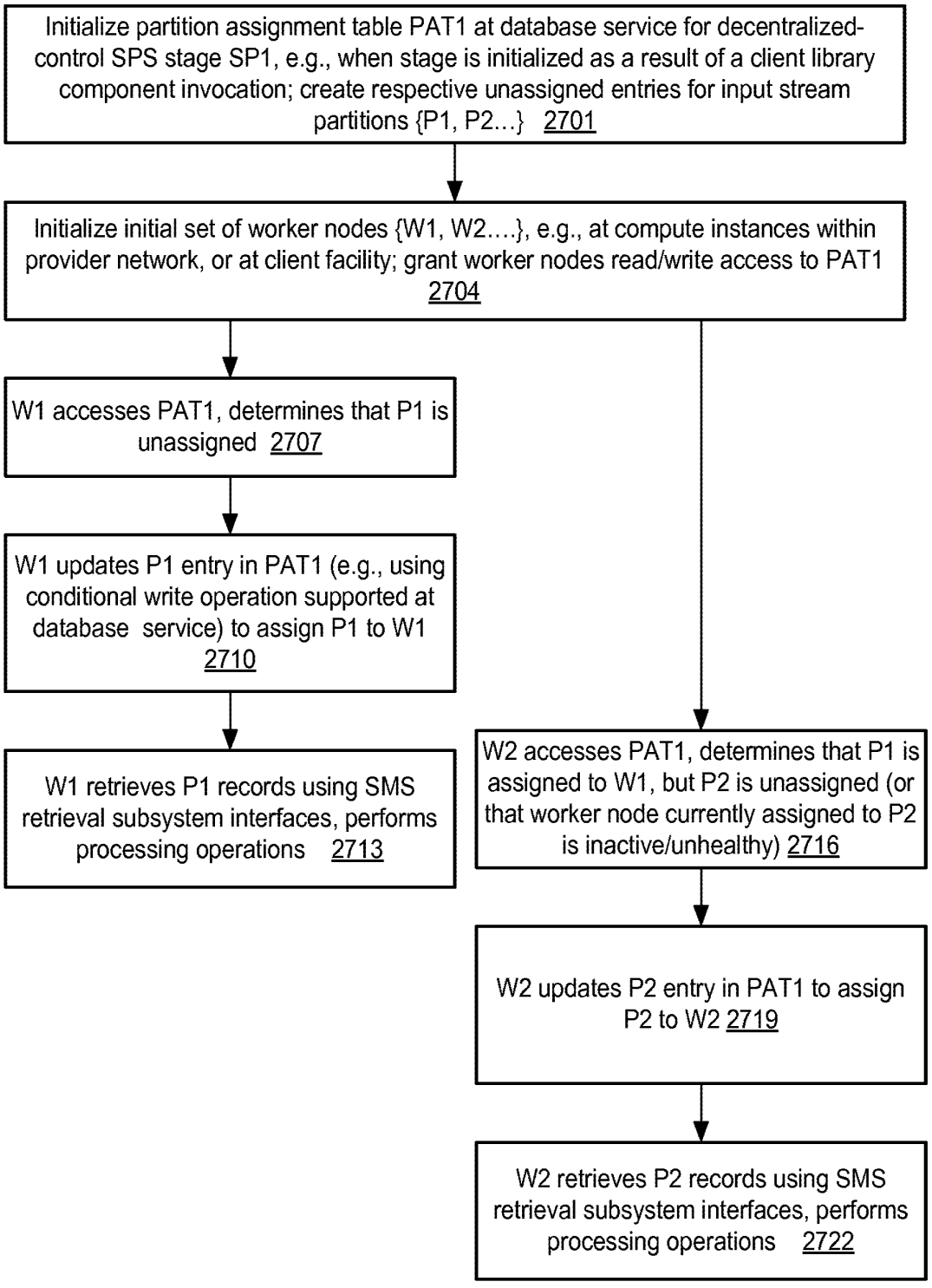

Initialize partition assignment table PAT1 at database service for decentralized-control SPS stage SP1, e.g., when stage is initialized as a result of a client library component invocation; create respective unassigned entries for input stream partitions {P1, P2...} 2701
Initialize initial set of worker nodes {W1, W2....}, e.g., at compute instances within provider network, or at client facility; grant worker nodes read/write access to PAT1 2704
W1 accesses PAT1, determines that P1 is unassigned 2707
W1 updates P1 entry in PAT1 (e.g., using conditional write operation supported at database service) to assign P1 to W1 2710
W1 retrieves P1 records using SMS retrieval subsystem interfaces, performs processing operations 2713
W2 accesses PAT1, determines that P1 is assigned to W1, but P2 is unassigned (or that worker node currently assigned to P2 is inactive/unhealthy) 2716
W2 updates P2 entry in PAT1 to assign P2 to W2 2719
W2 retrieves P2 records using SMS retrieval subsystem interfaces, performs processing operations 2722

FIG. 27

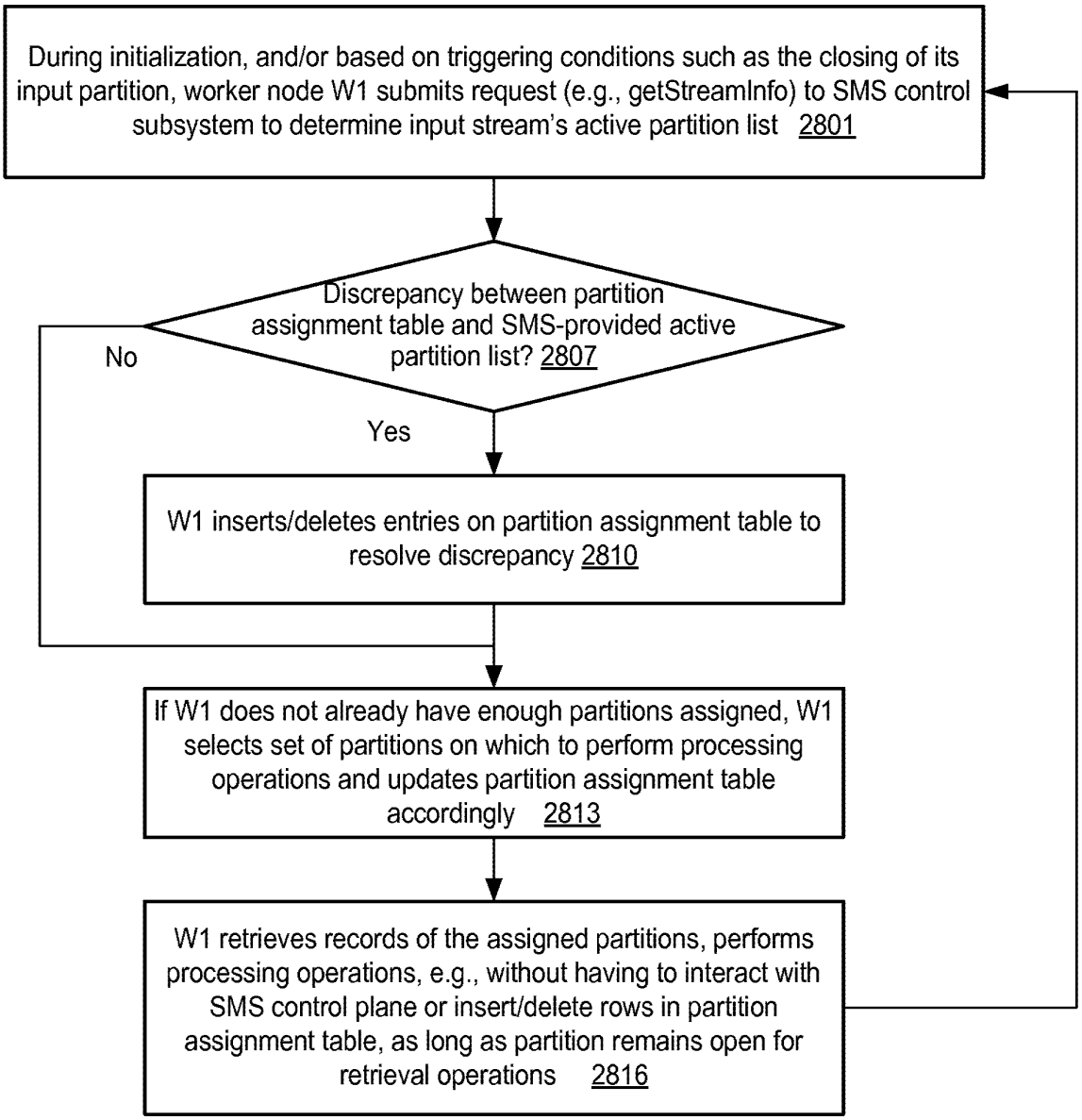

During initialization, and/or based on triggering conditions such as the closing of its input partition, worker node W1 submits request (e.g., getStreamInfo) to SMS control subsystem to determine input stream's active partition list 2801
Discrepancy between partition assignment table and SMS-provided active partition list? 2807
No
Yes
W1 inserts/deletes entries on partition assignment table to resolve discrepancy 2810
If W1 does not already have enough partitions assigned, W1 selects set of partitions on which to perform processing operations and updates partition assignment table accordingly 2813
W1 retrieves records of the assigned partitions, performs processing operations, e.g., without having to interact with SMS control plane or insert/delete rows in partition assignment table, as long as partition remains open for retrieval operations 2816

FIG. 28

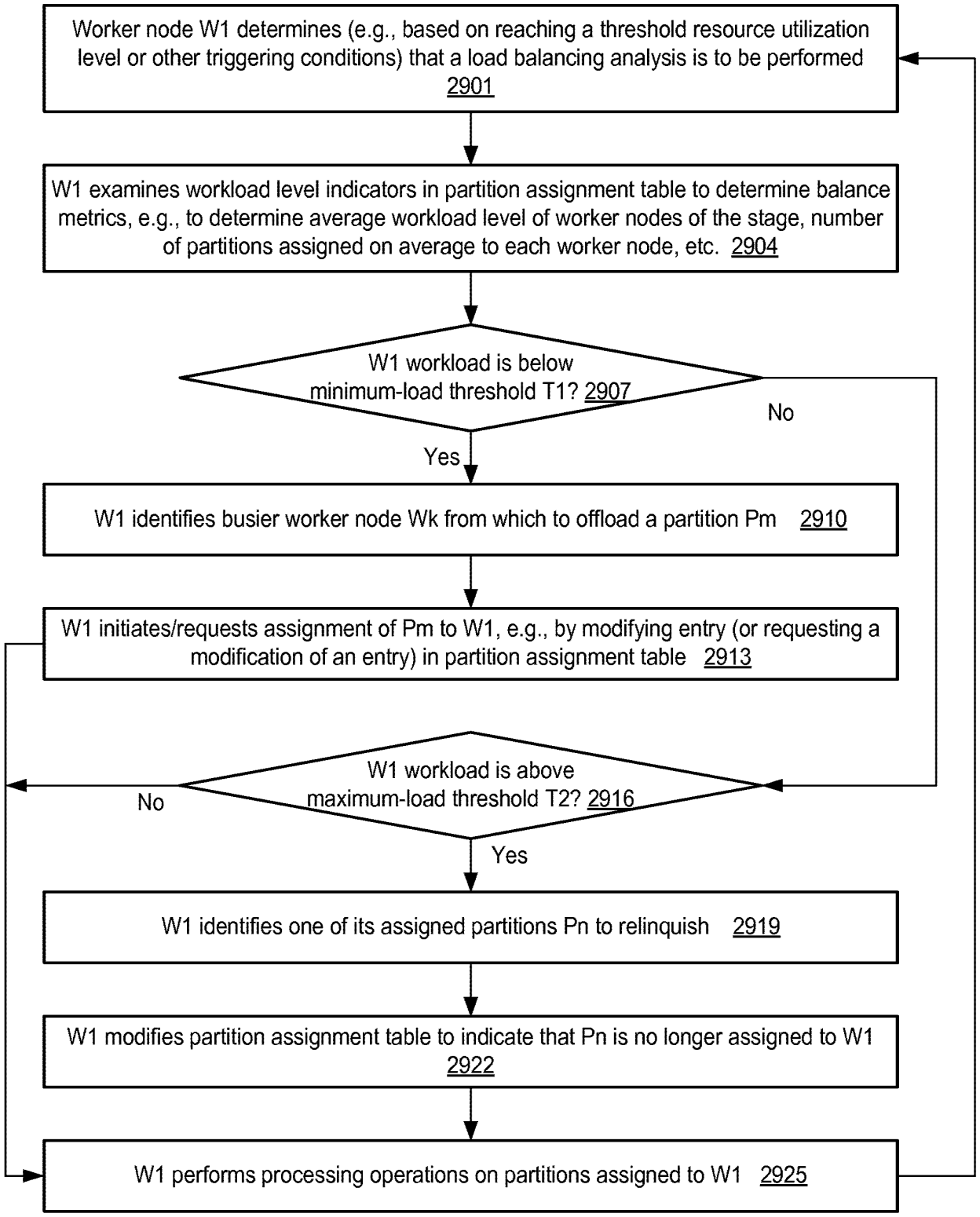

Worker node W1 determines (e.g., based on reaching a threshold resource utilization level or other triggering conditions) that a load balancing analysis is to be performed 2901
W1 examines workload level indicators in partition assignment table to determine balance metrics, e.g., to determine average workload level of worker nodes of the stage, number of partitions assigned on average to each worker node, etc. 2904
W1 workload is below minimum-load threshold T1? 2907
No
Yes
W1 identifies busier worker node Wk from which to offload a partition Pm 2910
W1 initiates/requests assignment of Pm to W1, e.g., by modifying entry (or requesting a modification of an entry) in partition assignment table 2913
W1 workload is above maximum-load threshold T2? 2916
No
Yes
W1 identifies one of its assigned partitions Pn to relinquish 2919
W1 modifies partition assignment table to indicate that Pn is no longer assigned to W1 2922
W1 performs processing operations on partitions assigned to W1 2925

FIG. 29

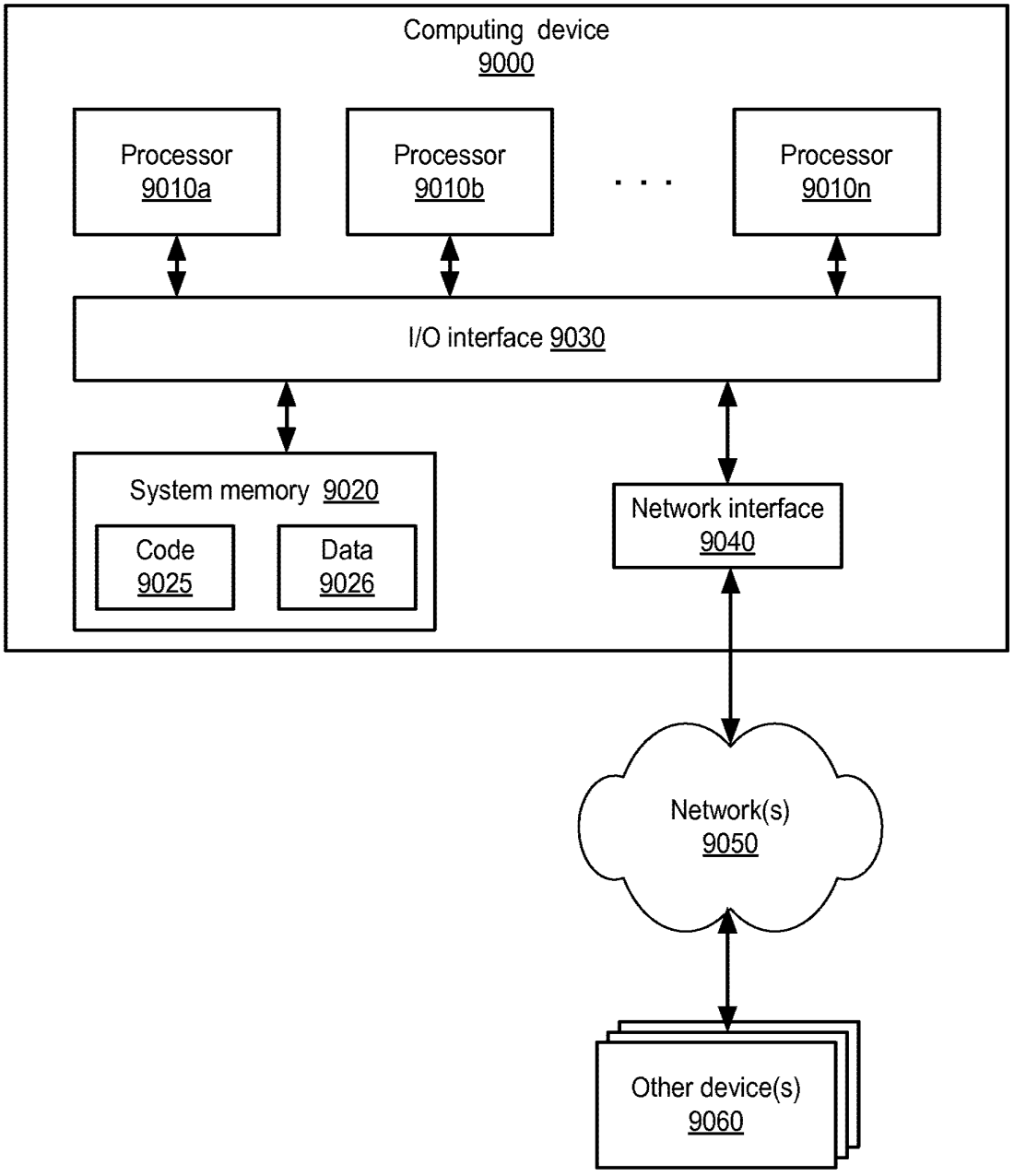
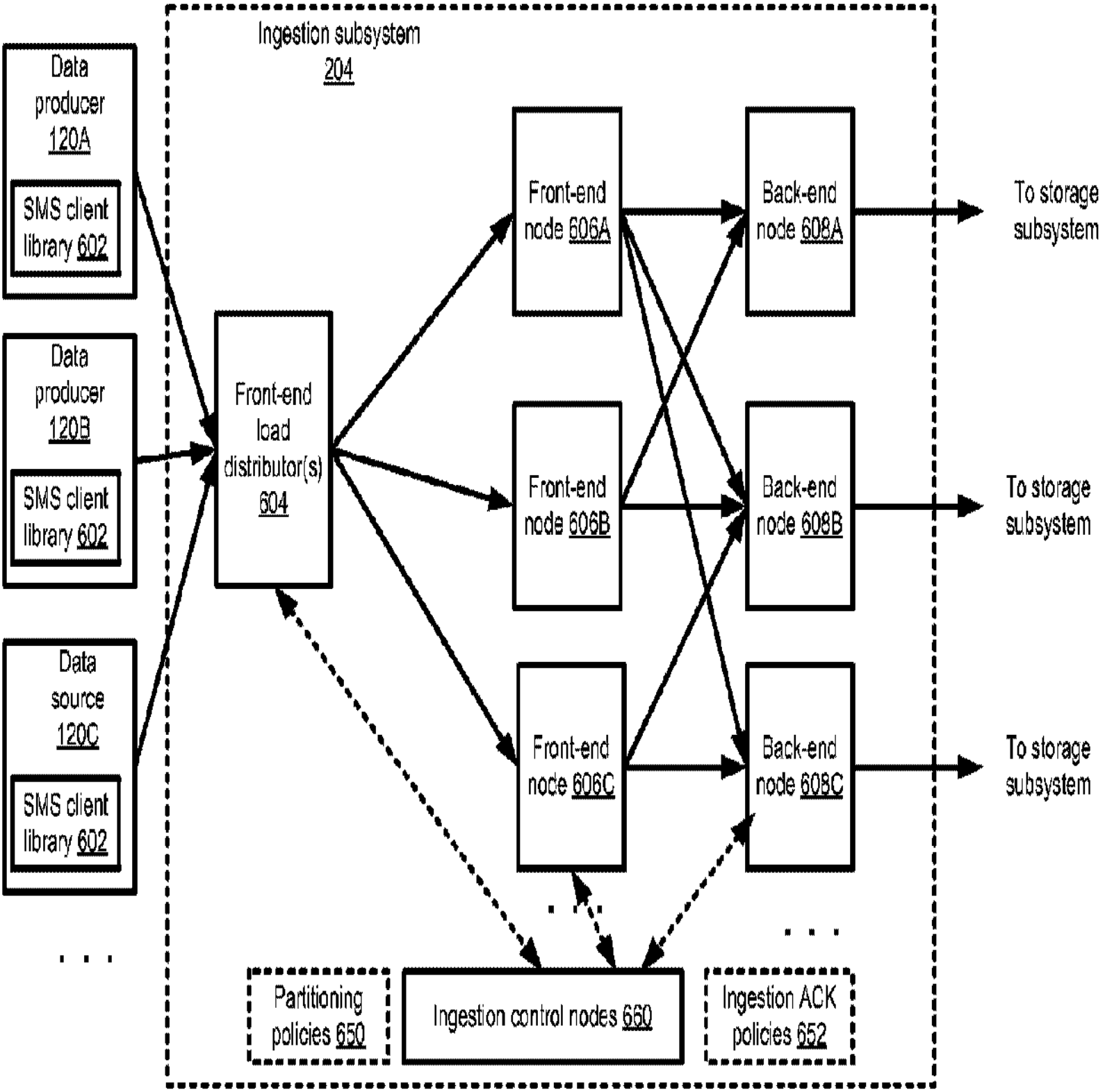

Computing device
9000
Processor
9010a
Processor
9010b
Processor
9010n
I/O interface 9030
System memory 9020
Code
9025
Data
9026
Network interface
9040
Network(s)
9050
Other device(s)
9060

FIG. 30


Ingestion subsystem
204
Data producer 120A
SMS client library 602
Data producer 120B
SMS client library 602
Data source 120C
SMS client library 602
Front-end load distributor(s) 604
Front-end node 606A
Front-end node 606B
Front-end node 606C
Back-end node 608A
Back-end node 608B
Back-end node 608C
To storage subsystem
To storage subsystem
To storage subsystem
Partitioning policies 650
Ingestion control nodes 660
Ingestion ACK policies 652

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
2 October 2014 (02.10.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/159150 A1

(51) International Patent Classification:
G06F 9/44 (2006.01)

(21) International Application Number:
PCT/US2014/022232

(22) International Filing Date:
10 March 2014 (10.03.2014)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
13/828,842 14 March 2013 (14.03.2013) US

(71) Applicant: **MICROSOFT CORPORATION** [US/US]; One Microsoft Way, Redmond, WA 98052-6399 (US).

(72) Inventors: **JUBRAN, Marwan, E.**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, WA 98052-6399 (US). **GERSHAFT, Aleksandr**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, WA 98052-6399 (US). **LIBENSON, Maksim**; c/o Microsoft Corporation, LCA - International Patents, One Microsoft Way, Redmond, WA 98052-6399 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

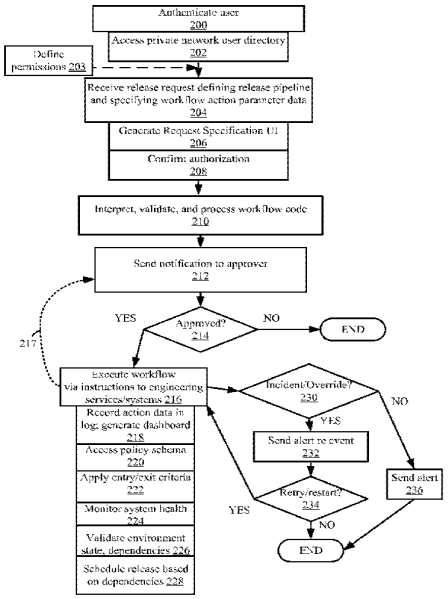
— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

[Continued on next page]

(54) **Title**: SOFTWARE RELEASE WORKFLOW MANAGEMENT

FIG. 3



(57) **Abstract**: A computer-implemented method for managing a release of a software product includes obtaining a request for the release, the request including workflow action parameter data to define a release pipeline involving a plurality of software engineering systems configured to process data indicative of the software product, and executing, with a processor, a workflow to implement the release pipeline in accordance with the workflow action parameter data. Executing the workflow includes sending a series of instructions to the plurality of software engineering systems. A successive instruction in the series of instructions is sent based on whether a gating rule for the release is met.

WO 2014/159150 A1

Published:

— *with international search report (Art. 21(3))*

SOFTWARE RELEASE WORKFLOW MANAGEMENT

BACKGROUND

Brief Description of Related Technology

[0001] Software development typically involves a build process that compiles source code files into binary packages for release to a production or delivery environment. Some software development projects may involve thousands, or even hundreds of thousands, of source code files having a complex dependency structure. A change in one source code file may thus cause undesirable conditions or unexpected results and failures for a large number of other source code files.

[0002] A software release typically includes one or more validation or testing procedures during which the results of a software build are tested. Software releases of complex products also usually involve multiple environments in which such testing occurs. For example, after a development team indicates that the source code development has reached a certain milestone, the software product is often sent to an integration environment in which a given component is tested against dependent systems or multiple development branches are tested. In some cases, after the integration environment, the software product may be released to a staging environment for further validation before eventually moving to a production environment.

SUMMARY

[0003] Methods, systems, and computer program products are directed to managing a software release workflow. One or more gating rules may be used to automate the management of the software release workflow.

[0004] In accordance with one aspect of the disclosure, successive instructions to a plurality of software engineering systems are implemented in a release pipeline for a software product based on whether a gating rule for the software product is met.

[0005] This Summary is provided to introduce a selection of concepts in a simplified form that are further described below in the Detailed Description. This Summary is not intended to identify key features or essential features of the claimed subject matter, nor is it intended to be used to limit the scope of the claimed subject matter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0006] For a more complete understanding of the disclosure, reference is made to the following detailed description and accompanying drawing figures, in which like reference numerals may be used to identify like elements in the figures.

[0007] Figure 1 is a block diagram of an exemplary system configured for software release workflow management in accordance with one embodiment.

[0008] Figure 2 is a block diagram of a release pipeline orchestrator of the system of Figure 1 in accordance with one embodiment.

[0009] Figure 3 is a flow diagram of an exemplary computer-implemented method for software release workflow management in accordance with one embodiment.

[0010] Figure 4 is a block diagram of a computing environment in accordance with one embodiment for implementation of the disclosed methods and systems or one or more components or aspects thereof.

[0011] While the disclosed systems and methods are susceptible of embodiments in various forms, specific embodiments are illustrated in the drawing (and are hereafter described), with the understanding that the disclosure is intended to be illustrative, and is not intended to limit the invention to the specific embodiments described and illustrated herein.

DETAILED DESCRIPTION

[0012] Methods, systems, and computer program products are described for managing a release of a software product. The management of the release is supported via workflows directing the operations of multiple software engineering systems involved in the release. The workflows of the disclosed embodiments establish a release pipeline that integrates the services of the software engineering systems. Implementation of the release pipeline is automated by the disclosed embodiments despite the often complex nature of the services provided by the software engineering systems, which may involve internal workflows of their own.

[0013] Interaction with the software engineering systems and execution of the workflow is automated via the application of a number of policy or gating rules. The gating rules may provide decision logic and other customizations to support an efficient implementation of the workflow. For example, execution of the workflow may include automated checks for various compliance parameters, such as whether the correct individuals have approved the release or a specific step of the workflow. Other automated checks may involve policies or rules related to quality assurance. The workflow may thus include, direct, and otherwise control different aspects or phases of the release lifecycle, such as a build process and a validation process. The disclosed embodiments may also automate the definition of a release pipeline via a front end or user interface and/or other

interfaces (e.g., an application programming interface (API)) through which a release request is received or obtained to initiate a release.

[0014] The disclosed embodiments may be employed to improve the efficiency of the entire release process or cycle. The disclosed embodiments automate the release process beyond automation of the procedures implemented within a specific one of the software engineering systems (e.g., build automation). The automation of the entire release cycle provided by the disclosed embodiments may enable software developers to release software products more quickly and more often without sacrificing other interests, such as maintaining compliance and security. For example, with the disclosed embodiments, release transactions and other actions need not be tracked or implemented via emails, spreadsheets, or other inefficient, manual processes. By removing such manual steps in the release process, an agile, seamless, or continuous release experience may be provided via the disclosed embodiments. The efficiency improvements may also be facilitated through scaling (e.g., cloud-based scaling) and/or higher availability (e.g., protection against local or regional outages) of the release management services provided by the disclosed embodiments.

[0015] Notwithstanding references herein to various commercially available software engineering services managed via the disclosed embodiments, the disclosed embodiments are not limited to any particular type of engineering service or provider thereof. Although described in connection with cloud-based services, the disclosed embodiments are not limited to any specific operating system, environment, platform, or computing infrastructure. The disclosed embodiments may be used to manage the release of various types of software products.

[0016] The nature of the releases managed via the disclosed embodiments may vary. For example, a release need not involve a full or complete release of a software product to production. Instead, the release may involve a partial release, such as a patch, hotfix or quick fix update to the software product, or a configuration change. The final stage or destination of the release need not involve production or deployment. For example, the release may be an internal release between release environments within the manufacturer, such as an integration environment and a staging environment.

[0017] Figure 1 depicts a software development environment 100 for development of one or more software products. The software development environment 100 includes a release management system 102 or service configured to manage a software release workflow. The software development environment 100 includes a plurality of software engineering

systems used during the software development process. Various combinations of the software engineering systems may be applied to the software product in a release pipeline established by the release workflow. The software engineering systems may be discrete systems or integrated to any desired extent. For example, one or more of the software engineering systems may be provided as a separate software service. In this example, the software engineering systems include a source control system 104, a build system 106, a validation system 108, a project management system 110, a deployment infrastructure 112, and an alert system 114. Additional, fewer, or alternative systems may be provided. For example, the software engineering systems may include a content distribution service configured to distribute build outputs (e.g., binary files, configuration data, and other artifacts) to various distributions (e.g., worldwide) and to provide services, such as a build directory service, which may discover builds and use validations to ensure the integrity of content is retained upon delivery. Such content distribution services may be integrated with the build system 106, the deployment infrastructure 112, or be configured as standalone services.

[0018] Each of the software engineering systems is configured to process data indicative of the software product(s) under development. One or more of the software engineering systems may be provided as a service (e.g., in a client-server framework) or otherwise involve a distributed computing infrastructure. One or more of the software engineering systems may thus be provided or supported via a cloud or other networked computing arrangement. For example, the build system 106 may be configured in accordance with the teachings of U.S. Patent Publication No. 2013/0055253 entitled "Cloud-based Build Service", published on February 28, 2013 (Microsoft Attorney Docket No. 333025.01). The software engineering systems need not be provided separately as shown in Figure 1. The software engineering systems may be integrated to any desired extent. For instance, some or all of the software engineering systems may be coupled in one or more ways. In some examples, the coupling may involve a shared repository or other resource and/or involve interaction or communications, such as via an API. For example, one or more source control services, one or more project management services, and/or other services may be provided via a common server, such as a server configured in accordance with the Team Foundation Server platform from Microsoft Corporation, which may be provided as part of a Visual Studio® system, such as the Visual Studio® Application Lifecycle Management system.

[0019] The source control system 104 may be directed to version or revision control and other management of the source code for the software product. In one example, the source control system 104 utilizes the Perforce revision control system (Perforce Software, Inc.), but any source control system may be used. The source control system 104 may include one or more repositories for the source code data or files. The repositories may be local or distributed. For example, the repositories may be located and arranged in a cloud infrastructure. The source code may be written in any one or more languages, and the platform or operating system for the source code may also vary.

[0020] The build system 106 may be directed to automating the process of compiling the source code data into binary code, and packaging the binary code. The build system 106 may include a number of tools and repositories for processing and handling the source code and binary code files and data. For example, the build system 106 may include, configure, or access a file server in which the results of a build are stored. Any build software, tool, and/or infrastructure may be used in the build system 106. In one example, the build system 106 utilizes the MSBuild build platform (Microsoft Corporation), which may be available via or in conjunction with the Microsoft® Visual Studio® integrated development environment. Other build systems and/or integrated development environments may be used.

[0021] The validation system 108 may be directed to validating the operability of the binary code, which may include a set of binaries. The validation system 108 may be configured to execute a number of tests configured to determine whether the binary code works as intended. For example, one or more tests may be configured to determine whether the binary code meets a number of specifications for the software product. A variety of different types of testing may be provided, including, for instance, unit testing, functional testing, stress testing, fault injection, penetration testing, etc. Other types of validation processes may be implemented by the validation system 108, including, for example, compliance validation and security validation.

[0022] The project management system 110 may be directed to gathering, tracking, and/or storing data indicative of a record of the release process. For example, work item tracking, incident tracking, deployment tracking, and other item tracking may be implemented by the project management system 110 to support the execution of the release pipeline. The project management system 110 may be used to provide a signal or other data to the release management system 102 to indicate that a particular action is complete, complete with faults, etc. The project management system 110 may

alternatively or additionally be configured to provide logging functionality for the release management system 102. Such logging may be useful in a variety of ways. For example, the log data may include, for example, compliance data, such as the identity of a user who approved a step in the release process. The project management system 110 may include a repository, such as a persistent store, in which such log and other release process data is stored. The project management system 110 may include a manager (e.g., a database manager) and user interface for accessing, handling, or otherwise processing the release process data (e.g., generating business intelligence reports). The project management system 110 may include a Microsoft® SQL Server® or SQL Azure™ database management system from Microsoft Corporation, but other database management systems or architectures may be used. In some examples, the project management system 110 may include the above-referenced Team Foundation Server platform from Microsoft Corporation, but other services may be used.

[0023] The deployment infrastructure 112 may be directed to an automated deployment of the software product. The deployment infrastructure 112 may include a gateway or other service configured to support software product updates. For example, if the software product is configured for cloud-based execution, the deployment infrastructure 112 may include a service for distributing the software product to one or more data centers, virtual machines, etc. The software development environment 100 need not include the deployment infrastructure 112 in some cases, such as when the software product is not electronically distributed.

[0024] The alert system 114 may include an alert or monitoring service to support communications with users beyond any dedicated or other user interfaces of the release management system 102. The communications may include alerts and other messages to inform users of an event, such as a failure (e.g., a build failure), that occurs during implementation of a workflow. The messages may utilize a variety of communication protocols and media, including, for instance, emails and text messages, but other messaging frameworks may be used. The alert system 114 may be a cloud-based monitoring service, such as the Windows Azure™ Alert and Incident Monitoring Service (AIMS) from Microsoft Corporation, but non-cloud monitoring services may be used.

[0025] In some embodiments, one or more functions of the alert system 114 may be integrated with the release management system 102. For example, monitoring and/or evaluation of the release process (e.g., to detect events warranting an alert) may be implemented by a module of the release management system 102. The embodiment of

Figure 2 provides some examples of such monitoring. The alert system 114 may then be instructed to transmit an email or other message regarding the event detected.

[0026] In the embodiment of Figure 1, the release management system 102 includes an orchestrator 116 (or orchestrator server), a policy engine 118, and a front end system 120 (or front end server). The orchestrator 116 may be configured to direct or otherwise communicate or interact with a combination of the above-referenced software engineering systems to establish a release pipeline. The orchestrator 116 may support or facilitate communications between the software engineering services. The policy engine 118 may be configured to apply or enforce gating rules or other policies or decision logic during the release process. The front end system 120 may be configured to support user configuration of the release management system 102, monitoring of the release process, and other user interaction with the release management system 102. The components of the release management system 102 may be implemented on separate computers (e.g., server computers) or computing systems (e.g., sets of server computers), or on a common computer or computing system. Each such computer or computing system may be configured as a virtual machine.

[0027] One or more of the components of the release management system 102, e.g., the orchestrator 116, may be configured to manage multiple release pipelines for one or more software products. As described herein, the release management system 102 may be configured to provide the release management services in a parallel processing architecture or arrangement. For example, the release management system 102 may include multiple instances of the orchestrator 116 to orchestrate multiple release pipelines for one or more software products. Similarly, the release management system 102 may include multiple instances of the policy engine 118 and/or the front end system 120. The multiple instances may be arranged in a cloud-based service framework that allows the number of instances to be scaled in accordance with demand. The components of the release management system 102 may be integrated to any desired extent. For example, the policy engine 118 may be an integral part of the orchestrator 116. In some cases, various combinations of the services provided by the orchestrator 116, the policy engine 118, and/or the front end system 120 are implemented on a common computer (e.g., server computer) or computing system.

[0028] The orchestrator 116 may include a workflow engine 122 to implement workflows involving the above-described software engineering services. The workflow engine 122 may correspond with the workflow manager provided as a part of the

Microsoft® Windows® Workflow Foundation framework from Microsoft Corporation, although other workflow managers or engines may be used. As described below, the workflow engine 122 and/or the orchestrator 116 are configured to manage communications with the software engineering services, process workflow parameter data to provide a workflow to implement a release pipeline configured in accordance with such workflow parameter data, and execute the workflow to send instructions to the software engineering systems. For example, the parameter data may specify that the release pipeline be configured to include an integration step, a build step, a deployment step, and a code movement step. In another example, the parameter data may specify that the release pipeline be configured to include a code movement step, a build step, a validation step (e.g., for unit testing), a deployment to integration step, a further validation step (e.g., for functional, stress, and penetration testing for validation of a service), a further code movement step, and a deploy to production step. The steps and step order in the release pipeline may vary considerably. For example, other release pipelines may not have code movement steps due to the absence of a branching architecture. Some steps need not involve processing of code for the software product. For example, other pipelines may include instructions for gathering and displaying metrics regarding the release, as described below.

[0029] Data indicative of and/or resulting from the workflows, such as temporary or log data, may be stored in a workflow data store 124. The temporary data may be indicative of the workflow execution, which may be useful to support recovery in the event of a failure. The workflow data store 124 may include one or more local data stores or a distributed data store in communication with one or more processors used to implement the workflow engine 122.

[0030] Implementation of the release workflow may be initiated via a request received or otherwise obtained via the front end system 120. The request may specify or otherwise include workflow action parameters or parameter data to define a release pipeline involving a combination of the above-described software engineering systems. For example, the request may specify a software product, one or more source control branches, file directories or other locations, one or more change lists, one or more release environments (e.g., a target environment), and a combination of actions (e.g., build, validate, deploy, validate, etc.) to be implemented in the release. In the embodiment of Figure 1, the front end system 120 may be configured to generate a user interface 126 to support the entry of the request and the workflow action parameter data. A workflow

action parameter data package 128 may be passed from the front end system 120 to the orchestrator 116 and/or the workflow engine 122. The workflow action parameter data package 128 may include parameters or data indicative of or specifying one or more aspects of the release pipeline or the workflow. The parameters or data may include data in addition to that provided via the request or the user interface 126. The workflow action parameter data package 128 may be configured as a release specification. In some cases, the workflow action parameter data package 128 is received by the orchestrator 116 via an API end point of the orchestrator 116. Other communication schemes may be used. For example, the workflow action parameter data package 128 may be derived from a user interface input process that provides a user an opportunity to reuse or otherwise select a previously used workflow, workflow action, or other workflow parameter or data.

[0031] The release pipeline and the workflow may be defined by the orchestrator 116 and/or the workflow engine 122 in accordance with the workflow parameter data in the workflow action parameter data package 128. For example, the orchestrator 116 and/or the workflow engine 122 may incorporate the workflow parameter data into a workflow template. The orchestrator 116 and/or the workflow engine 122 may be configured to select a template from a plurality of templates stored in, e.g., the workflow data store 124. The selection of the template may be based on one or more aspects of the workflow parameter data, such as the combination of actions requested. In one embodiment, a template is provided for each respective release environment (e.g., a test environment, a production environment, etc.). The front end system 120 may be configured to facilitate modifications to the workflow templates. In some embodiments, an API may be provided to facilitate an upload of one or more workflows for selection. Alternatively or additionally, the workflows may be modified during execution thereof.

[0032] During the implementation of the release workflow, a record of the workflow actions or steps executed may be stored by the release management system 102 in a persistent store 130 in communication with the orchestrator 116 and/or the release management system 102. The persistent store 130 may be provided by a cloud-based storage service. The persistent store 130 may be configured for durable storage of a variety of different types of data. For example, data indicative of each workflow action implemented during the workflow, as well as the results of each action, may be recorded or stored in the persistent store 130. For instance, the binary files resulting from a build may be stored in the persistent store 130. The workflow may be configured with one or more checkpoints at which data indicative of the state of the release is stored in the

persistent store 130. If a failure occurs after the checkpoint, the data stored at that time may be relied upon to restart the workflow from the checkpoint. The data stored at a checkpoint may also be used to support the execution of multiple instances of the release pipeline. The data may also be used for scalability. For example, a workflow started on a first server may be resumed by another server, if, for instance, the first server becomes overloaded or is taken out of service (e.g., through maintenance or by failure).

[0033] Other types of result data may be stored by the release management system 102 in the persistent store 130. For example, result data identifying storage locations for the build results may be recorded in the persistent store 130. In another example, the result data may be indicative of the status or other operational characteristic of the workflow or release. The operational characteristic(s) to be recorded or gathered may be identified in the request received via the front end system 120. The persistent store 130 may provide storage redundancy for one or more types of data stored in the workflow data store 124.

[0034] Metric data 132 indicative of the release workflow(s) may be returned from the release management system 102 or the workflow engine 122 to the front end system 120. The front end system 120 may, in turn, pass the metric data 132 to the persistent store 130. The metric data 132 may alternatively or additionally be stored directly in the persistent store 130 or through a different intermediary device or system, such as the above-referenced monitoring system or other monitoring system or service. The metric data 132 may be alternatively or additionally provided to the user via a dashboard 134 generated by the front end system 120. The metric data 132 may be indicative of the progress, status, or other operational characteristic of the release workflows being managed by the release management service 102. For example, the metric data 132 may be representative of how long a particular step of a release took to execute, of how many releases are currently being executed, or of the results of one or more tests at varying levels of specificity (e.g., either individual or aggregated test results). One or more characteristics of the metric data 132 may be specified via the user interface 126. For instance, the user requesting a release may specify the types of metric data 132 to be displayed via the dashboard 134. The front end system 120 may be configured for telemetric communications with the orchestrator 116 to support the gathering of the metric data 132.

[0035] A request for the metric data 132 may be provided or received at various times. The metric data 132 may be requested along with the request to initiate a release workflow. Thus, in some embodiments, the workflow action parameter data package 128 includes data indicative of the types of metric data 132 requested. Alternatively or

additionally, a request for the metric data 132 may be received via the user interface 126 after the initiation of the workflow. The front end system 120 may then be configured to send a message to the orchestrator 116 requesting the metric data 132. The message may include instructions to gather the metric data 132. The request may be received by the orchestrator by the above-referenced API end point. The metric data 132 may alternatively or additionally be used to trigger automatic alerts about, e.g., system states. For example, an alert may be triggered in the event that the metric data 132 indicates that the duration of a certain process in the system exceeds a predetermined threshold. The data may also be aggregated and reported at a later time across multiple releases.

[0036] The persistent store 130 may be used by the release management system 102 as a long term data storage for some or all of the data stored in the workflow data store 124. The workflow data may include log data. The log data may include data recorded for compliance purposes, including, for instance, the identity of each person providing approval during the workflow. The release management system 102 may be configured in accordance with auditing data of, e.g., a retention or other compliance policy, to maintain the data in the persistent store 130 for a time period specified via the auditing data. Any number of stores may be used to provide capacity for the storage of data from a number of releases. For example, the persistent store 130 may include a varying number of data stores in a cloud-based framework or other distributed computing architecture. The data may also be recorded by or as part of the project management system 110, which may be useful in, for instance, correlating a release with work items, incidents, or other workflow events.

[0037] The front end system 120 may provide functionality beyond requesting releases and dashboard displays. In the embodiment of Figure 1, the front end system 120 includes an authentication module 136 and an authorization module 138. The authentication module 136 is configured to authenticate users of the release management system 102. The authentication module 136 may be configured to access one or more user directories (e.g., a corporate user directory) or other databases to confirm user credentials. The authorization module 138 may be configured to establish or determine user permissions or roles, and the permissions of such roles, which may then be associated with specific users. In some embodiments, each authenticated user may be assigned to a user role having a set of permissions to, e.g., approve certain workflow actions, as described further below. User or user role permissions may be specified via an administrator control interface 140 generated by the front end system 120. Alternatively or additionally, user permissions may

be specified indirectly, e.g., via a remote service. For example, the user permissions may be specified in accordance with a title (e.g., project manager) indicated via data made available by the database(s) relied upon for authentication. The user may also have a different role (e.g., approver, initiator, developer, etc.) based on the origin, scope, or other aspect of the release, as described herein. The authentication module 136 and the authorization module 138 may be configured to access a federated database system or architecture.

[0038] Data indicative of a user who approved a particular workflow action may be stored in the persistent store 130 and/or the workflow data store 124. A complete record of the approvals provided during each release workflow may thus be stored, which may be useful for compliance with corporate, governmental, or other regulations.

[0039] The administrator control interface 140 may be configured to allow an administrator or other authorized user to control other aspects of the release management system 102. In some embodiments, the administrator control interface 140 may provide various workflow control options. For example, a user may direct the workflow engine 122 to suspend, resume, or cancel one or more releases via the administrator control interface 140. Suspension of one or more releases may be useful in the event of, for instance, a failure (e.g., test failure) in one development branch that may have adverse impacts in other branches. The administrator control interface 140 may alternatively or additionally be used to override a variety of different actions automated via the release management system 102. For example, a workflow action specified via a template workflow may be skipped or otherwise cancelled. An override to lock any further workflow progress may be useful in certain situations, such as an outage that, until addressed, may adversely impact subsequent releases for the software product involving other clusters.

[0040] The policy engine 118 may provide control logic for the release management system 102. The control logic may be called or otherwise relied upon to determine whether to proceed with a workflow action in the release workflow. The control logic may include or be configured as a set of policy or gating rules. In this embodiment, the gating rules are set forth in a policy schema 142. The policy schema 142 may be configured as a data file or document, such as an extensible markup language (XML) document. The format, language, and other characteristics of the policy schema 142 may vary. The policy engine 118 may access a variety of different policy or configuration specifications to obtain the gating rules. With a particular gating rule of the policy schema 142 available,

the policy engine 118 is configured to direct the orchestrator 116 and/or the workflow engine 122 to send a successive instruction based on whether the gating rule is met. Alternatively or additionally, such direction of the orchestrator 116 and/or the workflow engine 122 may follow a request sent to the policy engine 118 from the orchestrator 116 and/or the workflow engine 122 on whether to proceed with a successive step in the workflow.

[0041] The policy schema 142 and/or the gating rules may be used to customize or configure the release workflow. Such customization may be at a high level. For example, the release workflow may be configured as a test workflow (e.g., a test to release workflow), a release-to-production workflow, or other release workflow. Such high level customization may be determinative of which software engineering systems are involved or included in the release pipeline and, in some cases, the order in which such systems are called upon.

[0042] The policy schema 142 and/or gating rules may customize the release workflow at an intermediate level in which the operation of the software engineering systems is configured. For example, the policy schema 142 may specify a test schema or set of tests to be implemented by the validation system 108. A workflow may be designed to implement only certain tests (or types of tests) made available by the validation system 108. Alternatively or additionally, the policy schema 142 and/or the gating rules may be configured to initiate an automated test selection routine, in which the tests (or types of tests) are selected based on environmental variables or other parameters of the release. For example, the test selection routine provided in the Team Foundation Server platform from Microsoft Corporation may be used.

[0043] Even more detailed levels of logic may be specified via the gating rules. For example, the policy schema 142 may include a gating rule specifying a code coverage threshold (e.g., 50% pass or 70% pass) to be met during validation. If the code coverage is below the threshold, then the workflow does not proceed to the next workflow action (e.g., a subsequent test, a deployment, etc.). The orchestrator 116 may instead send an instruction to the alert system 114 regarding the code coverage failure. In another example, the policy engine 118 may be configured via the policy schema 142 to apply gating rules regarding a state of a production environment. The policy engine 118 may instruct the orchestrator 116 to validate that the state of the production environment before releasing a build of the software product to the production environment. The policy schema 142 may specify one or more validation criteria as gating rules for the validation

process. In yet another example, a gating rule may be used to specify a location (e.g., a specific cluster) to which the software product is to be deployed and/or the circumstances under which such deployment occurs. A variety of other parameters for the release workflow and pipeline may be specified via the gating rules, including, for instance, whether the build results are to be replicated based on which code branch is built, and when or where to implement forward or reverse integration across branches.

[0044] The policy engine 118 may be directed to verification and quality assurance enforcement. In some embodiments, the gating rules are used by the policy engine 118 and/or the orchestrator 116 to determine whether a workflow incident occurs. Failure to meet a gating rule may cause an alert (e.g., a text message, email, or other message) to be sent to a user(s) authorized to respond to the workflow incident. The user(s) to which the alert is sent may be determined via the policy schema, the workflow, the workflow action parameter data package 128, and/or data specified via the authorization module 138. For example, the authorization module 138 may be used to establish that only users having project manager or higher roles are authorized to respond to workflow incidents of a certain level. The policy schema 142 may be relied upon to classify the various workflow incidents by level. The workflow engine 122 may be configured to resume the workflow based on a response to the alert from an authorized user.

[0045] The policy schema 142 (or other logic source) may include or specify a variety of transition criteria for the release workflow. Some transition criteria may be configured or applied as entry criteria. A gating rule may specify that a certain condition is met before allowing a workflow action to begin. For example, a condition may specify that a user having a Security Expert role provides signoff that a release has been reviewed for compliance with security practices, that appropriate Secure Development Lifecycle (SDL) tools from Microsoft Corporation (or other security assurances processes) have been run on the code to verify proper processes, or that code review has been done for the change by appropriate person(s) as specified by a compliance policy. Other gating rules may be configured or applied as exit criteria. A gating rule may specify that a certain condition is met before allowing a workflow action to be completed. For example, a condition may specify that continuous testing showed proper results (e.g., in accordance with service level agreement (SLA) restrictions) for all actions, or that no customer-impacting incidents were created during the time taken to validate the released changes (e.g., bake time), or that all of the service components entered steady-state and have not resulted in new failures.

[0046] The gating rules and, thus, the workflow policy, may be predetermined or predefined to support the implementation of the release workflow in a variety of ways. For example, the gating rules need not be specified via a discrete configuration file as shown in Figure 1. One or more gating rules may instead be specified in a workflow template and/or the workflow action parameter data package 128. For example, the workflow action parameter data package 128 (or other source of the gating rules) may specify a code coverage threshold (e.g., 80%). In another example, the workflow action parameter data package 128 (or other source of the gating rules) may specify a location of the source code for processing. The decision logic for the gating rules may alternatively or additionally be specified via workflow instructions incorporated (e.g., hard-coded) into the workflow template. For instance, the gating rules may be partially or fully hard-coded into the workflow template.

[0047] The policy engine 118 may include an interpreter or other data processor configured to evaluate data indicative of the gate rules. In one embodiment, the data may be arranged in a set of classes, an object library, or other data structure. The data structure may be set forth in an XML or other framework in one or more data files. The interpreter or other data processor of the policy engine 118 may then be configured to read the data structure to determine or load the gate rules. Alternatively or additionally, data indicative of the gate rules is set forth in a dynamic link library (DLL) file(s) or other compiled or processed format. In some cases, the gate rules may be set forth as code, instructions, or other logic, rather than data used to configure or customize such logic.

[0048] Figure 2 shows the orchestrator 116 and the workflow engine 122 thereof in greater detail. The orchestrator 116 may include a number of modules directed to interactions with the software engineering systems to achieve desired workflow results. In this example, the orchestrator 116 includes a communication manager 150 or other interface to support communications with the software engineering systems. The communication manager 150 may include or consult a lookup table or other logic to direct messages or other data exchanges to the appropriate software engineering system(s). The format, protocol, and other characteristics of the communications supported by the communication manager 150 may vary. For example, the communication manager 150 may send instructions to the software engineering systems via data exchanges configured in accordance with the capabilities of the software engineering systems, such as a respective API of the software engineering system. In some embodiments, interaction with an engineering system may be implemented by defining one or more interfaces that

describe or reflect the capabilities of the engineering system. For example, an interface may be configured in accordance with the expected capabilities of the above-referenced source control system. The interface may then be customized or otherwise implemented to support data exchanges (e.g., data retrieval) with specific instances of the source control system.

[0049] The communication manager 150 may also be configured to support interaction with other components of the release management system 102 (Figure 1), such as the policy engine 118 (Figure 1) and/or the front end system 120 (Figure 1). For example, the communication manager 150 may provide an interface for receiving and processing the workflow action parameter data package 128 (Figure 1) and/or other workflow action parameter data. Alternatively or additionally, the communication manager 150 may be configured to receive and process data indicative of the results of the application of the gating rules by the policy engine 118.

[0050] The workflow engine 122 may be configured to generate the instructions sent by the communication manager 150 and/or process the messages received by the communication manager 150. In this embodiment, the workflow engine 122 includes a workflow interpreter 152, a workflow validator 154, an execution environment 156, and an event scheduler 158. The workflow interpreter 152 may be configured to interpret workflow code authored in a workflow language, such as the workflow language established as part of the Microsoft® Windows® Workflow Foundation framework. Any workflow language may be used. The workflow language may define a set of primitive instructions configured to implement basic workflow actions (e.g., skip, cancel, run, run in a loop, run in parallel, change, etc.). The workflow language may be defined or based upon code written in the C# language and/or in XML script, although other high-level languages may be used. The workflow engine 122 may also include a compiler if the workflow code is presented in an intermediate language for which a compiler is not already available.

[0051] The workflow validator 154 may be one of several tools provided by the workflow engine 122 to author, modify, customize, or otherwise prepare a workflow or workflow template for implementation in the execution environment 156. The workflow validator 154 may be integrated with the workflow interpreter 152 to any desired extent. Once a workflow (or workflow template) has been validated, the workflow may be made available for implementation within the execution environment 156. The execution environment 156 may be based on or include the Common Language Runtime (CLR)

environment of the Microsoft® .NET framework and/or Microsoft® Windows® operating system, but other software runtime environments and frameworks may be used.

[0052] In some embodiments, the workflow engine 122 is configured to call the policy engine 118 (Figure 1) to incorporate the gating rules into the workflow during execution of the workflow (e.g., in the execution environment 156). The workflow may thus be configured or customized on the fly. Alternatively or additionally, data indicative of the gating rules (and/or other parameter data) is incorporated into the workflow before the execution environment 156 is implemented.

[0053] The workflow engine 122 and/or other components of the orchestrator 116 and the release management system 102 may be implemented in a cloud-based framework in which multiple virtual machines are implementing and managing a number of workflows. The cloud-based framework may be configured in a manner similar to the cloud-based service described in the above-referenced U.S. Patent Publication. For example, the cloud-based framework may provide a hosted service, such as a Windows Azure™ hosted service, with a scalable number of virtual machines providing multiple instances of an orchestrator role, a policy engine role, and/or a front end role of the release management system 102. The cloud-based framework may be useful for supporting parallel processing of one or more steps or actions of the release workflow. For example, virtual machines within one or more data centers may be used to implement validation testing in parallel. The workflow engine may be executed across multiple machines (virtual or physical), which may facilitate transfers of workflows across the machines, e.g., through the above-referenced persistent store and/or other predetermined persistence points. Such transfers may provide increased scalability in which many workflows are executed in parallel and/or increased resiliency to failure (e.g., when a machine executing a particular workflow fails).

[0054] The orchestrator 116 may include a number of modules directed to monitoring the execution of the workflows. In this example, the orchestrator 116 includes a system health monitoring module 160 and a dependency validation module 162. Additional, fewer, or alternative monitoring modules may be provided. In some cases, one or more of the modules may be provided as a component of the policy engine 118 (Figure 1) rather than the orchestrator 116. For example, the policy engine 118 may include the dependency validation module 162. Either way, the policy schema 142 (Figure 1) may be used to declare dependencies or other constraints (e.g., an upgrade order) for the software product.

[0055] The monitoring modules may be configured to detect an event, such as a failure, during the execution of a workflow that warrants an alert or other action. The system health monitoring module 160 may be configured to monitor the health or other state of the computing system implementing the orchestrator 116 (or any one or more components thereof). The system health monitoring module 160 may thus be configured to detect improper operating conditions and other conditions that may adversely impact the execution of the workflow. For example, the system health monitoring module 160 may be relied upon to detect when the computing resources (e.g., processor or memory capacity) of the computing system are inadequate for the load provided by the workflow. If a system health parameter does not meet a threshold, an alert or other message may then be sent to a user authorized (e.g., via the policy schema 142 (Figure 1)) to respond. In some cases, the system health monitoring module 160 may be configured to direct the workflow engine 122 to suspend the workflow. The workflow may then resume based on a response to the user alert or other user instruction.

[0056] Other components or aspects of the release environment may alternatively or additionally be monitored. The release environment may be an environment in which the software product is being used and, thus, subject to an SLA prohibiting or limiting downtime. In this example, the dependency validation module 162 is configured to validate and otherwise monitor a state of a production environment before releasing a build of the software product thereto. The validation may be based on one or more validation criteria specified in the policy schema 142 (Figure 1). For example, the validation criteria may involve or specify a number of dependencies for the portion of the software product to be released. The dependency validation module 162 may be configured to determine a state of each dependency (e.g., determine a version of a component) and validate the state of the dependency before implementing the release (e.g., confirm that the version is compatible with the release). Such validation may avoid rendering the software product inoperable as a result of the release. The health of the operating environment governed by the SLA may thus be maintained.

[0057] The orchestrator 116 may include additional modules for supporting the management of the release workflow. In this embodiment, the orchestrator 116 includes an authorization module 164 directed to managing the permissions of users and/or user roles. The functionality of the authorization module 164 may alternatively or additionally be provided by one or more components of the front end system 120 (Figure 1). The authorization module 164 may be integrated with such components to any desired extent.

In some embodiments, the role management functionality described below may thus be provided by a component of the front end system 120.

[0058] The authorization module 164 may implement a role-based access management framework for the orchestrator 116 and other components of the disclosed systems. Users may be assigned roles to define, limit, or otherwise control the workflow actions, events, or other items with which the user may interact or control. For instance, a user assigned to a developer role may be prohibited from approving a release or deployment to production. The set of permissions may vary considerably with the nature of the actions involved in a release process. Some roles may be capable of initiating a forward integration within a release workflow, but not deployments. Some roles may initiate a build process, but not approve the results of a build.

[0059] The policy schema 142 (Figure 1) may be used to establish rules for the roles. For example, the policy schema 142 may be used to allow only users assigned to a project manager role to approve a release for a certain software product. In this way, other roles may be capable of approving a release for other software products. Alternatively or additionally, the authorization module 164 and/or the authorization module 138 (Figure 1) of the front end system 120 (Figure 1) may be used to establish permissions for the roles. In some cases, the permissions may be global (e.g., across all software products, environments, etc.) or default values that may be customized or overridden by permissions specified via the policy schema 142.

[0060] A variety of different roles and permissions for such roles may be established via the authorization module 164. An administrator role may be defined to provide override and other control capabilities, including, for instance, those referenced above. Some roles may classify a user in accordance with a team involved in the software release process, such as an operations team or a testing team.

[0061] The authorization module 164 may include role manager functionality. For example, the authorization module 164 may be configured to map users to roles and/or permissions, and/or to map roles to software products. The role manager functionality may allow the authorization module 164 to apply permissions on a product-by-product basis. Users may thus be associated with one or more specific products and/or one or more specific services.

[0062] In some embodiments, the role access framework may include a data structure for classifying role membership, role access or permissions, and scope definition. The role membership data may specify which users belong to which role(s). Roles need not

correspond with a corporate structure (e.g., project manager, etc.), but rather may be specific to the release process (e.g., approvers, initiators, etc.). Alternatively or additionally, the role membership data may specify the products or services with which the user is associated. The role membership data may also specify the scope of such associations, such as the components of such products or services with which the user is associated. A user may thus be associated with a team building a portal for the software product, but not a team building a software development kit for the same product. The role access data may specify the permissions for each role.

[0063] The front end system 120 (Figure 1) may be used to establish the role access framework. For example, an administrator user may enter the role membership data, the role access data, and/or the scope definition data via the authorization module 138. Alternatively or additionally, such authorization data may be specified and/or managed via an external service (e.g., a directory service) or system, such as a cloud-based system or a corporate system.

[0064] Figure 3 depicts an exemplary method for managing a software release. The method is computer-implemented. For example, one or more server computers of the release management system 102 shown in Figure 1 may be configured to implement one or more of the acts of the method. The implementation of each act may be directed by respective computer-readable instructions executed by a processor of the release management system 102 and/or another processor or processing system.

[0065] In the embodiment of Figure 3, the method begins with an act 200 in which a user is authenticated. The user may be authenticated in connection with an attempt to access the release management system 102 (Figure 1) via the front end system 120 (Figure 1). The user may be accessing the release management system 102 for various reasons. Given the appropriate permissions, the user may access the release management system 102, for instance, to initiate, suspend, or cancel a release, to monitor an ongoing release, to configure or manage a release workflow, and/or to configure the release management system 102 (e.g., modify a template workflow). The authentication may be managed by the authentication module 136 (Figure 1) of the front end system 136 and/or one or more external authentication systems (e.g., a federated database system). For example, the authentication process may include accessing a private network user directory in an act 202.

[0066] In some cases, an administrator user may access the release management system 102 to define the permissions of various user roles in an act 203. Data indicative of the

permissions may be stored in, for example, the workflow data store 124 (Figure 1) or other data store for subsequent access by the authorization module 138 (Figure 1) and/or the authorization module 164 (Figure 2). Additional or alternative authorization data may be stored. Data indicative of the permissions may be stored at another time, such as during an initial configuration session.

[0067] In act 204, a request is received or otherwise obtained for a release. The request specifies or includes workflow action parameter data to define a release pipeline involving a combination of software engineering systems. For example, a user may specify a software product to be released (including, e.g., a source code location), a target environment to which the software is to be released, an order of the software engineering systems in the pipeline, and/or a selection of a subset of such systems. Additional, fewer, or alternative workflow parameters may be specified in connection with the release request. Each software engineering system may be configured to process data indicative of the software product to be released, as described above.

[0068] The front end system 120 (Figure 1) may facilitate the specification of the workflow action parameter data by generating the user interface 126 (Figure 1) or other request specification user interface in an act 206. The layout and other characteristics of the user interface may vary. In some cases, the user interface may allow the user to create, modify, or otherwise configure a workflow for execution. The configuration may include entering workflow code to customize a template workflow to create a new workflow.

[0069] Once the workflow request and/or the workflow parameter action data is specified, the method may then include confirming that the user is authorized to initiate the workflow in an act 208. The authorization process may include determining a user role of the user and determining the permissions of the user role for a given software product. The permissions may be configured to authorize the user to initiate certain types of releases or workflow actions, but not others. The authorization process may be implemented by the authorization module 138 (Figure 1) and/or the authorization module 164 (Figure 2).

[0070] Code and/or data indicative of the workflow may be processed in an act 210. In some embodiments, the workflow may be based on a template workflow modified in accordance with the workflow parameter action data. The workflow template may be processed in preparation for execution once modified in accordance with the workflow parameter action data. The processing may include code interpretation, workflow validation, compilation, linking, and other code processing.

[0071] In some embodiments, an act 212 is implemented to obtain an approval or sign-off of the initiation of a release workflow before the execution of the workflow begins. For example, a notification may be sent to a user authorized to approve the release. The notification may include an email, a text message, or any other transmission. The notification may specify data indicative of the release workflow, including, for instance, one or more parameters specified via the workflow action parameter data, such as the release pipeline. A decision block 214 determines whether the release is approved. If yes, then control passes to a set of blocks associated with execution of the workflow. If the release is not approved, then the method ends and the release process is discontinued. Alternatively, control may pass to a state in which a user is given an opportunity to take a corrective action (e.g., an update or modification to address a comment from an approver).

[0072] The workflow is executed in an act 216 in which a series of instructions are sent to the software engineering systems in the release pipeline. The release pipeline is implemented in accordance with the workflow action parameter data. The execution of the workflow may include or involve data generation for monitoring, compliance, or other purposes. At various points in the workflow, further execution of the workflow (e.g., a next step in the pipeline) may involve obtaining an approval from an authorized user. In the embodiment of Figure 3, a callback 217 to the act 212 may be implemented to send another notification to an approver. Further execution of the workflow may then continue if the decision block 214 determines that the approval is received.

[0073] In this example, data indicative of such approvals, workflow metrics, and/or other data indicative of actions implemented during workflow execution are recorded in an act 218. For example, such data may be recorded in one or more log files, which may be stored in one of the above-referenced data stores and/or another data store. Such data may specify a user who approved a specific action (e.g., the initiation of the workflow), as described herein. The act 218 may also include the presentation of data indicative of the workflow execution via a dashboard generated by the front end system 120 (Figure 1). The dashboard may display workflow metric data as described above. One or more metrics to be displayed via the dashboard during execution of the workflow may be specified via the workflow action parameter data provided in connection with the release request.

[0074] A policy schema may be accessed in an act 220 to obtain any gating or other rules for the workflow. The policy schema may include a configuration file, such as an XML file, as described above. Alternatively or additionally, data indicative of some or all of the policy schema (and/or gating rules) may be accessed via another mechanism, such as via a

user interface generated by the release management system 102 (Figure 1). For example, a gating rule may be provided via a user interface during the execution of the workflow. In some embodiments, the policy schema may be accessed to configure an instruction in the series of instructions provided to the software engineering systems, which may, for example, involve application of one or more gating rules.

[0075] During execution of the workflow, a successive instruction in the series of instructions is sent based on whether a transition criterion or other gating rule of the configuration file or other specification of the policy schema for the release is met. In this embodiment, an entry criterion or an exit criterion is applied in an act 222. Such transition criteria may specify, for example, quality assurance rules for entering or exiting a step in the workflow, as described above. Alternatively, one of the transition criteria may specify a production environment compliance rule for entering or exiting a step in the workflow, as also described above. In yet another example, one of the transition criteria may specify a user role to approve implementation of a successive instruction in the series of instructions, including, for instance, initiation of the workflow. The transition criteria may be applied via a policy engine. The policy engine may configure an instruction in the series of instructions based on the gating rule(s) of the policy schema.

[0076] The embodiment of Figure 3 includes a number of acts that may be implemented during execution of the workflow. In some cases, the workflow action parameter data specifies that such acts be implemented. A system health parameter of a computer (e.g., a server computer) or computing system executing the workflow may be monitored in an act 224. A state of the production environment may be monitored or validated in an act 226 to ensure that a validation criterion (relating to, e.g., a dependency of the software product) is met before, for instance, a build is released to the production environment. In dependency validation examples, validating the state of the production environment may thus include determining a state of the dependency, and validating the state of the dependency before implementing the release. A release action or other event (e.g., a release to the production environment) may be scheduled in an act 228 for a later time based on the state of the production environment (or other state monitored during workflow execution). The scheduling may include specifying a constraint or condition to be met (relating to, e.g., a state of the dependency) before allowing the release or workflow action to be implemented.

[0077] A decision block 230 may determine whether an incident has occurred during workflow execution. A workflow incident may occur due to a failure to meet a transition

criterion or other gating rule. For example, the decision block 230 may determine whether a system health parameter does not meet a threshold during the monitoring of the act 224, or whether a dependency of the software product does not meet a validation criterion during the validation of the act 226. Additional, fewer, or alternative types of incidents may be monitored. If an incident occurs, an alert is sent in an act 232 to a user authorized to respond. The authorization of the user to respond to the incident may be established or specified via the policy schema as described above (e.g., via a user role assignment). Another decision block 234 determines whether the workflow may be restarted, resumed (e.g., a workflow action may be retried), or skipped based on a response to the alert (e.g., from an authorized user). If a suitable response is received, control may return to the act 216 for further workflow execution. If not, then the execution of the release workflow may terminate. In some embodiments, the response to the alert may involve or include a gating rule change (e.g., within the above-referenced decision engine) or other change that renders the original criterion no longer applicable. In such cases, the workflow may resume automatically.

[0078] The release workflow may also be terminated via an override or other control provided by an administrator or other authorized user. For example, the decision block 230 may be configured to suspend the execution of the workflow as shown in Figure 3.

[0079] If the decision block 230 is reached after successful implementation of the series of instructions in the workflow (e.g., without any incidents or any unaddressed incidents), then control may pass to an act 236 in which an alert or other message is sent indicating the successful completion of the release.

[0080] The order of the acts of the method may vary in addition to those variations identified above. For example, the policy schema may be accessed before the execution of the workflow and/or the processing of the workflow code. The workflow may thus be configured in accordance with the policy schema and the workflow action parameter data before execution. In such cases, one or more gating rules of the policy schema may alternatively or additionally be specified during execution, as described above.

[0081] With reference to Figure 4, an exemplary computing environment 300 may be used to implement one or more aspects or elements of the above-described methods and/or systems. The computing environment 300 may be used by, or incorporated into, one or more elements of the release management system 102 (Figure 1). For example, the computing environment 300 may be used to implement the orchestrator 116, the workflow engine 122, the policy engine 118, and/or the front end system 120. The computing

environment 300 may be used or included as a client, network server, application server, or database management system or other data store manager, of any of the aforementioned elements or system components. The computing environment 300 may be used to implement one or more of the acts described in connection with Figure 3.

[0082] The computing environment 300 includes a general-purpose computing device in the form of a computer 310. Components of computer 310 may include, but are not limited to, a processing unit 320, a system memory 330, and a system bus 321 that couples various system components including the system memory to the processing unit 320. The system bus 321 may be any of several types of bus structures including a memory bus or memory controller, a peripheral bus, and a local bus using any of a variety of bus architectures. By way of example, and not limitation, such architectures include Industry Standard Architecture (ISA) bus, Micro Channel Architecture (MCA) bus, Enhanced ISA (EISA) bus, Video Electronics Standards Association (VESA) local bus, and Peripheral Component Interconnect (PCI) bus also known as Mezzanine bus. The units, components, and other hardware of computer 310 may vary from the example shown.

[0083] Computer 310 typically includes a variety of computer readable storage media configured to store instructions and other data. Such computer readable storage media may be any available media that may be accessed by computer 310 and includes both volatile and nonvolatile media, removable and non-removable media. Such computer readable storage media may include computer storage media as distinguished from communication media. Computer storage media includes both volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer readable instructions, data structures, program modules or other data. Computer storage media includes, but is not limited to, RAM, ROM, EEPROM, flash memory or other memory technology, CD-ROM, digital versatile disks (DVD) or other optical disk storage, magnetic cassettes, magnetic tape, magnetic disk storage or other magnetic storage devices, or any other medium which may be used to store the desired information and which may accessed by computer 310.

[0084] The system memory 330 includes computer storage media in the form of volatile and/or nonvolatile memory such as read only memory (ROM) 331 and random access memory (RAM) 332. A basic input/output system 333 (BIOS), containing the basic routines that help to transfer information between elements within computer 310, such as during start-up, is typically stored in ROM 331. RAM 332 typically contains data and/or program modules that are immediately accessible to and/or presently being operated on by

processing unit 320. By way of example, and not limitation, Figure 4 illustrates operating system 334, application programs 335, other program modules 336, and program data 337. For example, one or more of the application programs 335 may be directed to implementing one or more components of the orchestrator 116, the workflow engine 122, the policy engine 118, the front end system 120, and/or any instruction sets of the systems and methods described above. In this or another example, any one or more the instruction sets in the above-described memories or data storage devices may be stored as program data 337.

[0085] Any one or more of the operating system 334, the application programs 335, the other program modules 336, and the program data 337 may be stored on, and implemented via, a system on a chip (SOC). Any of the above-described modules may be implemented via one or more SOC devices. The extent to which the above-described modules are integrated in a SOC or other device may vary.

[0086] The computer 310 may also include other removable/non-removable, volatile/nonvolatile computer storage media. By way of example only, Figure 4 illustrates a hard disk drive 341 that reads from or writes to non-removable, nonvolatile magnetic media, a magnetic disk drive 351 that reads from or writes to a removable, nonvolatile magnetic disk 352, and an optical disk drive 355 that reads from or writes to a removable, nonvolatile optical disk 356 such as a CD ROM or other optical media. Other removable/non-removable, volatile/nonvolatile computer storage media that may be used in the exemplary operating environment include, but are not limited to, magnetic tape cassettes, flash memory cards, digital versatile disks, digital video tape, solid state RAM, solid state ROM, and the like. The hard disk drive 341 is typically connected to the system bus 321 through a non-removable memory interface such as interface 340, and magnetic disk drive 351 and optical disk drive 355 are typically connected to the system bus 321 by a removable memory interface, such as interface 350.

[0087] The drives and their associated computer storage media discussed above and illustrated in Figure 4, provide storage of computer readable instructions, data structures, program modules and other data for the computer 310. For example, hard disk drive 341 is illustrated as storing operating system 344, application programs 345, other program modules 346, and program data 347. These components may either be the same as or different from operating system 334, application programs 335, other program modules 336, and program data 337. Operating system 344, application programs 345, other program modules 346, and program data 347 are given different numbers here to illustrate

that, at a minimum, they are different copies. A user may enter commands and information into the computer 310 through input devices such as a keyboard 362 and pointing device 361, commonly referred to as a mouse, trackball or touch pad. Other input devices (not shown) may include a microphone (e.g., for voice control), touchscreen (e.g., for touch-based gestures and other movements), ranger sensor or other camera (e.g., for gestures and other movements), joystick, game pad, satellite dish, and scanner. These and other input devices are often connected to the processing unit 320 through a user input interface 360 that is coupled to the system bus, but may be connected by other interface and bus structures, such as a parallel port, game port or a universal serial bus (USB). In some implementations of the computer 310, a monitor 391 or other type of display device is also connected to the system bus 321 via an interface, such as a video interface 390. In addition to the monitor, computers may also include other peripheral output devices such as printer 396 and speakers 397, which may be connected through an output peripheral interface 395.

[0088] The computer 310 may operate in a networked environment using logical connections to one or more remote computers, such as a remote computer 380. The remote computer 380 may be a personal computer, a server, a router, a network PC, a peer device or other common network node, and typically includes many or all of the elements described above relative to the computer 310, although only a memory storage device 381 has been illustrated in Figure 4. The logical connections include a local area network (LAN) 371 and a wide area network (WAN) 373, but may also include other networks. Such networking environments are commonplace in offices, enterprise-wide computer networks, intranets and the Internet.

[0089] When used in a LAN networking environment, the computer 310 is connected to the LAN 371 through a network interface or adapter 370. When used in a WAN networking environment, the computer 310 typically includes a modem 372 or other means for establishing communications over the WAN 373, such as the Internet. The modem 372, which may be internal or external, may be connected to the system bus 321 via the user input interface 360, or other appropriate mechanism. In a networked environment, program modules depicted relative to the computer 310, or portions thereof, may be stored in the remote memory storage device. Figure 4 illustrates remote application programs 385 as residing on memory device 381. The network connections shown are exemplary and other means of establishing a communications link between the computers may be used.

[0090] The computing environment 300 of Figure 4 is only one example of a suitable computing environment and is not intended to suggest any limitation as to the scope of use or functionality of the technology herein. Neither should the computing environment 300 be interpreted as having any dependency or requirement relating to any one or combination of components illustrated in the exemplary operating environment 300.

[0091] The technology described herein is operational with numerous other general purpose or special purpose computing system environments or configurations. Examples of well-known computing systems, environments, and/or configurations that may be suitable for use with the technology herein include, but are not limited to, personal computers, server computers (including server-client architectures), hand-held or laptop devices, mobile phones or devices, multiprocessor systems, microprocessor-based systems, set top boxes, programmable consumer electronics, network PCs, minicomputers, mainframe computers, distributed computing environments that include any of the above systems or devices, and the like.

[0092] The technology herein may be described in the general context of computer-executable instructions, such as program modules, being executed by a computer. Generally, program modules include routines, programs, objects, components, data structures, and so forth that perform particular tasks or implement particular abstract data types. The technology herein may also be practiced in distributed computing environments where tasks are performed by remote processing devices that are linked through a communications network. In a distributed computing environment, program modules may be located in both local and remote computer storage media including memory storage devices.

[0093] While the present invention has been described with reference to specific examples, which are intended to be illustrative only and not to be limiting of the invention, it will be apparent to those of ordinary skill in the art that changes, additions and/or deletions may be made to the disclosed embodiments without departing from the spirit and scope of the invention.

[0094] The foregoing description is given for clearness of understanding only, and no unnecessary limitations should be understood therefrom, as modifications within the scope of the invention may be apparent to those having ordinary skill in the art.

CLAIMS

1. A computer-implemented method for managing a release of a software product, the method comprising:

obtaining a request for the release, the request comprising workflow action parameter data to define a release pipeline involving a plurality of software engineering systems configured to process data indicative of the software product; and

executing, with a processor, a workflow to implement the release pipeline in accordance with the workflow action parameter data;

wherein executing the workflow comprises sending a series of instructions to the plurality of software engineering systems; and

wherein a successive instruction in the series of instructions is sent based on whether a gating rule for the release is met.

2. The computer-implemented method of claim 1, further comprising accessing a configuration file that comprises data indicative of the gating rule.

3. The computer-implemented method of claim 1, wherein the gating rule specifies a quality assurance rule for entering or exiting a step in the workflow.

4. The computer-implemented method of claim 1, wherein executing the workflow comprises resuming the workflow if the gating rule is no longer applicable.

5. The computer-implemented method of claim 1, wherein executing the workflow comprises accessing a policy schema to determine the gating rule and to configure an instruction in the series of instructions.

6. The computer-implemented method of claim 1, wherein the gating rule specifies a user role to approve implementation of a successive instruction in the series of instructions.

7. The computer-implemented method of claim 1, wherein executing the workflow comprises:

determining whether a workflow incident occurs due to failure to meet the gating rule;

sending an alert to a user authorized via a policy schema to respond to the workflow incident; and

resuming the workflow based on a response to the alert from the user.

8. The computer-implemented method of claim 1, wherein executing the workflow comprises recording, in a data store, data indicative of an action implemented during the workflow and a user who approved the action, the user being assigned to a user role having a permission via a policy schema to approve the action.

9. The computer-implemented method of claim 1, wherein:

executing the workflow comprises monitoring a system health parameter of a computer comprising the processor;

sending an alert to a user authorized via a policy schema to respond if the system health parameter does not meet a threshold; and

resuming the workflow based on a response to the alert from the user.

10. The computer-implemented method of claim 1, further comprising validating a state of a production environment before releasing a build of the software product to the production environment based on a validation criterion specified in a policy schema.

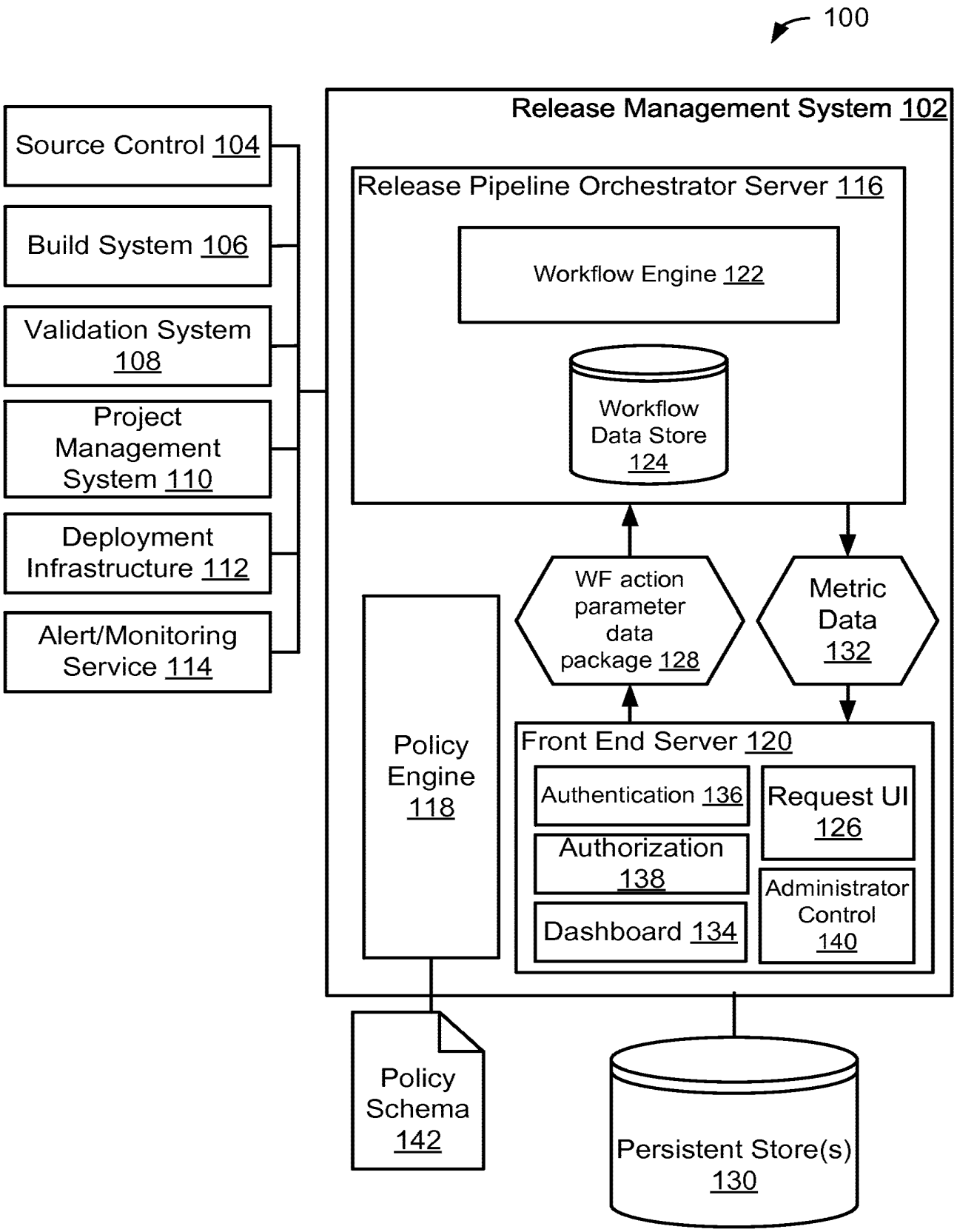

100
Release Management System 102
Source Control 104
Build System 106
Validation System 108
Project Management System 110
Deployment Infrastructure 112
Alert/Monitoring Service 114
Release Pipeline Orchestrator Server 116
Workflow Engine 122
Workflow Data Store 124
WF action parameter data package 128
Metric Data 132
Policy Engine 118
Front End Server 120
Authentication 136
Request UI 126
Authorization 138
Administrator Control 140
Dashboard 134
Policy Schema 142
Persistent Store(s) 130

FIG. 1

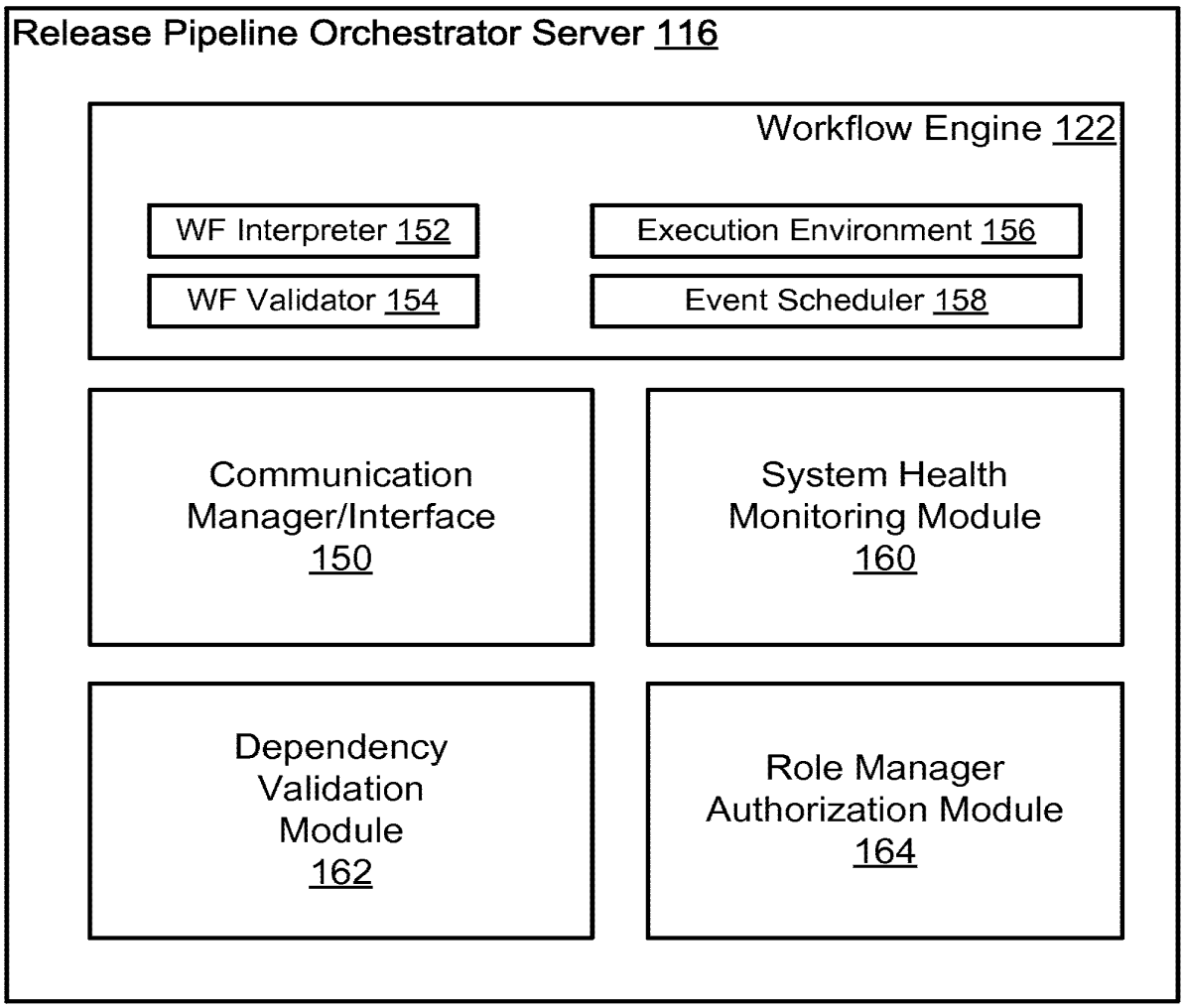

Release Pipeline Orchestrator Server 116
Workflow Engine 122
WF Interpreter 152
Execution Environment 156
WF Validator 154
Event Scheduler 158
Communication Manager/Interface 150
System Health Monitoring Module 160
Dependency Validation Module 162
Role Manager Authorization Module 164

FIG. 2

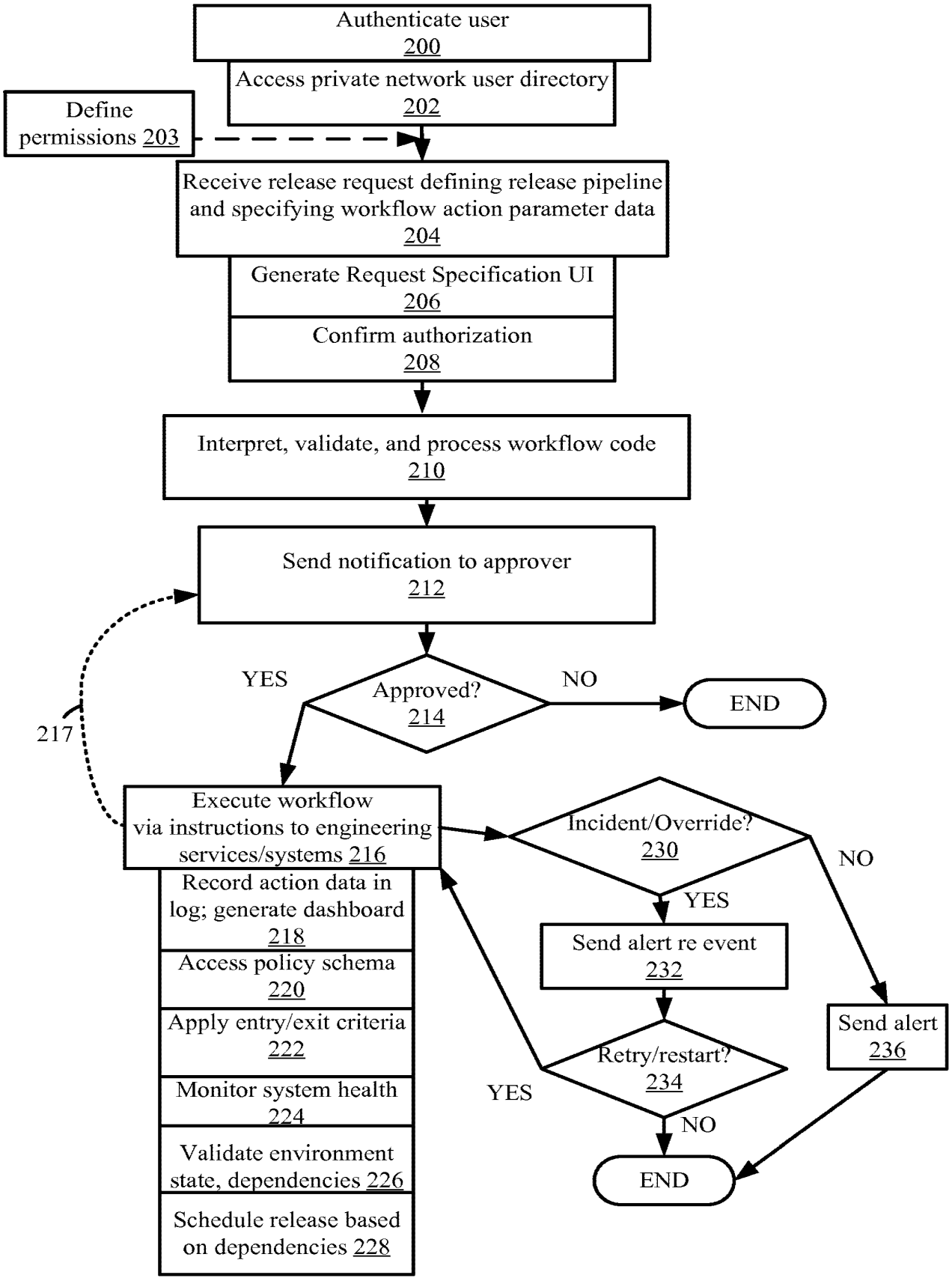

Authenticate user
200
Access private network user directory
202
Define permissions 203
Receive release request defining release pipeline and specifying workflow action parameter data
204
Generate Request Specification UI
206
Confirm authorization
208
Interpret, validate, and process workflow code
210
Send notification to approver
212
YES
Approved?
214
NO
END
217
Execute workflow via instructions to engineering services/systems 216
Record action data in log; generate dashboard
218
Access policy schema
220
Apply entry/exit criteria
222
Monitor system health
224
Validate environment state, dependencies 226
Schedule release based on dependencies 228
Incident/Override?
230
YES
NO
Send alert re event
232
Retry/restart?
234
YES
NO
Send alert
236
END

FIG. 3

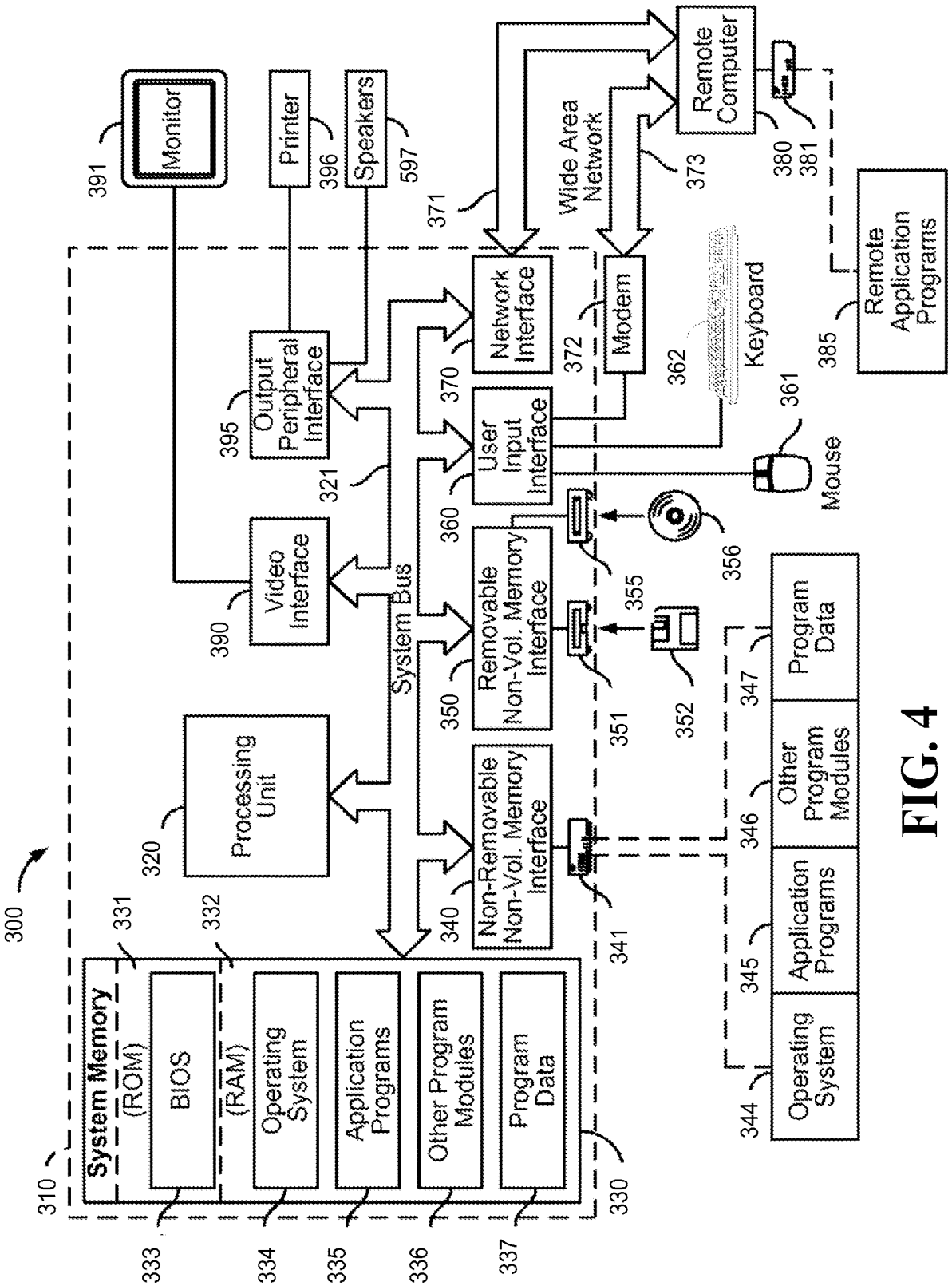

300
310
System Memory
(ROM)
BIOS
331
(RAM)
Operating System
332
Application Programs
Other Program Modules
Program Data
333
334
335
336
337
320
Processing Unit
340
Non-Removable Non-Vol. Memory Interface
341
350
Removable Non-Vol. Memory Interface
351
352
355
356
360
User Input Interface
361
Mouse
362
Keyboard
370
Network Interface
371
Wide Area Network
372
Modem
373
Remote Computer
380
381
385
Remote Application Programs
390
Video Interface
391
Monitor
395
Output Peripheral Interface
396
Printer
597
Speakers
321
System Bus
330
Program Data
347
Other Program Modules
346
Application Programs
345
Operating System
344

FIG. 4

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2014/022232

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F9/44
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	EP 1 881 447 A1 (MICROSOFT CORP [US]) 23 January 2008 (2008-01-23) abstract paragraphs [0006], [0007], [0010] paragraphs [0035], [0038], [0043], [0044] paragraph [0057] -----	1-10
A	MOORE R J: "Evolving to a lighter softwar process: a case study", SOFTWARE ENGINEERING WORKSHOP, 2001. PROCEEDINGS. 26TH ANNUAL NASA GOD DARD 27-29 NOV. 2001, PISCATAWAY, NJ, USA,IEEE, 27 November 2001 (2001-11-27), pages 14-21, XP010585334, ISBN: 978-0-7695-1456-7 the whole document ----- -/--	1-10

[X] Further documents are listed in the continuation of Box C.

[X] See patent family annex.

* Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
30 June 2014	08/07/2014

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Uhlmann, Nikolay

1

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2014/022232

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2007/162904 A1 (KIMMERLY RANDY S [US]) 12 July 2007 (2007-07-12) the whole document -----	1-10

1

Form PCT/ISA/210 (continuation of second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2014/022232

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP 1881447 A1	23-01-2008	NONE	
US 2007162904 A1	12-07-2007	CN 101361053 A	04-02-2009
		EP 1971931 A1	24-09-2008
		JP 5147724 B2	20-02-2013
		JP 2009524129 A	25-06-2009
		KR 20080087812 A	01-10-2008
		US 2007162904 A1	12-07-2007
		WO 2007081508 A1	19-07-2007

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC ACKNOWLEDGEMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
18/779,029	**07/21/2024 06:41:16 PM Z ET**	**15305FTNK**

Title of Invention

RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	-
CONFIRMATION #	4119	FILED BY	Aura Morrey
PATENT CENTER #	66453776	FILING DATE	-
CUSTOMER #	82318	FIRST NAMED INVENTOR	Jason Crabtree
CORRESPONDENCE ADDRESS	-	AUTHORIZED BY	Brian Galvin

Documents

TOTAL DOCUMENTS: 46

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
240721_15305FTNK_ADS_AS_FILED.pdf	15	Application Data Sheet	2227 KB
240721_15305TNK_TrackOne Request_AS_FILED.pdf	2	Track One Request	122 KB
240721_15305FTNK_Final_AS_FILED-APP.TEXT.docx	37	Application body structured text document	46 KB
240721_15305FTNK_Drawings_AS_FILED.pdf	16	Drawings-only black and white line drawings	3517 KB
240721_15305FTNK_QOMPLX LLC_POA_AS_FILED.pdf	1	Power of Attorney	135 KB

240721_15305FTNK_Declaration_Crabtree_AS_FILED.pdf	1	Oath or Declaration filed	540 KB
240721_15305FTNK_Declaration_Sellers_AS_FILED.pdf	1	Oath or Declaration filed	811 KB
240721_15305FTNK_IDS_1of 3_AS_FILED.pdf	33	Information Disclosure Statement (IDS) Form (SB08)	1339 KB
240721_15305FTNK_Final_AS_FILED.pdf	37	Auxiliary PDF of Application	212 KB
240721_15305FTNK_IDS_2of 3_AS_FILED.pdf	13	Information Disclosure Statement (IDS) Form (SB08)	1334 KB
240721_15305FTNK_IDS_3of 3_AS_FILED.pdf	5	Information Disclosure Statement (IDS) Form (SB08)	1332 KB
240721_15305FTNK_ForeignRef_Atsmon_WO2020079685A1_AS_FILED.pdf	54	Foreign Reference	483 KB
240721_15305FTNK_ForeignRef_Gupta_WO2015089463A1_AS_FILED.pdf	55	Foreign Reference	2457 KB
240721_15305FTNK_ForeignRef_Jubran_WO2014159150A1_AS_FILED.pdf	39	Foreign Reference	2074 KB
240721_15305FTNK_ForeignRef_Theimer_CA2930026C_AS_FILED.pdf	127	Foreign Reference	6753 KB
240721_15305FTNK_NPL_Cui et al_Non-Intrusive...Monitor System..._AS_FILED.pdf	10	Non Patent Literature	1263 KB
240721_15305FTNK_NPL_G	17	Non Patent Literature	1734 KB

edik et al_Elastic Scaling for...Processing_AS_FILED.pdf			
240721_15305FTNK_NPL_Ekelhart et al_Integrating attker behavior..._AS_FILED.pdf	14	Non Patent Literature	1892 KB
240721_15305FTNK_NPL_JPCert_Detect Lateral Movement...v2_AS_FILED.pdf	17	Non Patent Literature	458 KB
240721_15305FTNK_NPL_Kbar_Wireless Netwrk...Authentication_AS_FILED.pdf	7	Non Patent Literature	882 KB
240721_15305FTNK_NPL_Kotenko et al_Cyber Attk Model..._AS_FILED.pdf	25	Non Patent Literature	730 KB
240721_15305FTNK_NPL_Boukhtouta et al_Graph-Theoretic Character..._AS_FILED.pdf	13	Non Patent Literature	3300 KB
240721_15305FTNK_NPL_Kumar et al_DFuse..._AS_FILED.pdf	12	Non Patent Literature	862 KB
240721_15305FTNK_NPL_Lu et al_Sybil Attk Detect..._AS_FILED.pdf	11	Non Patent Literature	776 KB
240721_15305FTNK_NPL_Microsoft_Mitigat P-t-H Attks..._AS_FILED.pdf	82	Non Patent Literature	1100 KB
240721_15305FTNK_NPL_Kiesling et al_Selecting Secur	35	Non Patent Literature	2980 KB

Cntrl Porfolios..._AS_FILED.pdf			
240721_15305FTNK_NPL_Patapanchala_Explr Secure Metrics...Graphs_AS_FILED.pdf	94	Non Patent Literature	1156 KB
240721_15305FTNK_NPL_Yang et al_Attack Projection_AS_FILED.pdf	24	Non Patent Literature	604 KB
240721_15305FTNK_NPL_Shandilya_...Attk Graphs in Security Systems_AS_FILED.pdf	14	Non Patent Literature	1532 KB
240721_15305FTNK_NPL_Zargar et al_0 Knwldge Anomaly-Based...Threats_AS_FILED.pdf	7	Non Patent Literature	657 KB
240721_15305FTNK_NPL_Fisk et al_ Agile nd Scalable Analysis..._AS_FILED.pdf	6	Non Patent Literature	6854 KB
240721_15305FTNK_NPL_Jajodia et al_Advncd Cyber Attk..._AS_FILED.pdf	113	Non Patent Literature	8259 KB
240721_15305FTNK_NPL_Pasqualetti et al_Attack Detection...Systems_AS_FILED.pdf	15	Non Patent Literature	3840 KB
240721_15305FTNK_NPL_Apache_Beam_AS_FILED.pdf	31	Non Patent Literature	30 KB
240721_15305FTNK_NPL_Apache_Airflow_AS_FILED.pdf	75	Non Patent Literature	65 KB

240721_15305FTNK_NPL_Apache_Apex_AS_FILED.pdf	158	Non Patent Literature	179 KB
240721_15305FTNK_NPL_Apache_AS_FILED.pdf	286	Non Patent Literature	296 KB
240721_15305FTNK_NPL_Apache_Ignite_AS_FILED.pdf	103	Non Patent Literature	88 KB
240721_15305FTNK_NPL_Apache_Kafka_AS_FILED.pdf	13	Non Patent Literature	13 KB
240721_15305FTNK_NPL_Apache_Flink_AS_FILED.pdf	712	Non Patent Literature	675 KB
240721_15305FTNK_NPL_Apache_Logging_Flume_AS_FILED.pdf	200	Non Patent Literature	191 KB
240721_15305FTNK_NPL_Apache_Samza_AS_FILED.pdf	221	Non Patent Literature	253 KB
240721_15305FTNK_NPL_Apache_Hadoop_AS_FILED.pdf	1640	Non Patent Literature	1505 KB
240721_15305FTNK_NPL_Apache_Storm_AS_FILED.pdf	589	Non Patent Literature	613 KB
240721_15305FTNK_NPL_Apache_Spark_AS_FILED.pdf	2122	Non Patent Literature	1972 KB
240721_15305FTNK_IntCommAuth_AS_FILED.pdf	2	Internet Communications Authorization	83 KB

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)
240721_15305FTNK_ADS_AS_FILED.pdf	EF30E119729994C4EFA093E97ECE25F4658FFFE37639A6D3072E7772E3912E1EDD5A880F2AFE5202FFC53E41DACEF661EB3DFBC2E8171747B3C9183504219786
240721_15305TNK_TrackOneRequest_AS_FILED.pdf	D3B8D24990FDC6E17F5ED66603CC2471029A2F2641A54825F7DD325DADF445AD5ACF1C612CE3EF126E31DBA067B6032F1DB8C3D4835536A5A7AF16630B0C088D
240721_15305FTNK_Final_AS_FILED-APP.TEXT.docx	5975E089AC50FEE50AEE71C1A0AEE9E36959E320AE8F0925BF5EBE54FDD50373535A89D80B36AB1B5C953665DD395B61AEE3B9CBFD44E260D7A0A37D58ECEBDD
240721_15305FTNK_Drawings_AS_FILED.pdf	98138F81DA4F68C57230D29407BD041D030AF94ECB50538064196B6954814247800522346FF6BFBF672399A77B61249303DB5108BA897E4758040B27E4C748C7
240721_15305FTNK_QOMPLX LLC_POA_AS_FILED.pdf	5B1E94F95A5F07E4D3315EA25D6D9A53CC5B1FEAE95A4D9F0804261380 6B5B8FA2B819378C5111EA0DF327E358EBF125DDCC6D8C9CDD8C36866F866DF5534759
240721_15305FTNK_Declaration_Crabtree_AS_FILED.pdf	490BC86C6BFEC9BA156C6F16F140E0626B3A147A278821AAAE9A0687BA2833B77510C9C9C238E468D3CEE7B7B911FDF601CB4550D60C0AF183000ECF50B9D35C
240721_15305FTNK_Declaration_Sellers_AS_FILED.pdf	9C2DB2BD37C49C4B8BB8D19681840E227E00C19B29B7A9CE3561324CD107D7928633B89E6B1C7C3A515CA319E48A51F700BBDD4797EDB24FD394E0D3EC478CF4
240721_15305FTNK_IDS_1 of 3_AS_FILED.pdf	5D56F21AE3D1D7655EA534946FBDD662AFAA5EDFA6EB40199B1217AB319C504EF2AA501BFEF5C967A34AE3EC32974AE5D959D3EA942B4EE41307D351AEE8F8BD
240721_15305FTNK_Final_AS_FILED.pdf	0BE0E77B60C269383A355C731EB73A1D29BC74C505F021296D3E8C037C271AED809054DB662ABD8C46170FBAE852B0D373F139F4BD3F3C4A4264FD360C6DBE80
240721_15305FTNK_IDS_2 of 3_AS_FILED.pdf	2657EC5B878A018B494FDB68BAD509DE252CF455F1B9176F222DEE9CAC9A6E093A84B0C83AF2851F6056751BD8C5B9197A

	BDD26C78FCE90E168D4CAD74B341F4
240721_15305FTNK_IDS_3 of 3_AS_FILED.pdf	C72FF7B99EA73A3B0F97B7A876D75240A5D22B0861073B7647BF484BEB9720985FF08A3A6E4149E70644AD15A5A0ADCF27D4E3F56315EB24D73C31A5A8D3C273
240721_15305FTNK_ForeignRef_Atsmon_WO2020079685A1_AS_FILED.pdf	79C0FBB4C9D8CC4EDFE1640FD235ACD61AEAD49795E39E723EB289CD5E71446337B51A0D1EBE35302E13C9A4ADA1ADACEA7D3D3AA710001B18654D5B282F7C87
240721_15305FTNK_ForeignRef_Gupta_WO2015089463A1_AS_FILED.pdf	A3906E63EFA160A7F0DF216C71175624C7FB6E65E105CC9D22189E73C207CBB8A92735035DDE5CF1112FABC45B8CD1FC70E415BCFC50E23649B5ECBBB11A8285
240721_15305FTNK_ForeignRef_Jubran_WO2014159150A1_AS_FILED.pdf	3A91E865DCCA1559C121979E741CA959BBEA2FFA2B55D6014382694F1E441C6640CAC3014111E2936EF082DB74927B60994708FD5EE5DD56CFA98A316C905825
240721_15305FTNK_ForeignRef_Theimer_CA2930026C_AS_FILED.pdf	981531A01130F94153F2A9A798F1089450F8097668599E11D6C0BEBC1E36ABF6D9E06A6218CFE6CE05B1AAB3F87FD6637E9EE6ADDEEE4D5EDF464633A06E19DA
240721_15305FTNK_NPL_Cui et al_Non-Intrusive...Monitor System..._AS_FILED.pdf	073D66FFC970DE394DE442FFE4738D53C7018CE863EEDAE630882A81D7BEFA95D917C3B587D418E5F4F0E8FD7E80A7AD7A6ED333F06FAD1914C6EC8153527967
240721_15305FTNK_NPL_Gedik et al_Elastic Scaling for...Processing_AS_FILED.pdf	A5195EF911562C7C1DDE539AEDEEB328E925904465C6433B5BD37782DD927C8C6A8BE3D7EDA9AAD13A77B37E76BB3865CE0B53C1B9A192C1C367D64CBE36671C
240721_15305FTNK_NPL_Ekelhart et al_Integrating attker behavior..._AS_FILED.pdf	F9B6873F89D487647CFD4C528C68B3C51751DB5B9C69EC1D11689C1D88DB9456E40E45858A3A81337CF3BE3B535CB019A11CCE8F23C281DC95F1282C8305368B
240721_15305FTNK_NPL_JPCert_Detect Lateral Movement...v2_AS_FILED.pdf	7FACC854B3043E69C804BB6EC6B8F349C31EE0F4659DAA4ACECABC46591144175AF718C207A8CA37A292A8435B637AAE7E85922345AF31C99F7AC83CA89BDED6
240721_15305FTNK_NPL_Kbar_Wireless	A05EDBF86575C4364DDEB5D382F7506A4BBEE06CC25F6C1B17416DC7D09AE85695E6C942958CFDE205DFE37B1847BB797

Netwrk...Authentication_AS_FILED.pdf	C9C9A3F2FB5710F7111DC3E9294F916
240721_15305FTNK_NPL_Kotenko et al_Cyber Attk Model..._AS_FILED.pdf	474501F556F97A5DEB39E1B9CF9DBBA13F47DDCA5B7E72EA2FD543C1FE9DD4DCF7EC5CB95634E637CC298A52B5EAC651290C5B1B57CFC1A18394D4AFEAC73055
240721_15305FTNK_NPL_Boukhtouta et al_Graph-Theoretic Character..._AS_FILED.pdf	2987BF2E3D89655A17DF97124B6CE6BF6B82FD89FAB3EC85CE7E1C21D715C17F9A28DBD4B95BD316573AA274EBE27E3289219EDEF3E0DE9BF0C7EA6057BD2882
240721_15305FTNK_NPL_Kumar et al_DFuse..._AS_FILED.pdf	7509E06FF5DB70F7FC342DAB409790C0A9E84392EABA7CD067391D781914AF8AAF6DB545ECFFFD4B3E37326190264 6D489D135B74530CF490E144C6F2C003628
240721_15305FTNK_NPL_Lu et al_Sybil Attk Detect..._AS_FILED.pdf	50BCECB75CF6E53B275CA2215B51AE4E0551E12165B8F480586DB4027BA64EBA0CB7CED6F4E46A3432FF4A6BDE4D25C6708FE2693C28336BB1694DAFEE5CD386
240721_15305FTNK_NPL_Microsoft_Mitigat P-t-H Attks..._AS_FILED.pdf	5DBF378ED829BD37378D56016ADBAF6DFE83AD652534551FC522EF21B7A9EC9D7C53F482AF29929CBBBB55F725EF28DA7FEDB927DE132FB05AD934A20CD566B1
240721_15305FTNK_NPL_Kiesling et al_Selecting Secur Cntrl Porfolios..._AS_FILED.pdf	B4B77AB2BF5689AD467710F7DBD3854ED0E73F82E8DBE031EBC57EFE401BF40AAADC4E78C5E084A60E60F2565E3314946ACB506B8CA760AA3780CD9B89DE3CD1
240721_15305FTNK_NPL_Patapanchala_Explr Secure Metrics...Graphs_AS_FILED.pdf	FAFD9FFD878DA3AAD5116E3123B6032AEB18D8B7ADEE8B11A38F7A2A88C49A33760397EF4489F32FAA4669FC18F71625FDCAB03BD05CF440418E2B9A26F1AF25
240721_15305FTNK_NPL_Yang et al_Attack Projection_AS_FILED.pdf	374439E160111251961 1C50DAC797DE2A6E4BE7050DFB65C89F7BDA9AD7D11256AB167D42EA21A4EC121E4DC8A3BE2A42020F24E2FAB479F212C78766B713F60
240721_15305FTNK_NPL_Shandilya_...Attk Graphs in Security Systems_AS_FILED.pdf	756E5FF0F85747BE9EBC00A3DB541A0FE14BAB2144EE8290327DEE0F633BC7BAACEE10A01CEC2F631A2E5A120A49FC3DF290FBF621C6C86578AB0CFB67C98EA9
240721_15305FTNK_NPL_Zar	D6E28D53C2234F3F156867F08910FFCEE6E1B7B7F40309AC6

gar et al_0 Knwldge Anomaly-Based...Threats_AS_FILED.pdf	5FB717ACC53659D14B1EABBD43F30C4305615E7F103BBB50AB6FF6428AAAB58C6F62690B9EA2858
240721_15305FTNK_NPL_Fisk et al_ Agile nd Scalable Analysis..._AS_FILED.pdf	29EB92A3D63FFFB85BF643AB65498378A89820E8169D68BFEE6FBE8EF9D40CDE6B49A19F00998F1CF8685BF5C5F0ED33E64F64892F28BBA12C791C98BC5D622C
240721_15305FTNK_NPL_Jajodia et al_Advncd Cyber Attk..._AS_FILED.pdf	670E3E507AB29D2F0F0E9339178CA786499FCF0F6858CFE9C5844980AB885F38B28AEF1C278822B16A62360F8F1D13FBD81F785DA2500EFF168B5F33ACDEA766
240721_15305FTNK_NPL_Pasqualetti et al_Attack Detection...Systems_AS_FILED.pdf	0E4ADF3F3D52080371A63C2EB98208A1B65177B47823BEAFDEDE56BCEF2C17613B8B575E57BB7236521748065E9C7F1EB8065C7623AE5CF4E343FAD5465E9727
240721_15305FTNK_NPL_Apache_Beam_AS_FILED.pdf	AC15D8165BBD9F1A4A46541DE09C6C693B1685C89C99A6DF12C9B95862454F43AFB49E82649A04BA12E54E3D93404D7CA720447DF3F39BBAA19526390F0475B9
240721_15305FTNK_NPL_Apache_Airflow_AS_FILED.pdf	26C4F799E49FDBA382717B1F0445F333FFC25C74F12AAAC279B3BB39B3C7B9D2A1DF918E54871E75D39A7724368BC2E53AA9C82D9F5980B2FF1E3F129EA4824C
240721_15305FTNK_NPL_Apache_Apex_AS_FILED.pdf	D6C20B096DC061B0CFA9BB79F1881B4A5168FD1CF5C6DD7A68F95F1295A72EF2F7BF41F762F7E2E98DF29AD634EBC9FFA2E51D8622DA899F8D245DAFF3ADF5C9
240721_15305FTNK_NPL_Apache_AS_FILED.pdf	04F3A849747A04945F50BA11D20B9959AF986453DB18A17EC8E4BE48C4C21BFBE3A7DC28DDFEBE155D90CF83AD44D6A8B7F300FA6D5DBC9CAF1EE0828ABD14EB
240721_15305FTNK_NPL_Apache_Ignite_AS_FILED.pdf	F53B1D2AD68F3F8E267985F299A37A06B483E0CCC2530F819FAA95A7E71A53CA99EA8C092E213EBBAD06334613FE00B4278676329D19C040ED81CBE819DADB05
240721_15305FTNK_NPL_Apache_Kafka_AS_FILED.pdf	39D2DC340CE86752914F26A86D44B692971FEA806D5BDE825CBE6E3335541C71DC2B64B9341793592CF566FB4101C70A04B5996F2EF0B196C16538394C8AE242

240721_15305FTNK_NPL_Apache_Flink_AS_FILED.pdf	C57B0F8A8327D1B7FFDEACF0D5A5969BC1D41F9E17F14167F977F4AD7B282EF318FC070D07B68A3F2FF0D009F9AA1FAA733F1960FE554A97EEA9760DC56C219F
240721_15305FTNK_NPL_Apache_Logging_Flume_AS_FILED.pdf	E44CDF0E05C55D1FD2194D7058EBCB14C9D47B0F1EFD538C81A5CF8116A2944F05CE7EA8D6D19D6EABE4BE8425FA92E3A3AB998ECAA53C1AC34F84625F95408E
240721_15305FTNK_NPL_Apache_Samza_AS_FILED.pdf	C7E9E0E376E53A19E9C580B94B179707246D0BAC3FA81F81D6638C08F19B4FFCF49257700FF3B434FB203E852E97B9DB1E68D26120CC72F0D32C1F8126155B91
240721_15305FTNK_NPL_Apache_Hadoop_AS_FILED.pdf	DB88F9C9727561C32DC612C22F54A18E4B1BE10D66020BC18F8237F379A9843A69D45188D635D9841C84E87D654DBE3ED79AEEC14DFA4111AC3D6B29B32A8D80
240721_15305FTNK_NPL_Apache_Storm_AS_FILED.pdf	288ECE893EEF6ADAE0C62388BF513331628CF5AE04EC3019947AF79E0928426CFD52FA0C2F284CC038C3DD4C01A53065E5FFB4EE9821EE7F6CD25E0101E7190F
240721_15305FTNK_NPL_Apache_Spark_AS_FILED.pdf	CAA50F0B77114155473D03969FA0CC267D14A98CE23EEF047545B45E5F2FDC2AA5C4733D5D3A289B9D17C447471D6BAAC396DAA2C390D668B5B3F4FCB818B6E1
240721_15305FTNK_IntCommAuth_AS_FILED.pdf	D4ED3C5E180476A52858F6550482405A90AE0BF352AF4375752EC2E0AF775F4671B07D141F9418D11436EB78E002E1203B8D3B214A35432D8077A2261707CCCF

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application.

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an

international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation of U.S. Pat. App. No. 18/581,375, filed February 20, 2024; which is a continuation of U.S. Pat. App. No. 17/189,161, filed March 1, 2021; which is a continuation-in-part of U.S. Pat. App. No. 17/061,195, filed October 1, 2020, now issued as U.S. Pat. No. 11,570,214 on January 31, 2023; which is a continuation-in-part of U.S. Pat. App. No. 17/035,029, filed September 28, 2020, now issued as U.S. Pat. No. 11,546,380 on January 3, 2023; which is a continuation-in-part of U.S. Pat. App. No. 17/008,276, filed August 31, 2020, now issued as U.S. Pat. No. 11,323,484 on May 3, 2022; which is a continuation-in-part of U.S. Pat. App. No. 17/000,504, filed August 24, 2020, now issued as U.S. Pat. No. 11,477,245 on October 18, 2022; which is a continuation-in-part of U.S. Pat. App. No. 16/855,724, filed on April 22, 2020, now issued as U.S. Pat. No. 11,218,510 on January 4, 2022; which is a continuation-in-part of U.S. Pat. App. No. 16/836,717, filed on March 31, 2020, now issued as U.S. Pat. No. 10,917,428 on February 9, 2021; which is a continuation-in-part of U.S. Pat. App. No. 15/887,496, filed February 2, 2018, now issued as U.S. Pat. No. 10,783,241 on September 22, 2020; which is a continuation-in-part of U.S. Pat. App. No. 15/823,285, filed November 27, 2017, now issued as U.S. Pat. No. 10,740,096 on August 11, 2020; which is a continuation-in-part of U.S. Pat. App. No. 15/788,718, filed October 19, 2017, now issued as U.S. Pat. No. 10,861,014 on December 8, 2020; which is a continuation-in-part of U.S. Pat. App. No. 15/788,002, filed on October 19, 2017; which is a continuation-in-part of U.S. Pat. App. No. 15/787,601, filed on October 18, 2017, now issued as U.S. Pat. No. 10,860,660 on December 8, 2020; which claims the benefit of U.S. Provisional Pat. App. No. 62/568,312, filed on October 4, 2017; said application number 15/787,601 is a continuation-in-part of U.S. Pat. App. No. 15/616,427, filed on June 7, 2017; which is a continuation-in-part of U.S. Pat. App. No. 14/925,974 (which is explicitly incorporated by reference in its entirety herein), filed October 28, 2015; said application number 15/788,002 claims the benefit of U.S. Provisional Pat. App. No. 62/568,305, filed October 4, 2017; said application number 15/788,718 claims the benefit of U.S. Provisional Pat. App. No. 62/568,307, filed October 4, 2017; said application number 15/887,496 is a continuation-in-part of U.S. Pat. App. No. 15/818,733, filed November 20, 2017, now issued as U.S. Pat. No. 10,673,887 on June 2, 2020; which is a continuation-in-part of U.S. Pat. App. No.

15/725,274, filed October 4, 2017, now issued as U.S. Pat. No. 10,609,079 on March 31, 2020; which is a continuation-in-part of U.S. Pat. App. No. 15/655,113, filed July 20, 2017, now issued as U.S. Pat. No. 10,735,456 on August 4, 2020; which is a continuation-in-part of U.S. Pat. App. No. 15/616,427, filed June 7, 2017; said application number 15/655,113 is a continuation-in-part of U.S. Pat. App. No. 15/237,625, filed August 15, 2016, now issued as U.S. Pat. No. 10,248,910 on April 2, 2019; which is a continuation-in-part of U.S. Pat. App. No. 15/206,195, filed July 8, 2016; which is a continuation-in-part of U.S. Pat. App. No. 15/186,453, filed June 18, 2016; which is a continuation-in-part of U.S. Pat. App. No. 15/166,158, filed May 26, 2016; which is a continuation-in-part of U.S. Pat. App. No. 15/141,752, filed April 28, 2016, now issued as U.S. Pat. No. 10,860,962 on December 8, 2020; which is a continuation-in-part of U.S. Pat. App. No. 15/091,563, filed April 5, 2016, now issued as U.S. Pat. No. 10,204,147 on February 12, 2019; said application number 15/141,752 is a continuation-in-part of U.S. Pat. App. No. 14/986,536, filed December 31, 2015, now issued as U.S. Pat. No. 10,210,255 on February 19, 2019; said application number 15/141,752 is a continuation-in-part of U.S. Pat. App. No. 14/925,974, filed October 28, 2015; said application number 16/855,724 is a continuation-in-part of U.S. Pat. App. No. 16/777,270, filed January 30, 2020, now issued as U.S. Pat. No. 11,025,674 on June 1, 2021; which is a continuation-in-part of U.S. Pat. App. No. 16/720,383, filed December 19, 2019, now issued as U.S. Pat. No. 10,944,795 on March 9, 2021; which is a continuation of U.S. Pat. App. No. 15/823,363, filed November 27, 2017, now issued as U.S. Pat. No. 10,560,483 on February 11, 2020; which is a continuation-in-part of U.S. Pat. App. No. 15/725,274, filed October 4, 2017, now issued as U.S. Pat. No. 10,609,079 on March 31, 2020; said application number 17/000,504 is a continuation-in-part of U.S. Pat. App. No. 16/412,340, filed May 14, 2019, now issued as U.S. Pat. No. 11,539,663 on December 27, 2022; which is a continuation-in-part of U.S. Pat. App. No. 16/267,893, filed February 5, 2019; which is a continuation-in-part of U.S. Pat. App. No. 16/248,133, filed January 15, 2019; which is a continuation-in-part of U.S. Pat. App. No. 15/849,901, filed December 21, 2017, now issued as U.S. Pat. No. 11,023,284 on June 1, 2021; which is a continuation-in-part of U.S. Pat. App. No. 15/835,436, filed December 17, 2017, now issued as U.S. Pat. No. 10,572,828 on February 5, 2020; which is a continuation-in-part of U.S. Pat. App. No. 15/790,457, filed October 23, 2017, now issued as U.S. Pat. No. 10,884,999 on January 5, 2021; which is a continuation-in-part of U.S. Pat. App. No. 15/790,327, filed on October 23, 2017, now issued as U.S. Pat. No. 10,860,951 on December 8, 2020; which claims the benefit of U.S. Provisional Pat. App. No. 62/568,291, filed

October 4, 2017; said application number 15/790,327 is a continuation-in-part of U.S. Pat. App. No. 15/616,427, filed June 7, 2017; said application number 15/790,327 is a continuation-in-part of U.S. Pat. App. No. 15/141,752, filed April 28, 2016, now issued as U.S. Pat. No. 10,860,962 on December 8, 2020; said application number 15/790,457 claims the benefit of U.S. Provisional Pat. App. No. 62/568,298, filed October 4, 2017; said application number 15/849,901 is a continuation-in-part of U.S. Pat. App. No. 15/835,312, filed December 7, 2017, now issued as U.S. Pat. No. 11,055,451 on July 6, 2021; which is a continuation-in-part of U.S. Pat. App. No. 15/186,453, filed June 18, 2016; said application number 16/248,133 is a continuation-in-part of U.S. Pat. App. No. 15/813,097, filed November 14, 2017; which is a continuation-in-part of 15/616,427, filed June 7, 2017; said application number 16/248,133 is a continuation-in-part of U.S. Pat. App. No. 15/806,697, filed November 8, 2017; which is a continuation-in-part of U.S. Pat. App. No. 15/376,657, filed December 13, 2016, now issued as U.S. Pat. No. 10,402,906 on September 3, 2019; which is a continuation-in-part of U.S. Pat. App. No. 15/237,625, filed August 15, 2016, now issued as U.S. Pat. No. 10,248,910 on April 2, 2019; said application number 15/806,697 is a continuation-in-part of U.S. Pat. App. No. 15/343,209, filed November 4, 2016, now issued as U.S. Pat. No. 11,087,403 on August 10, 2021; which is a continuation-in-part of U.S. Pat. App. No. 15/237,625, filed August 15, 2016, now issued as U.S. Pat. No. 10,248,910 on April 2, 2019; said application number 15/343,209 is a continuation-in-part of U.S. Pat. App. No. 15/229,476, filed August 5, 2016, now issued as U.S. Pat. No. 10,454,791 on October 22, 2019; which is a continuation-in-part of U.S. Pat. App. No. 15/206,195, filed July 8, 2016; said application number 16/248,133 is a continuation-in-part of U.S. Pat. App. No. 15/673,368, filed August 9, 2017; which is a continuation-in-part of U.S. Pat. App. No. 15/376,657, filed December 13, 2016, now issued as U.S. Pat. No. 10,402,906 on September 3, 2019; said application number 17/061,195 is a continuation-in-part of U.S. Pat. App. No. 15/879,801, filed January 25, 2018; which is a continuation-in-part of U.S. Pat. App. No. 15/379,899, filed December 15, 2016; which is a continuation-in-part of U.S. Pat. App. No. 15/376,657, filed December 13, 2016, now issued as U.S. Pat. No. 10,402,906 on September 3, 2019; said application number 17/189,161 is a continuation-in-part of U.S. Pat. App. No. 16/709,598, filed December 10, 2019, now issued as U.S. Pat. No. 11,507,858 on November 22, 2022; which is a continuation-in-part of U.S. Pat. App. No. 14/925,974, filed October 28, 2015.

BACKGROUND OF THE INVENTION

Field of the Invention

[0002] The present invention is in the field of analysis of very large data sets using distributed computational graph tools which allow for transformation of data through both linear and non-linear transformation pipelines.

Discussion of the State of the Art

[0003] The ability to transfer information between individuals, even over large distances, is credited with allowing mankind to rise from a species of primate gatherer-scavengers to forming simple communities. The ability to stably record information so that it could be analyzed for repetitive events, trends, and serve as a base to be expanded and built upon. It is safe to say that the availability of information in formats that allow it to be analyzed and added to by both individuals contemporary to its accrual and those who come after is the most powerful tool available to mankind and likely is what has propelled us to the level of social and technological achievement we have attained.

[0004] Nothing has augmented our ability to gather and store information analogous to the rise of electronic and computer technology. There are sensors of all types to measure just about any condition one can imagine. Computers have allowed the health information for a large portion of the human population is stored and accessible. Similarly, detailed data on vehicular accidents, both environmental and vehicle component factors Airline mishaps and crashes can be recreated and studied in great detail. Item information is recorded for the majority of consumer purchases. Further examples abound, but the point has been made. Computer database technology has allowed all of this information to be reliably stored for future retrieval and analysis. The benefits of database technology are so strong that there are very few businesses large or small that do not make some use of a data and knowledge storage solution, either directly for such tasks as inventory control and forecasting or customer relations, or indirectly for ordering. The meteoric rise of computer networking the internet has only served to turn the accrual of information into a torrent as now huge populations can exchange observations, data and ideas, even invited to do so; vast arrays of sensors can be tied together in meaningful ways all of which can be stored for future analysis and use. The receipt and storage of data has gotten to the point where an expert has been quoted as estimating that as much data is currently accrued in two days as was accrued in all history prior to 2003 (Eric Schmidt, Google). Entirely new distributed data storage and retrieval technologies such as Hadoop, and map/reduce; and graph and column based data store organization have been developed to accommodate the influx of information and provide some ability to retrieve information in a guided

fashion, but such retrieval has proven to be too labor intensive and rigid to be of use in all but the more superficial and simple of campaigns. Presently, we are accruing vast amounts of information daily but do not have the tools to analyze all but a trickle into knowledge or informed action. What is needed is a system to allow the analysis of current, possibly complex and changing streaming data of interest in the context of the vast stored data that has accumulated relating to it such that meaningful conclusions made and effective action can be taken. To be of use, such a system would also need to possess the ability to self-assess its own operations and key intermediate factors in both the data stream and stored information and make changes to its own function to optimize function and maximize the probability of reliable conclusions.

[0005] Data pipelines, which are a progression of functions which each perform some action or transformation on a data stream, offer a mechanism to process quantities of data in the volume discussed directly above. To date however, data pipelines have either been extremely limited in what they do, for example "move data from a web based merchant site to a distributed data store; extract all purchases and classify by product type and region; store the result logs" or have been rigidly programmed and possibly required the uses of highly specific remote protocol calls to perform needed tasks. Even with these additions their capabilities have been very limited and, they have all been linear in configuration which precludes their use for analysis and conclusion or action discovery in a majority of complex situations where branching or even recurrent modification is needed.

[0006] What is needed is a system that intelligently combines processing of a current data stream with the ability to retrieve relevant stored data in such a way that conclusions or actions could be drawn in a predictive manner. To work in a timely and efficient manner, the system needs the ability to monitor for both operational issues within its components and should be able to learn and react to intermediate determinations of the analyses it runs and also should be able to self-modify to maintain optimal operation.

SUMMARY OF THE INVENTION

[0007] The inventor has developed a system for rapid predictive analysis of very large data sets using a distributed computational graph, that intelligently combines processing of a current data stream with the ability to retrieve relevant stored data in such a way that conclusions or actions could be drawn in a predictive manner.

[0008] According to a preferred embodiment of the invention, a system for rapid predictive analysis of very large data sets using the distributed computational graph, comprising a data receipt software module, a data filter software module, a data formalization software module, an input event data store module, a batch event analysis server, a system sanity and retrain software module, a messaging software module, a transformation pipeline software module, and an output software module, is disclosed. The data receipt software module: receives streams of input from one or more of a plurality of data sources, and sends the data stream to the data filter module. The filter software module: receives streams of data from the data receipt software module; removes data records from the stream for a plurality of reasons drawn from, but not limited to, a set comprising absence of all information, damage to data in the record, and presence of in-congruent information or missing information which invalidates the data record; splits filtered data stream into two or more identical parts; sends one identical data stream to the data formalization software module; and sends another identical data stream to the transformation pipeline module of the distributed graph computational module. The data formalization module: receives data stream from the data filter software module; formats the data within data stream based upon a set of predetermined parameters so as to prepare for meaningful storage in a data store; and places the formatted data stream into the input event data store. The input event data store: receives properly formatted data from the data formalization module; and stores the data by method suited to the long term availability, timely retrieval, and analysis of the accumulated data; The batch event analysis server: accesses the data store for information of interest based upon a set of predetermined parameters; aggregates data retrieved from the data store as predetermined that represent such interests as trends of importance, past instances of an event or set of events within a system under analysis or possible cause and effect relationships between two or more variables over many iterations; and provides summary information based upon the breadth of the data analyzed to the messaging software module; and receives communication from the messaging software module which may be in the form of requests for particular information or directives concerning the information being supplied at that time. The transformation pipeline software module: receives streaming data from the data filter software module; performs one or more functions on data within data stream; provides data resultant from the set of function pipeline back to the system; and receives directives from the system sanity and retrain module to modify the function of the pipeline. The messaging software module: receives administrative directives from those conducting the analysis; receives data store analysis summaries from batch event analysis

server; receives results of pipeline data functions from transformation pipeline software module; and sends data analysis status and progress related messages as well as administrative execution directives to the system sanity and retrain software module. The system sanity and retrain software module: receives data analysis status and progress information from the messaging software module; compares all incoming information against preassigned parameters to ensure system stability; changes operational behavior within other software modules of system using preexisting guidelines to return required system function; sends alert signal through the output module concerning degraded system status as necessary; and receives and applies any administrative requests for changes in system function. Finally, the output module: receives information destined for outside of the system; formats that information based upon designated end target; and routes that information to the proper port for intended further action.

[0009] According to another preferred embodiment of the invention, a method for a system for the predictive analysis of very large data sets using the distributed computational graph, the method comprising the following steps: To receive streaming input from one or more of a plurality of data sources. To filter data of incomplete, misconfigured or damaged input. To formalize input data for use in batch and streaming portions of method using pre-designed standard. To perform a set of one or more data transformations on formalized input. To perform sanity checks of results of transformation pipeline analysis of streaming data as well as analysis process retraining based upon batch analysis of input data. Finally, to output the results of the analysis process in format predecided upon by the authors of the analysis.

BRIEF DESCRIPTION OF THE DRAWING FIGURES

[0010] The accompanying drawings illustrate several embodiments of the invention and, together with the description, serve to explain the principles of the invention according to the embodiments. One skilled in the art will recognize that the particular embodiments illustrated in the drawings are merely exemplary, and are not intended to limit the scope of the present invention.

[0011] Fig. 1 is a block diagram illustrating an exemplary hardware architecture of a computing device used in various embodiments of the invention.

[0012] Fig. 2 is a block diagram illustrating an exemplary logical architecture for a client device, according to various embodiments of the invention.

[0013] Fig. 3 is a block diagram illustrating an exemplary architectural arrangement of clients, servers, and external services, according to various embodiments of the invention.

[0014] Fig. 4 is a block diagram illustrating an exemplary overview of a computer system as may be used in any of the various locations throughout the system

[0015] Fig. 5 is a diagram of an exemplary architecture for a system where streams of input data from one or more of a plurality of sources are analyzed to predict outcome using both batch analysis of acquired data and transformation pipeline manipulation of current streaming data according to an embodiment of the invention.

[0016] Fig. 6 (Prior Art) is a diagram of an exemplary architecture for a linear transformation pipeline system which introduces the concept of the transformation pipeline as a directed graph of transformation nodes and messages according to an embodiment of the invention.

[0017] Fig. 7 is a diagram of an exemplary architecture for a transformation pipeline system where one of the transformations receives input from more than one source which introduces the concept of the transformation pipeline as a directed graph of transformation nodes and messages according to an embodiment of the invention.

[0018] Fig. 8 is a diagram of an exemplary architecture for a transformation pipeline system where the output of one data transformation servers as the input of more than one downstream transformations which introduces the concept of the transformation pipeline as a directed graph of transformation nodes and messages according to an embodiment of the invention.

[0019] Fig. 9 is a diagram of an exemplary architecture for a transformation pipeline system where a set of three data transformations act to form a cyclical pipeline which also introduces the concept of the transformation pipeline as a directed graph of transformation nodes and messages according to an embodiment of the invention.

[0020] Fig. 10 is a process flow diagram of a method for the receipt, processing and predictive analysis of streaming data using a system of the invention.

[0021] Fig. 11 is a process flow diagram of a method for representing the operation of the transformation pipeline as a directed graph function using a system of the invention.

[0022] Fig. 12 (Prior Art) is a process flow diagram of a method for a linear data transformation pipeline using a system of the invention.

[0023] Fig. 13 is a process flow diagram of a method for the disposition of input from two antecedent data transformations into a single data transformation of transformation pipeline using a system of the invention.

[0024] Fig. 14 is a process flow diagram of a method for the disposition of output of one data transformation that then serves as input to two postliminary data transformations using a system of the invention.

[0025] Fig. 15 is a process flow diagram of a method for processing a set of three or more data transformations within a data transformation pipeline where output of the last member transformation of the set serves as input of the first member transformation thereby creating a cyclical relationship using a system of the invention.

[0026] Fig. 16 is a process flow diagram of a method for the receipt and use of streaming data into batch storage and analysis of changes over time, repetition of specific data sequences or the presence of critical data points using a system of the invention.

DETAILED DESCRIPTION OF THE INVENTION

[0027] The inventor has conceived, and reduced to practice, various systems and methods for predictive analysis of very large data sets using a distributed computational graph.

[0028] One or more different inventions may be described in the present application. Further, for one or more of the inventions described herein, numerous alternative embodiments may be described; it should be understood that these are presented for illustrative purposes only. The described embodiments are not intended to be limiting in any sense. One or more of the inventions may be widely applicable to numerous embodiments, as is readily apparent from the disclosure. In general, embodiments are described in sufficient detail to enable those skilled in the art to practice one or more of the inventions, and it is to be understood that other embodiments may be utilized and that structural, logical, software, electrical and other changes may be made without departing from the scope of the particular inventions. Accordingly, those skilled in the art will recognize that one or more of the inventions may be practiced with various modifications and alterations. Particular features of one or more of the inventions may be described with reference to one or more particular embodiments or figures that form a part of the present disclosure, and in which are shown, by way of illustration, specific embodiments of one or more of the inventions. It should be understood, however, that such features are not limited to usage in the one or more particular embodiments or

figures with reference to which they are described. The present disclosure is neither a literal description of all embodiments of one or more of the inventions nor a listing of features of one or more of the inventions that must be present in all embodiments.

[0029] Headings of sections provided in this patent application and the title of this patent application are for convenience only, and are not to be taken as limiting the disclosure in any way.

[0030] Devices that are in communication with each other need not be in continuous communication with each other, unless expressly specified otherwise. In addition, devices that are in communication with each other may communicate directly or indirectly through one or more intermediaries, logical or physical.

[0031] A description of an embodiment with several components in communication with each other does not imply that all such components are required. To the contrary, a variety of optional components may be described to illustrate a wide variety of possible embodiments of one or more of the inventions and in order to more fully illustrate one or more aspects of the inventions. Similarly, although process steps, method steps, algorithms or the like may be described in a sequential order, such processes, methods and algorithms may generally be configured to work in alternate orders, unless specifically stated to the contrary. In other words, any sequence or order of steps that may be described in this patent application does not, in and of itself, indicate a requirement that the steps be performed in that order. The steps of described processes may be performed in any order practical. Further, some steps may be performed simultaneously despite being described or implied as occurring sequentially (e.g., because one step is described after the other step). Moreover, the illustration of a process by its depiction in a drawing does not imply that the illustrated process is exclusive of other variations and modifications thereto, does not imply that the illustrated process or any of its steps are necessary to one or more of the invention(s), and does not imply that the illustrated process is preferred. Also, steps are generally described once per embodiment, but this does not mean they must occur once, or that they may only occur once each time a process, method, or algorithm is carried out or executed. Some steps may be omitted in some embodiments or some occurrences, or some steps may be executed more than once in a given embodiment or occurrence.

[0032] When a single device or article is described, it will be readily apparent that more than one device or article may be used in place of a single device or article. Similarly, where more than one device or article is described, it will be readily apparent that a single device or article may be used in place of the more than one device or article.

[0033] The functionality or the features of a device may be alternatively embodied by one or more other devices that are not explicitly described as having such functionality or features. Thus, other embodiments of one or more of the inventions need not include the device itself.

[0034] Techniques and mechanisms described or referenced herein will sometimes be described in singular form for clarity. However, it should be noted that particular embodiments include multiple iterations of a technique or multiple manifestations of a mechanism unless noted otherwise. Process descriptions or blocks in figures should be understood as representing modules, segments, or portions of code which include one or more executable instructions for implementing specific logical functions or steps in the process. Alternate implementations are included within the scope of embodiments of the present invention in which, for example, functions may be executed out of order from that shown or discussed, including substantially concurrently or in reverse order, depending on the functionality involved, as would be understood by those having ordinary skill in the art.

Definitions

[0035] As used herein, "graph" is a representation of information and relationships, where each primary unit of information makes up a "node" or "vertex" of the graph and the relationship between two nodes makes up an edge of the graph. Nodes can be further qualified by the connection of one or more descriptors or "properties" to that node. For example, given the node "James R," name information for a person, qualifying properties might be "183 cm tall", "DOB 08/13/1965" and "speaks English". Similar to the use of properties to further describe the information in a node, a relationship between two nodes that forms an edge can be qualified using a "label". Thus, given a second node "Thomas G," an edge between "James R" and "Thomas G" that indicates that the two people know each other might be labeled "knows." When graph theory notation (Graph = (Vertices, Edges)) is applied this situation, the set of nodes are used as one parameter of the ordered pair,V and the set of 2 element edge endpoints are used as the second parameter of the ordered pair, E. When the order of the edge endpoints within the pairs of E is not significant, for example, the edge James R, Thomas G is equivalent to Thomas G, James R, the graph is designated as "undirected." Under circumstances when a relationship flows from one node to another in one direction, for example James R is "taller" than Thomas G, the order of the endpoints is significant. Graphs with such edges are designated as "directed." In the distributed computational graph system, transformations within transformation pipeline are represented as directed graph with each transformation comprising a node and the output messages between transformations comprising edges. Distributed computational

graph stipulates the potential use of non-linear transformation pipelines which are programmatically linearized. Such linearization can result in exponential growth of resource consumption. The most sensible approach to overcome possibility is to introduce new transformation pipelines just as they are needed, creating only those that are ready to compute. Such method results in transformation graphs which are highly variable in size and node, edge composition as the system processes data streams. Those familiar with the art will realize that transformation graph may assume many shapes and sizes with a vast topography of edge relationships. The examples given were chosen for illustrative purposes only and represent a small number of the simplest of possibilities. These examples should not be taken to define the possible graphs expected as part of operation of the invention

[0036] As used herein, "transformation" is a function performed on zero or more streams of input data which results in a single stream of output which may or may not then be used as input for another transformation. Transformations may comprise any combination of machine, human or machine-human interactions Transformations need not change data that enters them, one example of this type of transformation would be a storage transformation which would receive input and then act as a queue for that data for subsequent transformations. As implied above, a specific transformation may generate output data in the absence of input data. A time stamp serves as a example. In the invention, transformations are placed into pipelines such that the output of one transformation may serve as an input for another. These pipelines can consist of two or more transformations with the number of transformations limited only by the resources of the system. Historically, transformation pipelines have been linear with each transformation in the pipeline receiving input from one antecedent and providing output to one subsequent with no branching or iteration. Other pipeline configurations are possible. The invention is designed to permit several of these configurations including, but not limited to: linear, afferent branch, efferent branch and cyclical.

[0037] A "database" or "data storage subsystem" (these terms may be considered substantially synonymous), as used herein, is a system adapted for the long-term storage, indexing, and retrieval of data, the retrieval typically being via some sort of querying interface or language. "Database" may be used to refer to relational database management systems known in the art, but should not be considered to be limited to such systems. Many alternative database or data storage system technologies have been, and indeed are being, introduced in the art, including but not limited to

distributed non-relational data storage systems such as Hadoop, column-oriented databases, in-memory databases, and the like. While various embodiments may preferentially employ one or another of the various data storage subsystems available in the art (or available in the future), the invention should not be construed to be so limited, as any data storage architecture may be used according to the embodiments. Similarly, while in some cases one or more particular data storage needs are described as being satisfied by separate components (for example, an expanded private capital markets database and a configuration database), these descriptions refer to functional uses of data storage systems and do not refer to their physical architecture. For instance, any group of data storage systems of databases referred to herein may be included together in a single database management system operating on a single machine, or they may be included in a single database management system operating on a cluster of machines as is known in the art. Similarly, any single database (such as an expanded private capital markets database) may be implemented on a single machine, on a set of machines using clustering technology, on several machines connected by one or more messaging systems known in the art, or in a master/slave arrangement common in the art. These examples should make clear that no particular architectural approaches to database management is preferred according to the invention, and choice of data storage technology is at the discretion of each implementer, without departing from the scope of the invention as claimed.

Hardware Architecture

[0038] Generally, the techniques disclosed herein may be implemented on hardware or a combination of software and hardware. For example, they may be implemented in an operating system kernel, in a separate user process, in a library package bound into network applications, on a specially constructed machine, on an application-specific integrated circuit (ASIC), or on a network interface card.

[0039] Software/hardware hybrid implementations of at least some of the embodiments disclosed herein may be implemented on a programmable network-resident machine (which should be understood to include intermittently connected network-aware machines) selectively activated or reconfigured by a computer program stored in memory. Such network devices may have multiple network interfaces that may be configured or designed to utilize different types of network communication protocols. A general architecture for some of these machines may be disclosed herein in order to illustrate one or more exemplary means by which a given unit of functionality may be implemented. According to specific embodiments, at least some of the features or functionalities

of the various embodiments disclosed herein may be implemented on one or more general-purpose computers associated with one or more networks, such as for example an end-user computer system, a client computer, a network server or other server system possibly networked with others in a data processing center, a mobile computing device (e.g., tablet computing device, mobile phone, smartphone, laptop, and the like), a consumer electronic device, a music player, or any other suitable electronic device, router, switch, or the like, or any combination thereof. In at least some embodiments, at least some of the features or functionalities of the various embodiments disclosed herein may be implemented in one or more virtualized computing environments (e.g., network computing clouds, virtual machines hosted on one or more physical computing machines, or the like).

[0040] Referring now to **Fig. 1**, there is shown a block diagram depicting an exemplary computing device **100** suitable for implementing at least a portion of the features or functionalities disclosed herein. Computing device **100** may be, for example, any one of the computing machines listed in the previous paragraph, or indeed any other electronic device capable of executing software- or hardware-based instructions according to one or more programs stored in memory. Computing device **100** may be adapted to communicate with a plurality of other computing devices, such as clients or servers, over communications networks such as a wide area network a metropolitan area network, a local area network, a wireless network, the Internet, or any other network, using known protocols for such communication, whether wireless or wired.

[0041] In one embodiment, computing device **100** includes one or more central processing units (CPU) **102**, one or more interfaces **110**, and one or more buses **106** (such as a peripheral component interconnect (PCI) bus). When acting under the control of appropriate software or firmware, CPU **102** may be responsible for implementing specific functions associated with the functions of a specifically configured computing device or machine. For example, in at least one embodiment, a computing device **100** may be configured or designed to function as a server system utilizing CPU **102**, local memory **101** and/or remote memory **120**, and interface(s) **110**. In at least one embodiment, CPU **102** may be caused to perform one or more of the different types of functions and/or operations under the control of software modules or components, which for example, may include an operating system and any appropriate applications software, drivers, and the like.

[0042] CPU **102** may include one or more processors **103** such as, for example, a processor from one of the Intel, ARM, Qualcomm, and AMD families of microprocessors. In some embodiments,

processors **103** may include specially designed hardware such as application-specific integrated circuits (ASICs), electrically erasable programmable read-only memories (EEPROMs), field-programmable gate arrays (FPGAs), and so forth, for controlling operations of computing device **100**. In a specific embodiment, a local memory **101** (such as non-volatile random access memory (RAM) and/or read-only memory (ROM), including for example one or more levels of cached memory) may also form part of CPU **102**. However, there are many different ways in which memory may be coupled to system **100**. Memory **101** may be used for a variety of purposes such as, for example, caching and/or storing data, programming instructions, and the like.

[0043] As used herein, the term "processor" is not limited merely to those integrated circuits referred to in the art as a processor, a mobile processor, or a microprocessor, but broadly refers to a microcontroller, a microcomputer, a programmable logic controller, an application-specific integrated circuit, and any other programmable circuit.

[0044] In one embodiment, interfaces **110** are provided as network interface cards (NICs). Generally, NICs control the sending and receiving of data packets over a computer network; other types of interfaces **110** may for example support other peripherals used with computing device **100**. Among the interfaces that may be provided are Ethernet interfaces, frame relay interfaces, cable interfaces, DSL interfaces, token ring interfaces, graphics interfaces, and the like. In addition, various types of interfaces may be provided such as, for example, universal serial bus (USB), Serial, Ethernet, Firewire, PCI, parallel, radio frequency (RF), Bluetooth, near-field communications (e.g., using near-field magnetics), 802.11 (WiFi), frame relay, TCP/IP, ISDN, fast Ethernet interfaces, Gigabit Ethernet interfaces, asynchronous transfer mode (ATM) interfaces, high-speed serial interface (HSSI) interfaces, Point of Sale (POS) interfaces, fiber data distributed interfaces (FDDIs), and the like. Generally, such interfaces 110 may include ports appropriate for communication with appropriate media. In some cases, they may also include an independent processor and, in some instances, volatile and/or non-volatile memory (e.g., RAM).

[0045] Although the system shown in **Fig. 1** illustrates one specific architecture for a computing device **100** for implementing one or more of the inventions described herein, it is by no means the only device architecture on which at least a portion of the features and techniques described herein may be implemented. For example, architectures having one or any number of processors **103** may be used, and such processors **103** may be present in a single device or distributed among any number of devices. In one embodiment, a single processor **103** handles communications as well as routing

computations, while in other embodiments a separate dedicated communications processor may be provided. In various embodiments, different types of features or functionalities may be implemented in a system according to the invention that includes a client device (such as a tablet device or smartphone running client software) and server systems (such as a server system described in more detail below).

[0046] Regardless of network device configuration, the system of the present invention may employ one or more memories or memory modules (such as, for example, remote memory block **120** and local memory **101**) configured to store data, program instructions for the general-purpose network operations, or other information relating to the functionality of the embodiments described herein (or any combinations of the above). Program instructions may control execution of or comprise an operating system and/or one or more applications, for example. Memory **120** or memories **101**, **120** may also be configured to store data structures, configuration data, encryption data, historical system operations information, or any other specific or generic non-program information described herein.

[0047] Because such information and program instructions may be employed to implement one or more systems or methods described herein, at least some network device embodiments may include nontransitory machine-readable storage media, which, for example, may be configured or designed to store program instructions, state information, and the like for performing various operations described herein. Examples of such nontransitory machine- readable storage media include, but are not limited to, magnetic media such as hard disks, floppy disks, and magnetic tape; optical media such as CD- ROM disks; magneto-optical media such as optical disks, and hardware devices that are specially configured to store and perform program instructions, such as read-only memory devices (ROM), flash memory, solid state drives, memristor memory, random access memory (RAM), and the like. Examples of program instructions include both object code, such as may be produced by a compiler, machine code, such as may be produced by an assembler or a linker, byte code, such as may be generated by for example a Java compiler and may be executed using a Java virtual machine or equivalent, or files containing higher level code that may be executed by the computer using an interpreter (for example, scripts written in Python, Perl, Ruby, Groovy, or any other scripting language).

[0048] In some embodiments, systems according to the present invention may be implemented on a standalone computing system. Referring now to **Fig. 2**, there is shown a block diagram depicting a typical exemplary architecture of one or more embodiments or components thereof on a standalone

computing system. Computing device **200** includes processors **210** that may run software that carry out one or more functions or applications of embodiments of the invention, such as for example a client application **230**. Processors **210** may carry out computing instructions under control of an operating system **220** such as, for example, a version of Microsoft's Windows operating system, Apple's Mac OS/X or iOS operating systems, some variety of the Linux operating system, Google's Android operating system, or the like. In many cases, one or more shared services **225** may be operable in system **200**, and may be useful for providing common services to client applications **230**. Services **225** may for example be Windows services, user-space common services in a Linux environment, or any other type of common service architecture used with operating system **210**. Input devices **270** may be of any type suitable for receiving user input, including for example a keyboard, touchscreen, microphone (for example, for voice input), mouse, touchpad, trackball, or any combination thereof. Output devices **260** may be of any type suitable for providing output to one or more users, whether remote or local to system **200**, and may include for example one or more screens for visual output, speakers, printers, or any combination thereof. Memory **240** may be random-access memory having any structure and architecture known in the art, for use by processors **210**, for example to run software. Storage devices **250** may be any magnetic, optical, mechanical, memristor, or electrical storage device for storage of data in digital form. Examples of storage devices **250** include flash memory, magnetic hard drive, CD-ROM, and/or the like.

[0049] In some embodiments, systems of the present invention may be implemented on a distributed computing network, such as one having any number of clients and/or servers. Referring now to **Fig. 3**, there is shown a block diagram depicting an exemplary architecture for implementing at least a portion of a system according to an embodiment of the invention on a distributed computing network. According to the embodiment, any number of clients **330** may be provided. Each client **330** may run software for implementing client-side portions of the present invention; clients may comprise a system **200** such as that illustrated in **Fig. 2**. In addition, any number of servers **320** may be provided for handling requests received from one or more clients **330**. Clients **330** and servers **320** may communicate with one another via one or more electronic networks **310**, which may be in various embodiments of the Internet, a wide area network, a mobile telephony network, a wireless network (such as WiFi, Wimax, and so forth), or a local area network (or indeed any network topology known in the art; the invention does not prefer any one network topology over any other).

Networks **310** may be implemented using any known network protocols, including for example wired and/or wireless protocols.

[0050] In addition, in some embodiments, servers **320** may call external services **370** when needed to obtain additional information, or to refer to additional data concerning a particular call. Communications with external services **370** may take place, for example, via one or more networks **310**. In various embodiments, external services **370** may comprise web-enabled services or functionality related to or installed on the hardware device itself. For example, in an embodiment where client applications **230** are implemented on a smartphone or other electronic device, client applications **230** may obtain information stored in a server system **320** in the cloud or on an external service 370 deployed on one or more of a particular enterprise's or user's premises.

[0051] In some embodiments of the invention, clients **330** or servers **320** (or both) may make use of one or more specialized services or appliances that may be deployed locally or remotely across one or more networks **310**. For example, one or more databases **340** may be used or referred to by one or more embodiments of the invention. It should be understood by one having ordinary skill in the art that databases **340** may be arranged in a wide variety of architectures and using a wide variety of data access and manipulation means. For example, in various embodiments one or more databases **340** may comprise a relational database system using a structured query language (SQL), while others may comprise an alternative data storage technology such as those referred to in the art as "NoSQL" (for example, Hadoop, MapReduce, BigTable, and so forth). In some embodiments variant database architectures such as column-oriented databases, in-memory databases, clustered databases, distributed databases, key-value stores, or even flat file data repositories may be used according to the invention. It will be appreciated by one having ordinary skill in the art that any combination of known or future database technologies may be used as appropriate, unless a specific database technology or a specific arrangement of components is specified for a particular embodiment herein. Moreover, it should be appreciated that the term "database" as used herein may refer to a physical database machine, a cluster of machines acting as a single database system, or a logical database within an overall database management system. Unless a specific meaning is specified for a given use of the term "database", it should be construed to mean any of these senses of the word, all of which are understood as a plain meaning of the term "database" by those having ordinary skill in the art.

[0052] Similarly, most embodiments of the invention may make use of one or more security systems **360** and configuration systems **350**. Security and configuration management are common information technology (IT) and web functions, and some amount of each are generally associated with any IT or web systems. It should be understood by one having ordinary skill in the art that any configuration or security subsystems known in the art now or in the future may be used in conjunction with embodiments of the invention without limitation, unless a specific security **360** or configuration **350** system or approach is specifically required by the description of any specific embodiment.

[0053] **Fig. 4** shows an exemplary overview of a computer system **400** as may be used in any of the various locations throughout the system. It is exemplary of any computer that may execute code to process data. Various modifications and changes may be made to computer system **400** without departing from the broader scope of the system and method disclosed herein. CPU **401** is connected to bus **402**, to which bus is also connected memory **403**, nonvolatile memory **404**, display **407**, I/O unit **408**, and network interface card (NIC) **413**. I/O unit **408** may, typically, be connected to keyboard **409**, pointing device **410**, hard disk **412**, and real-time clock **411**. NIC **413** connects to network **414**, which may be the Internet or a local network, which local network may or may not have connections to the Internet. Also shown as part of system **400** is power supply unit **405** connected, in this example, to ac supply **406**. Not shown are batteries that could be present, and many other devices and modifications that are well known but are not applicable to the specific novel functions of the current system and method disclosed herein. It should be appreciated that some or all components illustrated may be combined, such as in various integrated applications (for example, Qualcomm or Samsung SOC-based devices), or whenever it may be appropriate to combine multiple capabilities or functions into a single hardware device (for instance, in mobile devices such as smartphones, video game consoles, in-vehicle computer systems such as navigation or multimedia systems in automobiles, or other integrated hardware devices).

[0054] In various embodiments, functionality for implementing systems or methods of the present invention may be distributed among any number of client and/or server components. For example, various software modules may be implemented for performing various functions in connection with the present invention, and such modules may be variously implemented to run on server and/or client components.

Conceptual Architecture

[0055] Fig. 5 is a block diagram of an exemplary architecture for a system **500** for predictive analysis of very large data sets using a distributed computational graph. According to the embodiment, streaming input feeds **510** may be a variety of data sources which may include but are not limited to the internet **511**, arrays of physical sensors **512**, database servers **513**, electronic monitoring equipment **514** and direct human interaction **515** ranging from a relatively few number of participants to a large crowd sourcing campaign. Streaming data from any combinations of listed sources and those not listed may also be expected to occur as part of the operation of the invention as the number of streaming input sources is not limited by the design. All incoming streaming data may be passed through a data filter software module **520** to remove information that has been damaged in transit, is misconfigured, or is malformed in some way that precludes use. Many of the filter parameters may be expected to be preset prior to operation, however, design of the invention makes provision for the behavior of the filter software module **520** to be changed as progression of analysis requires through the automation of the system sanity and retrain software module **563** which may serve to optimize system operation and analysis function. The data stream may also be split into two identical substreams at the data filter software module **520** with one substream being fed into a streaming analysis pathway that includes the transformation pipeline software module **561** of the distributed computational graph **560**. The other substream may be fed to data formalization software module **530** as part of the batch analysis pathway. The data formalization module **530** formats the data stream entering the batch analysis pathway of the invention into data records to be stored by the input event data store **540**. The input event data store **540** can be a database of any architectural type known to those knowledgeable in the art, but based upon the quantity of the data the data store module would be expected to store and retrieve, options using highly distributed storage and map reduce query protocols, of which Hadoop is one, but not the only example, may be generally preferable to relational database schema.

[0056] Analysis of data from the input event data store may be performed by the batch event analysis software module **550**. This module may be used to analyze the data in the input event data store for temporal information such as trends, previous occurrences of the progression of a set of events, with outcome, the occurrence of a single specific event with all events recorded before and after whether deemed relevant at the time or not, and presence of a particular event with all documented possible causative and remedial elements, including best guess probability information. Those knowledgeable in the art will recognize that while examples here focus on having stores of

information pertaining to time, the use of the invention is not limited to such contexts as there are other fields where having a store of existing data would be critical to predictive analysis of streaming data **561**. The search parameters used by the batch event analysis software module **550** are preset by those conducting the analysis at the beginning of the process, however, as the search matures and results are gleaned from the streaming data during transformation pipeline software module **561** operation, providing the system more timely event progress details, the system sanity and retrain software module **563** may automatically update the batch analysis parameters **550**. Alternately, findings outside the system may precipitate the authors of the analysis to tune the batch analysis parameters administratively from outside the system **570, 562, 563**. The real-time data analysis core **560** of the invention should be considered made up of a transformation pipeline software module **561**, messaging module **562** and system sanity and retrain software module **563**.The messaging module **562** has connections from both the batch and the streaming data analysis pathways and serves as a conduit for operational as well as result information between those two parts of the invention. The message module also receives messages from those administering analyses **580**. Messages aggregated by the messaging module **562** may then be sent to system sanity and retrain software module **563** as appropriate. Several of the functions of the system sanity and retrain software module have already been disclosed. Briefly, this is software that may be used to monitor the progress of streaming data analysis optimizing coordination between streaming and batch analysis pathways by modifying or "retraining" the operation of the data filter software module **520**, data formalization software module **530** and batch event analysis software module **540** and the transformation pipeline module **550** of the streaming pathway when the specifics of the search may change due to results produced during streaming analysis. System sanity and retrain module **563** may also monitor for data searches or transformations that are processing slowly or may have hung and for results that are outside established data stability boundaries so that actions can be implemented to resolve the issue. While the system sanity and retrain software module **563** may be designed to act autonomously and employs computer learning algorithms, according to some arrangements status updates may be made by administrators or potentially direct changes to operational parameters by such, according to the embodiment.

[0057] Streaming data entering from the outside data feeds **510** through the data filter software module **520** may be analyzed in real time within the transformation pipeline software module **561**. Within a transformation pipeline, a set of functions tailored to the analysis being run are applied to

the input data stream. According to the embodiment, functions may be applied in a linear, directed path or in more complex configurations. Functions may be modified over time during an analysis by the system sanity and retrain software module **563** and the results of the transformation pipeline, impacted by the results of batch analysis are then output in the format stipulated by the authors of the analysis which may be human readable printout, an alarm, machine readable information destined for another system or any of a plurality of other forms known to those in the art.

[0058] Fig. 6 (Prior Art) is a block diagram of a preferred architecture for a transformation pipeline within a system for predictive analysis of very large data sets using distributed computational graph **600**. According to the embodiment, streaming input from the data filter software module **520**, **615** serves as input to the first transformation node **620** of the transformation pipeline. Transformation node's function is performed on input data stream and transformed output message **625** is sent to transformation node 2 **630**. The progression of transformation nodes **620, 630, 640, 650, 660** and associated output messages from each node **625, 635, 645, 655** is linear in configuration this is the simplest arrangement and, as previously noted, represents the current state of the art. While transformation nodes are described according to various embodiments as uniform shape (referring to **Figs. 6-9**), such uniformity is used for presentation simplicity and clarity and does not reflect necessary operational similarity between transformations within the pipeline. It should be appreciated that one knowledgeable in the field will realize that certain transformations in a pipeline may be entirely self-contained; certain transformations may involve direct human interaction **630**, such as selection via dial or dials, positioning of switch or switches, or parameters set on control display, all of which may change during analysis; other transformations may require external aggregation or correlation services or may rely on remote procedure calls to synchronous or asynchronous analysis engines as might occur in simulations among a plurality of other possibilities. Further according to the embodiment, individual transformation nodes in one pipeline may represent function of another transformation pipeline. It should be appreciated that the node length of transformation pipelines depicted in no way confines the transformation pipelines employed by the invention to an arbitrary maximum length **640, 650, 660** as, being distributed, the number of transformations would be limited by the resources made available to each implementation of the invention. It should be further appreciated that there need be no limits on transform pipeline length. Output of the last transformation node and by extension, the transform pipeline **660** may be sent back to messaging software module **562** for predetermined action.

[0059] Fig. 7 is a block diagram of another preferred architecture for a transformation pipeline within a system for predictive analysis of very large data sets using distributed computational graph **700**. According to the embodiment, streaming input from a data filter software module **520**, **705** serves as input to the first transformation node **710** of the transformation pipeline. Transformation node's function is performed on input data stream and transformed output message **715** is sent to transformation node 2 **720**. In this embodiment, transformation node 2 **720** has a second input stream **760**. The specific source of this input is inconsequential to the operation of the invention and could be another transformation pipeline software module, a data store, human interaction, physical sensors, monitoring equipment for other electronic systems or a stream from the internet as from a crowdsourcing campaign, just to name a few possibilities **760**. Functional integration of a second input stream into one transformation node requires the two input stream events be serialized. The invention performs this serialization using a decomposable transformation software module (not shown), the function of which is described below, referring to **Fig. 13**. While transformation nodes are described according to various embodiments as uniform shape (referring to **Figs. 6-9**), such uniformity is used for presentation simplicity and clarity and does not reflect necessary operational similarity between transformations within the pipeline. It should be appreciated that one knowledgeable in the field will realize that certain transformations in a pipeline may be entirely self-contained; certain transformations may involve direct human interaction **630**, such as selection via dial or dials, positioning of switch or switches, or parameters set on control display, all of which may change during analysis; other transformations may require external aggregation or correlation services or may rely on remote procedure calls to synchronous or asynchronous analysis engines as might occur in simulations among a plurality of other possibilities. Further according to the embodiment, individual transformation nodes in one pipeline may represent function of another transformation pipeline. It should be appreciated that the node length of transformation pipelines depicted in no way confines the transformation pipelines employed by the invention to an arbitrary maximum length **710, 720, 730**, **740, 750,** as, being distributed, the number of transformations would be limited by the resources made available to each implementation of the invention. It should be further appreciated that there need be no limits on transform pipeline length. Output of the last transformation node and by extension, the transform pipeline, **750** may be sent back to messaging software module **562** for pre-decided action.

[0060] Fig. 8 is a block diagram of another preferred architecture for a transformation pipeline within a system for predictive analysis of very large data sets using distributed computational graph **700**. According to the embodiment, streaming input from a data filter software module **520**, **805** serves as input to the first transformation node **810** of the transformation pipeline. Transformation node's function is performed on input data stream and transformed output message **815** is sent to transformation node 2 **820**. In this embodiment, transformation node 2 **820** sends its output stream to two transformation pipelines **830, 840, 850; 865, 875**. This allows the same data stream to undergo two disparate, possibly completely unrelated, analyses without having to duplicate the infrastructure of the initial transform manipulations, greatly increasing the expressivity of the invention over current transform pipelines. Functional integration of a second output stream from one transformation node 820 requires that the two output stream events be serialized. The invention performs this serialization using a decomposable transformation software module (not shown), the function of which is described below, referring to **Fig. 14**. While transformation nodes are described according to various embodiments as uniform shape (referring to **Figs. 6-9**), such uniformity is used for presentation simplicity and clarity and does not reflect necessary operational similarity between transformations within the pipeline. It should be appreciated that one knowledgeable in the field will realize that certain transformations in pipelines, which may be entirely self-contained; certain transformations may involve direct human interaction **630**, such as selection via dial or dials, positioning of switch or switches, or parameters set on control display, all of which may change during analysis; other transformations may require external aggregation or correlation services or may rely on remote procedure calls to synchronous or asynchronous analysis engines as might occur in simulations, among a plurality of other possibilities. Further according to the embodiment, individual transformation nodes in one pipeline may represent function of another transformation pipeline. It should be appreciated that the node number of transformation pipelines depicted in no way confines the transformation pipelines employed by the invention to an arbitrary maximum length **810, 820, 830**, **840, 850; 865, 875** as, being distributed, the number of transformations would be limited by the resources made available to each implementation of the invention. Further according to the embodiment, there need be no limits on transform pipeline length. Output of the last transformation node and by extension, the transform pipeline **850** may be sent back to messaging software module **562** for contemporary enabled action.

[0061] Fig. 9 is a block diagram of another preferred architecture for a transformation pipeline within a system for predictive analysis of very large data sets using distributed computational graph **700**. According to the embodiment, streaming input from a data filter software module **520**, **905** serves as input to the first transformation node **910** of the transformation pipeline. Transformation node's function may be performed on an input data stream and transformed output message **915** may then be sent to transformation node 2 **920**. Likewise, once the data stream is acted upon by transformation node 2 **920**, its output is sent to transformation node 3 **930** using its output message **925** In this embodiment, transformation node 3 **930** sends its output stream back to transform node 1 **910** forming a cyclical relationship between transformation nodes 1 **910**, transformation node 2 **920** and transformation node 3 **930**. Upon the achievement of some gateway result, the output of cyclical pipeline activity may be sent to downstream transformation nodes within the pipeline **940**, **945**. The presence of a generalized cyclical pathway construct allows the invention to be used to solve complex iterative problems with large data sets involved, expanding ability to rapidly retrieve conclusions for complicated issues. Functional creation of a cyclical transformation pipeline requires that each cycle be serialized. The invention performs this serialization using a decomposable transformation software module (not shown), the function of which is described below, referring to **Fig. 15**. While transformation nodes are described according to various embodiments as uniform shape (referring to **Figs. 6-9**), such uniformity is used for presentation simplicity and clarity and does not reflect necessary operational similarity between transformations within the pipeline. It should be appreciated that one knowledgeable in the field will appreciate that certain transformations in pipelines, may be entirely self-contained; certain transformations may involve direct human interaction **630**, such as selection via dial or dials, positioning of switch or switches, or parameters set on control display, all of which may change during analysis; still other transformations may require external aggregation or correlation services or may rely on remote procedure calls to synchronous or asynchronous analysis engines as might occur in simulations, among a plurality of other possibilities. Further according to the embodiment, individual transformation nodes in one pipeline may represent the cumulative function of another transformation pipeline. It should be appreciated that the node number of transformation pipelines depicted in no way confines the transformation pipelines employed by the invention to an arbitrary maximum length **910, 920, 930, 940, 950; 965, 975** as, being distributed, the number of transformations would be limited by the resources made available to each implementation of the invention. It should be further appreciated

that there need be no limits on transform pipeline length. Output of the last transformation node and by extension, the transform pipeline **955** may be sent back to messaging software module **562** for concomitant enabled action.

Description of Method Embodiments

[0062] Fig. 10 is a process flow diagram of a method **1000** for predictive analysis of very large data sets using the distributed computational graph. One or more streams of data from a plurality of sources, which includes, but is in no way not limited to, a number of physical sensors, web based questionnaires and surveys, monitoring of electronic infrastructure, crowd sourcing campaigns, and direct human interaction, may be received by system **1001**. The received stream is filtered **1002** to exclude data that has been corrupted, data that is incomplete or misconfigured and therefore unusable, data that may be intact but nonsensical within the context of the analyses being run, as well as a plurality of predetermined analysis related and unrelated criteria set by the authors. Filtered data may be split into two identical streams at this point (second stream not depicted for simplicity), wherein one substream may be sent for batch processing **1600** while another substream may be formalized **1003** for transformation pipeline analysis **1004, 561, 600, 700, 800, 900**. Data formalization for transformation pipeline analysis acts to reformat the stream data for optimal, reliable use during analysis. Reformatting might entail, but is not limited to: setting data field order, standardizing measurement units if choices are given, splitting complex information into multiple simpler fields, and stripping unwanted characters, again, just to name a few simple examples. The formalized data stream may be subjected to one or more transformations. Each transformation acts as a function on the data and may or may not change the data. Within the invention, transformations working on the same data stream where the output of one transformation acts as the input to the next are represented as transformation pipelines. While the great majority of transformations in transformation pipelines receive a single stream of input, modify the data within the stream in some way and then pass the modified data as output to the next transformation in the pipeline, the invention does not require these characteristics. According to the embodiment, individual transformations can receive input of expected form from more than one source **1300** or receive no input at all as would a transformation acting as a timestamp. According to the embodiment, individual transformations, may not modify the data as would be encountered with a data store acting as a queue for downstream transformations **1303, 1305, 1405, 1407,1505**. According to the embodiment, individual transformations may provide output to more than one downstream

transformations **1400**. This ability lends itself to simulations where multiple possible choices might be made at a single step of a procedure all of which need to be analyzed. While only a single, simple use case has been offered for each example, in each case, that example was chosen for simplicity of description from a plurality of possibilities, the examples given should not be considered to limit the invention to only simplistic applications. Last, according to the invention, transformations in a transformation pipeline backbone may form a linear, a quasi- linear arrangement or may be cyclical **1500**, where the output of one of the internal transformations serves as the input of one of its antecedents allowing recursive analysis to be run. The result of transformation pipeline analysis may then be modified by results from batch analysis of the data stream **1600** and output in format predesigned by the authors of the analysis with could be human readable summary printout, human readable instruction printout, human-readable raw printout, data store, or machine encoded information of any format known to the art to be used in further automated analysis or action schema.

[0063] Fig. 11 is a process flow diagram of a method **1100** for an embodiment of modeling the transformation pipeline module **561** of the invention as a directed graph using graph theory. According to the embodiment, the individual transformations **1102, 1104, 1106** of the transformation pipeline $t_1..t_n$ such that each t_i T are represented as graph nodes. Transformations belonging to T are discrete transformations over individual datasets d_i , consistent with classical functions. As such, each individual transformation t_j, receives a set of inputs and produces a single output. The input of an individual transformation t_i, is defined with the function *in:* t_i $d_1..d_k$ such that $in(t_i)=\{d_1..d_k)$ and describes a transformation with k inputs. Similarly, the output of an individual transformation is defined as the function *out:* t_i $[ld_1]$ to describe transformations that produce a single output (usable by other transformations). A dependency function can now be defined such that dep(t_a,t_b) out(t_a)in(t_b)The messages carrying the data stream through the transformation pipeline **1101,1103, 1105** make up the graph edges. Using the above definitions, then, a transformation pipeline within the invention can be defined as G=(V,E) where message(t_1,t_2..$t_{(n-1)}$,t_n)V and all transformations $t_1..t_n$ and all dependencies $dep(t_i,t_j)$E **1107**.

[0064] Fig. 12 (Prior Art) is a process flow diagram of a method **1200** for one embodiment of a linear transformation pipeline **1201**. This is the simplest of configurations as the input stream is acted upon by the first transformation node **1202** and the remainder of the transformations within the pipeline are then performed sequentially **1202**, **1203**, **1204**, **1205** for the entire pipeline with no

introduction of new data internal to the initial node or splitting output stream prior to last node of the pipeline **1205**. This configuration is the current state of the art for transformation pipelines and is the most general form of these constructs. Linear transformation pipelines require no special manipulation to simplify the data pathway and are thus referred to as non-decomposable. The example depicted in this diagram was chosen to convey the configuration of a linear transformation pipeline and is the simplest form of the configuration felt to show the point. It in no way implies limitation of the invention.

[0065] Fig. 13 is a process flow diagram of a method **1300** for one embodiment of a transformation pipeline where one transformation node **1307** in a transformation pipeline receives data streams from two source transformation nodes **1301**. The invention handles this transformation pipeline configuration by decomposing or serializing the input events **1302-1303**, **1304-1305** heavily relying on post transformation function continuation**.** The results of individual transformation nodes **1302**, **1304** just antecedent to the destination transformation node **1306** and placed into a single specialized data storage transformation node **1303, 1305** (shown twice as process occurs twice). The combined results then retrieved from the data store **1306** and serve as the input stream for the transformation node within the transformation pipeline backbone **1307, 1308**. The example depicted in this diagram was chosen to convey the configuration of transformation pipelines with individual transformation nodes that receive input from two source nodes **1302, 1304** and is the simplest form of the configuration felt to show the point. It in no way implies limitation of the invention. One knowledgeable in the art will realize the great number of permutations and topologies possible, especially as the invention places no design restrictions on the number of transformation nodes receiving input from greater than one sources or the number sources providing input to a destination node.

[0066] Fig. 14 is a process flow diagram of a method **1400** for one embodiment of a transformation pipeline where one transformation node **1403** in a transformation pipeline sends output data stream to two destination transformation nodes **1401**, **1406, 1408** in potentially two separate transformation pipelines. The invention handles this transformation pipeline configuration by decomposing or serializing the output events **1404**,**1405-1406**, **1407-1408.** The results of the source transformation node **1403** just antecedent to the destination transformation nodes **1406** and placed into a single specialized data storage transformation node **1404, 1405, 1407** (shown three times as storage occurs and retrieval occurs twice). The results of the antecedent transformation node may then be retrieved

from a data store **1404** and serves as the input stream for the transformation nodes two downstream transformation pipeline **1406, 1408**. The example depicted in this diagram was chosen to convey the configuration of transformation pipelines with individual transformation nodes that send output streams to two destination nodes **1406, 1408** and is the simplest form of the configuration felt to show the point. It in no way implies limitation of the invention. One knowledgeable in the art will realize the great number of permutations and topologies possible, especially as the invention places no design restrictions on the number of transformation nodes sending output to greater than one destination or the number destinations receiving input from a source node.

[0067] Fig. 15 is a process flow diagram of a method **1500** for one embodiment of a transformation pipeline where the topology of all or part of the pipeline is cyclical **1501**. In this configuration the output stream of one transformation node **1504** acts as an input of an antecedent transformation node within the pipeline **1502** serialization or decomposition linearizes this cyclical configuration by completing the transformation of all of the nodes that make up a single cycle **1502, 1503, 1504** and then storing the result of that cycle in a data store **1505**. That result of a cycle is then reintroduced to the transformation pipeline as input to the first transformation node of the cycle. As this configuration is by nature recursive, special programming to unfold the recursions was developed for the invention to accommodate it. The example depicted in this diagram was chosen to convey the configuration of transformation pipelines with individual transformation nodes that for a cyclical configuration **1501, 1502, 1503, 1504** and is the simplest form of the configuration felt to show the point. It in no way implies limitation of the invention. One knowledgeable in the art will realize the great number of permutations and topologies possible, especially as the invention places no design restrictions on the number of transformation nodes participating in a cycle nor the number of cycles in a transformation pipeline.

[0068] Fig. 16 is a process flow diagram of a method **1600** for one embodiment of the batch data stream analysis pathway which forms part of the invention and allows streaming data to be interpreted with historic context. One or more streams of data from a plurality of sources, which includes, but is in no way not limited to, a number of physical sensors, web based questionnaires and surveys, monitoring of electronic infrastructure, crowd sourcing campaigns, and direct human interaction, is received by the system **1601**. The received stream may be filtered **1602** to exclude data that has been corrupted, data that is incomplete or misconfigured and therefore unusable, data that may be intact but nonsensical within the context of the analyses being run, as well as a plurality

of predetermined analysis related and unrelated criteria set by the authors. Data formalization **1603** for batch analysis acts to reformat the stream data for optimal, reliable use during analysis. Reformatting might entail, but is not limited to: setting data field order, standardizing measurement units if choices are given, splitting complex information into multiple simpler fields, and stripping unwanted characters, again, just to name a few simple examples. The filtered and formalized stream is then added to a distributed data store **1604** due to the vast amount of information accrued over time. The invention has no dependency for specific data stores or data retrieval model. During transformation pipeline analysis of the streaming pipeline, data stored in the batch pathway store can be used to track changes in specifics of the data important to the ongoing analysis over time, repetitive data sets significant to the analysis or the occurrence of critical points of data **1605**. The functions of individual transformation nodes **620** may be saved and can be edited also all nodes of a transformation pipeline **600** keep a summary or summarized view (analogous to a network routing table) of applicable parts of the overall route of the pipeline along with detailed information pertaining to adjacent two nodes. This framework information enables steps to be taken and notifications to be passed if individual transformation nodes **640** within a transformation pipeline **600** become unresponsive during analysis operations. Combinations of results from the batch pathway, partial and streaming output results from the transformation pipeline, administrative directives from the authors of the analysis as well as operational status messages from components of the distributed computational graph are used to perform system sanity checks and retraining of one or more of the modules of the system **1606**. These corrections are designed to occur without administrative intervention under all but the most extreme of circumstances with deep learning capabilities present as part of the system manager and retrain module **563** responsible for this task.

[0069] The skilled person will be aware of a range of possible modifications of the various embodiments described above. Accordingly, the present invention is defined by the claims and their equivalents.

1. A system comprising:

a distributed computing cluster comprising a first plurality of computer systems,

wherein each respective computer system of the first plurality of computer systems comprises a memory that stores a respective first data,

wherein the respective first data represents a respective portion of a distributed computational graph,

and wherein the distributed computational graph describes a flow of output data of a first transformation pipeline to an input of a second transformation pipeline,

wherein a first computer system of the first plurality of computer systems is configured to:

receive a first stream of input data from an input feed,

process the first stream of input data substantially in real time by executing software instructions that apply the first transformation pipeline to the first stream of input data to generate first pipeline output messages,

process the respective first data stored in the memory of the first computer system to determine information about the second transformation pipeline,

and transmit the first pipeline output messages to a second computer system of the first plurality of computer systems in accordance with the determined information,

wherein the second computer system is configured to:

receive the first pipeline output messages,

and process the first pipeline output messages substantially in real time by executing software instructions that apply the second transformation pipeline to the first pipeline output messages to generate second pipeline output messages,

wherein the first and second computer systems are distinct.

2. The system of claim 1, wherein at least one of the first transformation pipeline and the second transformation pipeline is non-linear.

3. The system of claim 1,

wherein the first computer system is further configured to transmit the first pipeline output messages to a third computer system of the first plurality of computer systems,

wherein the first computer system is distinct from the third computer system,

and wherein the second computer system is distinct from the third computer system.

4. The system of claim 2,

wherein the first computer system is further configured to transmit the first pipeline output messages to a third computer system of the first plurality of computer systems,

wherein the first computer system is distinct from the third computer system,

and wherein the second computer system is distinct from the third computer system.

5. The system of claim 2,

wherein a third computer system of the first plurality of computer systems is configured to:

identify a fourth computer system of the first plurality of computer systems,

wherein the fourth computer system is not configured to apply the first transformation pipeline to any stream of input data,

and cause the fourth computer system to process a second stream of input data received from an input feed substantially in real time by executing software instructions that apply the first transformation pipeline to the second stream of input data.

6. The system of claim 5, wherein the third computer system is further configured to execute software instructions that:

identify one or more of the respective memories that store respective first data representing at least a respective portion of the distributed computational graph,

wherein the respective first data includes information about the first transformation pipeline,

and store in the identified one or more memories information identifying the fourth computer system.

7. The system of claim 2,

wherein a third computer system of the first plurality of computer systems is configured to:

identify a fourth computer system of the first plurality of computer systems,

wherein the fourth computer system is not configured to apply the second transformation pipeline to any pipeline output message,

and cause the fourth computer system to process third pipeline output messages substantially in real time by executing software instructions that apply the second transformation pipeline to the third pipeline output messages.

8. The system of claim 7, wherein the third computer system is further configured to execute software instructions that:

identify one or more of the respective memories that store respective first data representing at least a respective portion of the distributed computational graph,

wherein the respective first data includes information about the second transformation pipeline,

and store in the identified one or more memories information identifying the fourth computer system.

9. The system of claim 1,

wherein the distributed computing cluster further comprises a second plurality of computer systems that are not executing any portion of any transformation pipeline in the flow of the distributed computational graph,

and wherein a third computer system of the first plurality of computer systems is configured to execute software instructions that cause a fourth computer system of the second plurality of computer systems to execute software instructions that apply at least one of the first transformation pipeline and the second transformation pipeline.

10. The system of claim 2,

wherein the distributed computing cluster further comprises a second plurality of computer systems that are not executing any portion of any transformation pipeline in the flow of the distributed computational graph,

and wherein a third computer system of the first plurality of computer systems is configured to execute software instructions that cause a fourth computer system of the second plurality of

computer systems to execute software instructions that apply at least one of the first transformation pipeline and the second transformation pipeline.

11. A system comprising:

a distributed computing cluster comprising a first plurality of computer systems,

wherein each respective computer system of the first plurality of computer systems comprises a respective memory that stores a respective first data that:

describes at least a respective portion of a configuration of a first transformation pipeline,

and identifies at least a second transformation pipeline,

wherein a first computer system of the first plurality of computer systems is configured to:

receive a first stream of input data from an input feed,

process the first stream of input data substantially in real time by executing software instructions that apply the first transformation pipeline to the first stream of input data to generate first pipeline output messages,

process the respective first data to determine an identification of the second transformation pipeline,

and transmit the first pipeline output messages to a second computer system of the first plurality of computer systems in accordance with the determined identification,

wherein the second computer system is configured to:

receive the first pipeline output messages,

and process the first pipeline output messages substantially in real time by executing software instructions that apply the second transformation pipeline to the first pipeline output messages,

wherein the first and second computer systems are distinct.

12. The system of claim 11, wherein one or more of the first transformation pipeline and the second transformation pipeline is non-linear.

13. The system of claim 11,

wherein the first computer system is further configured to transmit the first pipeline output messages to a third computer system of the first plurality of computer systems,

wherein the first computer system is distinct from the third computer system,

and wherein the second computer system is distinct from the third computer system.

14. The system of claim 12,

wherein a third computer system of the first plurality of computer systems is configured to:

identify a fourth computer system of the first plurality of computer systems,

wherein the fourth computer system is not configured to apply the first transformation pipeline to any stream of input data,

and cause the fourth computer system to process a second stream of input data received from an input feed substantially in real time by executing software instructions that apply the first transformation pipeline to the second stream of input data.

15. The system of claim 14, wherein the third computer system is further configured to execute software instructions that:

identify one or more of the respective memories that stores respective data that describes a respective portion of a configuration of the first transformation pipeline,

and store in the identified one or more memories information identifying the fourth computer system.

16. The system of claim 12,

wherein a third computer system of the first plurality of computer systems is configured to:

identify a fourth computer system of the first plurality of computer systems,

wherein the fourth computer system is not configured to apply the second transformation pipeline to any pipeline output message,

and cause the fourth computer system to process second pipeline output messages substantially in real time by executing software instructions that apply the second transformation pipeline to the second pipeline output messages.

17. The system of claim 16, wherein the third computer system is further configured to execute software instructions that:

identify one or more of the respective memories that stores respective first data that describes a respective portion of a configuration of the second transformation pipeline,

and store in the identified one or more memories information identifying the fourth computer system.

18. The system of claim 11,

wherein the distributed computing cluster further comprises a second plurality of computer systems that are not executing any portion of any transformation pipeline described in any configuration described by respective first data stored in any of the respective memories of the first plurality of computer systems,

and wherein a third computer system of the first plurality of computer systems is configured to execute software instructions that cause the second plurality of computer systems to execute at least one of the first transformation pipeline and the second transformation pipeline.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	TOT CLAIMS	IND CLAIMS
18/779,029	07/21/2024		1820	15305FTNK	18	2

CONFIRMATION NO. 4119

FILING RECEIPT

82318
Brian Galvin
Galvin Patent Law LLC
2916 NW Bucklin Hill Road
Suite 485
Silverdale, WA 98383

Date Mailed: 08/22/2024

Receipt is acknowledged of this non-provisional utility patent application. The application will be taken up for examination in due course. Applicant will be notified as to the results of the examination. Any correspondence concerning the application must include the following identification information: the U.S. APPLICATION NUMBER, FILING DATE, NAME OF FIRST INVENTOR, and TITLE OF INVENTION. Fees transmitted by check or draft are subject to collection.

Please verify the accuracy of the data presented on this receipt. If an error is noted on this Filing Receipt, please submit a written request for a corrected Filing Receipt, including a properly marked-up ADS showing the changes with strike-through for deletions and underlining for additions. If you received a "Notice to File Missing Parts" or other Notice requiring a response for this application, please submit any request for correction to this Filing Receipt with your reply to the Notice. When the USPTO processes the reply to the Notice, the USPTO will generate another Filing Receipt incorporating the requested corrections provided that the request is grantable.

Inventor(s)

Jason Crabtree, Vienna, VA;
Andrew Sellers, Monument, CO;

Applicant(s)

QOMPLX LLC, Reston, VA;

Power of Attorney: The patent practitioners associated with Customer Number 82318

Domestic Priority data as claimed by applicant

This application is a CON of 18/581,375 02/20/2024
which is a CON of 17/189,161 03/01/2021
which is a CIP of 17/061,195 10/01/2020 PAT 11,570,214
which is a CIP of 17/035,029 09/28/2020 PAT 11,546,380
which is a CIP of 17/008,276 08/31/2020 PAT 11,323,484
which is a CIP of 17/000,504 08/24/2020 PAT 11,477,245
which is a CIP of 16/855,724 04/22/2020 PAT 11,218,510
which is a CIP of 16/836,717 03/31/2020 PAT 10,917,428
which is a CIP of 15/887,496 02/02/2018 PAT 10,783,241
which is a CIP of 15/823,285 11/27/2017 PAT 10,740,096
which is a CIP of 15/788,718 10/19/2017 PAT 10,861,014
which is a CIP of 15/788,002 10/19/2017 ABN
which is a CIP of 15/787,601 10/18/2017 PAT 10,860,660
which claims benefit of 62/568,312 10/04/2017
and is a CIP of 15/616,427 06/07/2017 ABN

which is a CIP of 14/925,974 10/28/2015 ABN
and said 15/788,002 10/19/2017 claims benefit of 62/568,305 10/04/2017
and said 15/788,718 10/19/2017 claims benefit of 62/568,307 10/04/2017
and said 15/887,496 02/02/2018 is a CIP of 15/818,733 11/20/2017 PAT 10,673,887
which is a CIP of 15/725,274 10/04/2017 PAT 10,609,079
which is a CIP of 15/655,113 07/20/2017 PAT 10,735,456
which is a CIP of 15/616,427 06/07/2017 ABN
and is a CIP of 15/237,625 08/15/2016 PAT 10,248,910
which is a CIP of 15/206,195 07/08/2016 ABN
which is a CIP of 15/186,453 06/18/2016 ABN
which is a CIP of 15/166,158 05/26/2016 ABN
which is a CIP of 15/141,752 04/28/2016 PAT 10,860,962
which is a CIP of 15/091,563 04/05/2016 PAT 10,204,147
and is a CIP of 14/986,536 12/31/2015 PAT 10,210,255
and is a CIP of 14/925,974 10/28/2015 ABN
and said 16/855,724 04/22/2020 is a CIP of 16/777,270 01/30/2020 PAT 11,025,674
which is a CIP of 16/720,383 12/19/2019 PAT 10,944,795
which is a CON of 15/823,363 11/27/2017 PAT 10,560,483
which is a CIP of 15/725,274 10/04/2017 PAT 10,609,079
and said 17/000,504 08/24/2020 is a CIP of 16/412,340 05/14/2019 PAT 11,539,663
which is a CIP of 16/267,893 02/05/2019 ABN
which is a CIP of 16/248,133 01/15/2019 ABN
which is a CIP of 15/849,901 12/21/2017 PAT 11,023,284
which is a CIP of 15/835,436 12/07/2017 PAT 10,572,828
which is a CIP of 15/790,457 10/23/2017 PAT 10,884,999
which is a CIP of 15/790,327 10/23/2017 PAT 10,860,951
which claims benefit of 62/568,291 10/04/2017
and is a CIP of 15/616,427 06/07/2017 ABN
and is a CIP of 15/141,752 04/28/2016 PAT 10,860,962
and said 15/790,457 10/23/2017 claims benefit of 62/568,298 10/04/2017
and said 15/849,901 12/21/2017 is a CIP of 15/835,312 12/07/2017 PAT 11,055,451
which is a CIP of 15/186,453 06/18/2016 ABN
and said 16/248,133 01/15/2019 is a CIP of 15/813,097 11/14/2017 ABN
which is a CIP of 15/616,427 06/07/2017 ABN
and said 16/248,133 01/15/2019 is a CIP of 15/806,697 11/08/2017 ABN
which is a CIP of 15/376,657 12/13/2016 PAT 10,402,906
which is a CIP of 15/237,625 08/15/2016 PAT 10,248,910
and said 15/806,697 11/08/2017 is a CIP of 15/343,209 11/04/2016 PAT 11,087,403
which is a CIP of 15/237,625 08/15/2016 PAT 10,248,910
and is a CIP of 15/229,476 08/05/2016 PAT 10,454,791
which is a CIP of 15/206,195 07/08/2016 ABN
and said 16/248,133 01/15/2019 is a CIP of 15/673,368 08/09/2017 ABN
which is a CIP of 15/376,657 12/13/2016 PAT 10,402,906
and said 17/061,195 10/01/2020 is a CIP of 15/879,801 01/25/2018 ABN
which is a CIP of 15/379,899 12/15/2016 ABN
which is a CIP of 15/376,657 12/13/2016 PAT 10,402,906
and said 17/189,161 03/01/2021 is a CIP of 16/709,598 12/10/2019 PAT 11,507,858
which is a CIP of 14/925,974 10/28/2015 ABN

Foreign Applications for which priority is claimed (You may be eligible to benefit from the Patent Prosecution Highway program at the USPTO. Please see http://www.uspto.gov for more information.) - None.

Foreign application information must be provided in an Application Data Sheet in order to constitute a claim to foreign priority. See 37 CFR 1.55 and 1.76.

Permission to Access Application via Priority Document Exchange: Yes

Permission to Access Search Results: Yes

Applicant may provide or rescind an authorization for access using Form PTO/SB/39 or Form PTO/SB/69 as appropriate.

If Required, Foreign Filing License Granted: 08/20/2024

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 18/779,029**

Projected Publication Date: 11/28/2024

Non-Publication Request: No

Early Publication Request: No

Title

RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH

Preliminary Class

Statement under 37 CFR 1.55 or 1.78 for AIA (First Inventor to File) Transition Applications: No

PROTECTING YOUR INVENTION OUTSIDE THE UNITED STATES

Since the rights granted by a U.S. patent extend only throughout the territory of the United States and have no effect in a foreign country, an inventor who wishes patent protection in another country must apply for a patent in a specific country or in regional patent offices. Applicants may wish to consider the filing of an international application under the Patent Cooperation Treaty (PCT). An international (PCT) application generally has the same effect as a regular national patent application in each PCT-member country. The PCT process **simplifies** the filing of patent applications on the same invention in member countries, but **does not result** in a grant of "an international patent" and does not eliminate the need of applicants to file additional documents and fees in countries where patent protection is desired.

Almost every country has its own patent law, and a person desiring a patent in a particular country must make an application for patent in that country in accordance with its particular laws. Since the laws of many countries differ in various respects from the patent law of the United States, applicants are advised to seek guidance from specific foreign countries to ensure that patent rights are not lost prematurely.

Applicants also are advised that in the case of inventions made in the United States, the Director of the USPTO must issue a license before applicants can apply for a patent in a foreign country. The filing of a U.S. patent application serves as a request for a foreign filing license. The application's filing receipt contains further information and guidance as to the status of applicant's license for foreign filing.

Applicants may wish to consult the USPTO booklet, "General Information Concerning Patents" (specifically, the section entitled "Treaties and Foreign Patents") for more information on timeframes and deadlines for filing foreign

patent applications. The guide is available either by contacting the USPTO Contact Center at 800-786-9199, or it can be viewed on the USPTO website at http://www.uspto.gov/web/offices/pac/doc/general/index.html.

For information on preventing theft of your intellectual property (patents, trademarks and copyrights), you may wish to consult the U.S. Government website, http://www.stopfakes.gov. Part of a Department of Commerce initiative, this website includes self-help "toolkits" giving innovators guidance on how to protect intellectual property in specific countries such as China, Korea and Mexico. For questions regarding patent enforcement issues, applicants may call the U.S. Government hotline at 1-866-999-HALT (1-866-999-4258).

LICENSE FOR FOREIGN FILING UNDER

Title 35, United States Code, Section 184

Title 37, Code of Federal Regulations, 5.11 & 5.15

GRANTED

The applicant has been granted a license under 35 U.S.C. 184, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" followed by a date appears on this form. Such licenses are issued in all applications where the conditions for issuance of a license have been met, regardless of whether or not a license may be required as set forth in 37 CFR 5.15. The scope and limitations of this license are set forth in 37 CFR 5.15(a) unless an earlier license has been issued under 37 CFR 5.15(b). The license is subject to revocation upon written notification. The date indicated is the effective date of the license, unless an earlier license of similar scope has been granted under 37 CFR 5.13 or 5.14.

This license is to be retained by the licensee and may be used at any time on or after the effective date thereof unless it is revoked. This license is automatically transferred to any related applications(s) filed under 37 CFR 1.53(d). This license is not retroactive.

The grant of a license does not in any way lessen the responsibility of a licensee for the security of the subject matter as imposed by any Government contract or the provisions of existing laws relating to espionage and the national security or the export of technical data. Licensees should apprise themselves of current regulations especially with respect to certain countries, of other agencies, particularly the Office of Defense Trade Controls, Department of State (with respect to Arms, Munitions and Implements of War (22 CFR 121-128)); the Bureau of Industry and Security, Department of Commerce (15 CFR parts 730-774); the Office of Foreign AssetsControl, Department of Treasury (31 CFR Parts 500+) and the Department of Energy.

NOT GRANTED

No license under 35 U.S.C. 184 has been granted at this time, if the phrase "IF REQUIRED, FOREIGN FILING LICENSE GRANTED" DOES NOT appear on this form. Applicant may still petition for a license under 37 CFR 5.12, if a license is desired before the expiration of 6 months from the filing date of the application. If 6 months has lapsed from the filing date of this application and the licensee has not received any indication of a secrecy order under 35 U.S.C. 181, the licensee may foreign file the application pursuant to 37 CFR 5.15(b).

SelectUSA

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The U.S. offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to promote and facilitate business investment. SelectUSA provides information assistance to the international investor community; serves as an ombudsman for existing and potential investors; advocates on behalf of U.S. cities, states, and regions competing for global investment; and counsels U.S. economic development organizations on investment attraction best practices. To learn more about why the United States is the best country in the world to develop technology, manufacture products, deliver services, and grow your business, visit http://www.SelectUSA.gov or call +1-202-482-6800.

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number: 18/779,029

APPLICATION AS FILED - PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY FEE($)
BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A	N/A			N/A	320
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A			N/A	700
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A			N/A	800
TOTAL CLAIMS (37 CFR 1.16(i))	18 minus 20 =	* 0			OR	x =	0
INDEPENDENT CLAIMS (37 CFR 1.16(h))	2 minus 3 =	*				x =	0
APPLICATION SIZE FEE (37 CFR 1.16(s))	If the specification and drawings exceed 100 sheets of paper, the application size fee due is $310 ($155 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).						0
MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))							0
* If the difference in column 1 is less than zero, enter "0" in column 2.			TOTAL			TOTAL	1820

APPLICATION AS AMENDED - PART II

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY ADDITIONAL FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE($)
AMENDMENT A	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

		(Column 1) CLAIMS REMAINING AFTER AMENDMENT		(Column 2) HIGHEST NUMBER PREVIOUSLY PAID FOR	(Column 3) PRESENT EXTRA	SMALL ENTITY RATE($)	SMALL ENTITY ADDITIONAL FEE($)	OR	OTHER THAN SMALL ENTITY RATE($)	OTHER THAN SMALL ENTITY ADDITIONAL FEE($)
AMENDMENT B	Total (37 CFR 1.16(i))	*	Minus	**	=	x =		OR	x =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=	x =		OR	x =	
	Application Size Fee (37 CFR 1.16(s))									
	FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							OR		
						TOTAL ADD'L FEE		OR	TOTAL ADD'L FEE	

* If the entry in column 1 is less than the entry in column 2, write "0" in column 3.

** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 20, enter "20".

*** If the "Highest Number Previously Paid For" IN THIS SPACE is less than 3, enter "3".

The "Highest Number Previously Paid For" (Total or Independent) is the highest found in the appropriate box in column 1.

United States Patent and Trademark Office

Under Secretary of Commerce for Intellectual Property and
Director of the United States Patent and Trademark Office

WELCOME TO THE UNITED STATES PATENT AND TRADEMARK OFFICE (USPTO)

*Congratulations on taking the first step to protect your invention. You can now advertise or mark any products covered by this application for your invention as **"Patent Pending"** for as long as this application is pending!*

We are here to help you on the rest of your journey.

New to the process? The Patent Basics page will take you from "What is a patent?" to assistance with the application process.

- **www.uspto.gov/PatentBasics**

Consider cost savings. You might qualify for small entity or micro entity status, which means a discount on most USPTO patent fees.

- **www.uspto.gov/EntityStatus**

Become a registered electronic filer (eFiler). You'll gain access to the full capabilities of Patent Center. As a registered eFiler, you can electronically view your submission, track the progress of your application, and respond to USPTO correspondence.

- **www.uspto.gov/NewPatentFilers**

Learn how to identify which of your creative ideas might be intellectual property assets and how to protect them by using the USPTO Intellectual Property (IP) Identifier.

- **ipidentifier.uspto.gov**

We also provide Pro Se Assistance, support specifically geared for **independent inventors without an attorney or agent.** Follow the link below, call us at 866-767-3848, or email **proseassistancecenter@uspto.gov**.

- **www.uspto.gov/ProSePatents**

Need help near you? We have regional offices across the country, as well as Patent and Trademark Resource Centers in libraries in nearly every state, with specialists standing by to help you at no cost.

- **www.uspto.gov/locations**
- **www.uspto.gov/ptrc**

Contact us:

- **General information:** USPTO Contact Center
  - **www.uspto.gov/help**
- **Patent Center:**
  Patent Electronic Business Center
  - **www.uspto.gov/EBC**
- **Pre or post-examination:** Application Assistance Unit
  - **www.uspto.gov/PatentAssistance**

Want to learn more?

- **Access our free services** all in one spot. Find all of our contact centers, nationwide services, events, education, and training.
  - **www.uspto.gov/FreeServices**
- **Connect with us** on Instagram, LinkedIn, Twitter, Facebook, and YouTube.
- **Hear about resources for startups, inventors, and entrepreneurs,** as well as our initiatives engaging women, the military community, and others. These programs offer helpful tips and tricks on identifying and managing your IP, finding options for funding, and expanding your network to support your business.
  - **www.uspto.gov/startups**
  - **www.uspto.gov/inventors**
  - **www.uspto.gov/initiatives**
- We hope you connect with us by subscribing to receive important notifications by email.
  - **www.uspto.gov/subscribe**

Kathi Vidal

Kathi Vidal
Under Secretary of Commerce for Intellectual Property and
Director of the United States Patent and Trademark Office

Derrick Brent
Deputy Under Secretary of Commerce for Intellectual Property and
Deputy Director of the United States Patent and Trademark Office

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
18/779,029	07/21/2024	Jason Crabtree	15305FTNK	4119

82318 7590 09/06/2024

Brian Galvin
Galvin Patent Law LLC
2916 NW Bucklin Hill Road
Suite 485
Silverdale, WA 98383

EXAMINER	
CENTRAL, DOCKET	

ART UNIT	PAPER NUMBER
OPAP	

NOTIFICATION DATE	DELIVERY MODE
09/06/2024	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

USPTO@GALVINPATENTLAW.COM
eofficeaction@appcoll.com

Decision Granting Request for Prioritized Examination (Track I)	**Application No.** 18/779,029	**Applicant(s)** Crabtree et al.	
	Examiner FIKIRTE A GEREMEW	**Art Unit** OMBL	**AIA (FITF) Status** Yes

1. THE REQUEST FILED 21 July 2024 IS **GRANTED** .

   The above-identified application has met the requirements for prioritized examination
   - A. ☑ for an original nonprovisional application (Track I).
   - B. ☐ for an application undergoing continued examination (RCE).

2. **The above-identified application will undergo prioritized examination**. The application will be accorded special status throughout its entire course of prosecution until one of the following occurs:
   - A. filing a **petition for extension of time** to extend the time period for filing a reply;
   - B. filing an **amendment to amend the application to contain more than four independent claims, more than thirty total claims** , or a multiple dependent claim;
   - C. filing a **request for continued examination** ;
   - D. filing a notice of appeal;
   - E. filing a request for suspension of action;
   - F. mailing of a notice of allowance;
   - G. mailing of a final Office action;
   - H. completion of examination as defined in 37 CFR 41.102; or
   - I. abandonment of the application.

Telephone inquiries with regard to this decision should be directed to FIKIRTE GEREMEW at (703) 756-1930. In his/her absence, calls may be directed to Petition Help Desk at (571) 272-3282.

/FIKIRTE A GEREMEW/ PROGRAM SUPPORT SPECIALIST, OMBL	

UNITED STATES
PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

APPROVAL LETTER

APPLICATION #	FILING DATE	APPLICANT/PATENT UNDER REEXAMINATION
18/779,029	07/21/2024	Jason Crabtree

Title of Invention

RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH

Electronic terminal disclaimer filed on 09/26/2024

Approved

This patent is subject to a Terminal Disclaimer

Approved / Disapproved by: Electronic Terminal Disclaimer automatically approved

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC PAYMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
18/779,029	**09/26/2024 06:49:07 PM Z ET**	**15305FT**

Title of Invention

RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	-
CONFIRMATION #	4119	FILED BY	Aura Morrey
PATENT CENTER #	67343320	AUTHORIZED BY	Brian Galvin
CUSTOMER #	82318	FILING DATE	07/21/2024
CORRESPONDENCE ADDRESS	-	FIRST NAMED INVENTOR	Jason Crabtree

Payment Information

PAYMENT METHOD	PAYMENT TRANSACTION ID	PAYMENT AUTHORIZED BY
CARD / 1009	**E20249PI49458017**	**Brian Galvin**

FEE CODE	DESCRIPTION	ITEM PRICE($)	QUANTITY	ITEM TOTAL($)
1814	STATUTORY DISCLAIMER, INCLUDING TERMINAL DISCLAIMER	170.00	1	170.00
			TOTAL AMOUNT:	**$170.00**

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC ACKNOWLEDGEMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
18/779,029	**09/26/2024 06:49:07 PM Z ET**	**15305FT**

Title of Invention

RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	-
CONFIRMATION #	4119	FILED BY	Aura Morrey
PATENT CENTER #	67343320	FILING DATE	07/21/2024
CUSTOMER #	82318	FIRST NAMED INVENTOR	Jason Crabtree
CORRESPONDENCE ADDRESS	-	AUTHORIZED BY	Brian Galvin

Documents

TOTAL DOCUMENTS: 2

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
petition-request.pdf	3	Terminal Disclaimer-Filed (Electronic)	44 KB
grantLetter.pdf	1	Terminal Disclaimer-Electronic-Approved	18 KB

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)
petition-request.pdf	4CE1244464AF6F0A858AD1150BB8D23F29A5E9EFF0D816CC ADFFC7F8A2588CCF6FD9BBA97B8A02A8F07BFDC1FC8E16E 338A8B26C3D102598D3FE127ECD919E98

grantLetter.pdf

16CA091E2D15A4D2903232E5DAED04C6B837B1BE62E5A33A
28F83243739DE2E3600EBD8B0759697459848DE704D2A6C730
6C98F7F7F44FC2CE83E00C9CB93B3D

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

TERMINAL DISCLAIMER TO OBVIATE A DOUBLE PATENTING REJECTION OVER A PRIOR PATENT

APPLICATION #	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET #
18779029	**07/21/2024**	**Jason Crabtree**	**15305FT**

Title of Invention

RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH

☑ Filing of terminal disclaimer does not obviate requirement for response under 37 CFR 1.111 to outstanding Office Action

☑ This electronic Terminal Disclaimer is not being used for a Joint Research Agreement.

Owner	Percent interest
QOMPLX LLC	100%
Total	**100%**

The owner(s) of percent interest listed above in the instant application hereby disclaims, except as provided below, the terminal part of the statutory term of any patent granted on the instant application which would extend beyond the expiration date of the full statutory term of any patent granted on pending reference Application Number(s)

Application #	Filing Date

as the term of any patent granted on said reference application may be shortened by any terminal disclaimer filed prior to the grant of any patent on the pending reference application. The owner hereby agrees that any patent so granted on the instant application shall be enforceable only for and during such period that it and any patent granted on the reference application are commonly owned. This agreement runs with any patent granted on the instant application and is binding upon the grantee, its successors or assigns.
In making the above disclaimer, the owner does not disclaim the terminal part of any patent granted on the instant application that would extend to the expiration date of the full statutory term of any patent granted on said reference application, "as the term of any patent granted on said reference application may be shortened by any terminal disclaimer filed prior to the grant of any patent on the pending reference application," in the event that any such patent granted on the pending reference application: expires for failure to pay a maintenance fee, is held unenforceable, is found invalid by a court of competent jurisdiction, is statutorily disclaimed in whole or terminally disclaimed under 37 CFR 1.321, has all claims canceled by a reexamination certificate, is reissued, or is in any manner terminated prior to the expiration of its full statutory term as shortened by any terminal disclaimer filed prior to its grant.

The owner(s) of percent interest listed above in the instant application hereby disclaims, except as provided below, the terminal part of the statutory term of any patent granted on the instant application which would extend beyond the expiration date of the full statutory term of prior patent number(s)

Patent #

as the term of said prior patent is presently shortened by any terminal disclaimer. The owner hereby agrees that any patent so granted on the instant application shall be enforceable only for and during such period that it and the prior patent are commonly owned. This agreement runs with any patent granted on the instant application and is binding upon the grantee, its successors or assigns.
In making the above disclaimer, the owner does not disclaim the terminal part of the term of any patent granted on the instant application that would extend to the expiration date of the full statutory term of the prior patent, "as the term of said prior patent is presently shortened by any terminal disclaimer," in the event that said prior patent later:
- expires for failure to pay a maintenance fee;
- is held unenforceable;
- is found invalid by a court of competent jurisdiction;
- is statutorily disclaimed in whole or terminally disclaimed under 37 CFR 1.321;
- has all claims canceled by a reexamination certificate;
- is reissued; or
- is in any manner terminated prior to the expiration of its full statutory term as presently shortened by any terminal disclaimer.

Terminal disclaimer fee under 37 CFR 1.20(d) included with Electronic Terminal Disclaimer request.



Applicant claims the following entity status:
Regular Undiscounted

I hereby declare that all statements made herein of my own knowledge are true and that all statemnts made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

I certify, in accordance with 37 CFR 1.4(d)(4) that I am: An attorney or agent registered to practice before the Patent and Trademark Office who is of record in this application

Signature	Name	Registration #
/BRIAN R. GALVIN/	Brian Galvin	50966

* Statement under 37 CFR 3.73(b) is required if terminal disclaimer is signed by the assignee (owner). Form PTO/SB/96 may be used for making this certification. See MPEP 324.

Examiner-Initiated Interview Summary	**Application No.** 18/779,029	**Applicant(s)** Crabtree et al.		
	Examiner CATHERINE B THIAW	**Art Unit** 2493	**AIA (First Inventor to File) Status** Yes	**Page** 1 of 1

All Participants (applicant, applicants representative, PTO personnel)	**Title**	**Type**
CATHERINE B THIAW	Primary Examiner	Telephonic
BRIAN GALVIN	Attorney	

Date of Interview: 25 September 2024

Issues Discussed:

Proposed Amendment(s)

The examiner proposed to add the substance of claim 9 into the independent claims to place the claims in condition for allowance. The Applicant Representative submitted the amendments, included as examiner's amendments.

/Catherine Thiaw/ Primary Examiner, Art Unit 2493	

Applicant is reminded that a complete written statement as to the substance of the interview must be made of record in the application file. It is the applicants responsibility to provide the written statement, unless the interview was initiated by the Examiner and the Examiner has indicated that a written summary will be provided. See MPEP 713.04
Please further see:
MPEP 713.04
Title 37 Code of Federal Regulations (CFR) § 1.133 Interviews, paragraph (b)
37 CFR § 1.2 Business to be transacted in writing

Applicant recordation instructions: It is not necessary for applicant to provide a separate record of the substance of interview.

Examiner recordation instructions: Examiners must summarize the substance of any interview of record. A complete and proper recordation of the substance of an interview should include the items listed in MPEP 713.04 for complete and proper recordation including the identification of the general thrust of each argument or issue discussed, a general indication of any other pertinent matters discussed regarding patentability and the general results or outcome of the interview, to include an indication as to whether or not agreement was reached on the issues raised.

Notice of References Cited	Application/Control No. 18/779,029	Applicant(s)/Patent Under Reexamination Crabtree et al.	
	Examiner CATHERINE B THIAW	Art Unit 2493	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date YYYY-MM-DD	Name	CPC Classification	US Classification
*	A	US-20090327668-A1	2009-12-31	Sudzilouski; Uladzislau	G06F9/5038	712/225
*	B	US-20160275123-A1	2016-09-22	LIN; WUJUAN	G06F9/5083	1/1
*	C	US-20140351827-A1	2014-11-27	LLAMAS; Ignacio	G06F9/4881	718/107
*	D	US-9886273-B1	2018-02-06	Eldar; Alon	G06F15/80	1/1
*	E	US-20130073573-A1	2013-03-21	Huang; Wei	G06F16/254	707/774
*	F	US-6629167-B1	2003-09-30	Undy; Stephen	G06F9/3867	712/E9.046
*	G	US-9348602-B1	2016-05-24	Alapati; Nishanth	H04L67/01	1/1
*	H	US-20130067558-A1	2013-03-14	Markham; Thomas R.	H04L63/1425	726/11
*	I	US-20160323216-A1	2016-11-03	LeVasseur; Thierry	H04L51/212	1/1
*	J	US-9479720-B1	2016-10-25	Hegar; Ryan P.	H04L65/765	1/1
*	K	US-20150261580-A1	2015-09-17	Shau; Albert Yen	H04L67/1001	718/106
*	L	US-20080288621-A1	2008-11-20	Snell; Dustin M.	G06Q10/06	715/765
	M					

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date YYYY-MM-DD	Country	Name	CPC Classification
	N					
	O					
	P					
	Q					
	R					
	S					
	T					

NON-PATENT DOCUMENTS

*		Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
	U	Chambers, et al., "FlumeJava: easy, efficient data-parallel pipelines" , ACM Sigplan Notices 45.6 (2010): 363-375 (Year: 2010)
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in YYYY-MM-DD format are publication dates. Classifications may be US or foreign.

Index of Claims	Application/Control No.	Applicant(s)/Patent Under Reexamination
	18/779,029	Crabtree et al.
	Examiner	Art Unit
	CATHERINE B THIAW	2493

✓	Rejected	-	Cancelled	N	Non-Elected	A	Appeal
=	Allowed	÷	Restricted	I	Interference	O	Objected

CLAIMS

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☑ T.D. ☐ R.1.47

CLAIM		DATE								
Final	Original	09/27/2024								
1	1	=								
2	2	=								
7	3	=								
8	4	=								
3	5	=								
4	6	=								
5	7	=								
6	8	=								
-	9	-								
-	10	-								
9	11	=								
10	12	=								
13	13	=								
11	14	=								
12	15	=								
-	16	-								
-	17	-								
-	18	-								

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	18/779,029	Crabtree et al.
	Examiner	**Art Unit**
	CATHERINE B THIAW	2493

CPC						
Symbol					**Type**	**Version**
H04L	/	63	/	20	F	2013-01-01
H04L	/	63	/	1425	I	2013-01-01
G06F	/	16	/	951	I	2019-01-01
G06F	/	16	/	2477	I	2019-01-01
H04L	/	63	/	1441	I	2013-01-01
G06F	/	9	/	5038	I	2013-01-01
G06F	/	9	/	4881	A	2013-01-01

CPC Combination Sets							
Symbol					**Type**	**Set**	**Ranking**	**Version**
	/		/					

Use images from artifact folder? (•) Yes ( ) No

NONE		**Total Claims Allowed:**	
(Assistant Examiner)	(Date)	13	
/Catherine Thiaw/ Primary Examiner, Art Unit 2493	26 September 2024	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	1	7

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	18/779,029	Crabtree et al.
	Examiner	**Art Unit**
	CATHERINE B THIAW	2493

INTERNATIONAL CLASSIFICATION				
CLAIMED				
G06F9/50	/	9	/	50
G06F9/48	/	9	/	48

NON-CLAIMED				
H04L9/40	/	9	/	40
G06F16/951	/	16	/	951
G06F16/2458	/	16	/	2458

US ORIGINAL CLASSIFICATION	
CLASS	**SUBCLASS**

CROSS REFERENCES(S)						
CLASS	**SUBCLASS (ONE SUBCLASS PER BLOCK)**					

Use images from artifact folder? (•) Yes ( ) No

NONE (Assistant Examiner)	(Date)	**Total Claims Allowed:** 13	
/Catherine Thiaw/ Primary Examiner, Art Unit 2493 (Primary Examiner)	26 September 2024 (Date)	O.G. Print Claim(s) 1	O.G. Print Figure 7

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	18/779,029	Crabtree et al.
	Examiner	**Art Unit**
	CATHERINE B THIAW	2493

☐ Claims renumbered in the same order as presented by applicant ☐ CPA ☑ T.D. ☐ R.1.47

CLAIMS

Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original	Final	Original
1	1	-	10												
2	2	9	11												
7	3	10	12												
8	4	13	13												
3	5	11	14												
4	6	12	15												
5	7	-	16												
6	8	-	17												
-	9	-	18												

Use images from artifact folder? ◉ Yes ○ No

NONE (Assistant Examiner)	(Date)	**Total Claims Allowed:** 13
/Catherine Thiaw/ Primary Examiner, Art Unit 2493 (Primary Examiner)	26 September 2024 (Date)	O.G. Print Claim(s): 1 O.G. Print Figure: 7

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	18/779,029	Crabtree et al.
	Examiner	Art Unit
	CATHERINE B THIAW	2493

CPC - Searched*		
Symbol	Date	Examiner
G06F9/5038 G06F9/4881	09/26/2024	CT

CPC Combination Sets - Searched*		
Symbol	Date	Examiner

US Classification - Searched*			
Class	Subclass	Date	Examiner

* See search history printout included with this form or the SEARCH NOTES box below to determine the scope of the search.

Search Notes		
Search Notes	Date	Examiner
Pe2e search inventor/assignee name => see SEARCH-history (DP found)	09/26/2024	CT
Pe2e text+cpc search => see SEARCH-history	09/26/2024	CT
NPL search with Google Scholar => see GS-result	09/26/2024	CT

Interference Search			
US Class/CPC Symbol	US Subclass/CPC Group	Date	Examiner
G06F9	G06F9/5038 G06F9/4881	09/26/2024	CT

/Catherine Thiaw/ Primary Examiner, Art Unit 2493	

A performance study on the **vm** startup time in the cloud
M Mao, M Humphrey - 2012 IEEE Fifth International ..., 2012 - ieeexplore.ieee.org
... **hosting** front-**end** web applications and processing backend ... requests to **create VM** instances using the cloud provider API. ... a fast **start** technique by restoring previously **created VM** ...
☆ Save 99 Cite Cited by 731 Related articles All 9 versions

Autonomic virtual resource management for service **hosting** platforms
HN Van, FD Tran, JM Menaud - 2009 ICSE Workshop on ..., 2009 - ieeexplore.ieee.org
... In the **first** experiment, we nearly **do** not take into account the ... if new criteria for **VM**-to-PM **assignment** are **introduced**. Moreover ... to determine the **configuration** (**VM** quantity & placement) ...
☆ Save 99 Cite Cited by 381 Related articles All 13 versions

VM consolidation: A real case based on OpenStack Cloud
A Corradi, M Fanelli, L Foschini - Future Generation Computer Systems, 2014 - Elsevier
... **VMs**; all **VMs** run on the same one physical **host**. In particular, we **start** with an ideal situation where we allocate only one **VM** to ... **hosting** Apache HTTP Server (**configured** with the default ...
☆ Save 99 Cite Cited by 340 Related articles All 5 versions

Optimizing the migration of virtual computers
CP Sapuntzakis, R Chandra, B Pfaff, J Chow ... - ACM SIGOPS ..., 2002 - dl.acm.org
... **end**. We demonstrate these optimizations in a prototype system that uses **VMware** GSX Server virtual, machine monitor to **create** ... many **hosts** since they were **created** on different **hosts**. ...
☆ Save 99 Cite Cited by 726 Related articles All 29 versions

VMScatter: Migrate **virtual machines** to many **hosts**
L Cui, J Li, B Li, J Huai, C Hu, T Wo... - Proceedings of the 9th ..., 2013 - dl.acm.org
... construct a **complete** memory space for each **VM** at the target **host**. ... We **configure** 1GB memory for each **VM** unless specified ... We **start first** by measuring the performance on a single **VM** ...
☆ Save 99 Cite Cited by 46 Related articles All 4 versions

HPC-aware **VM** placement in infrastructure clouds
A Gupta, LV Kale, D Milojicic... - 2013 IEEE ..., 2013 - ieeexplore.ieee.org
... In these experiments, we did not **perform VM** consolidation. ... was on mapping of **VMs** to physical **hosts**, we **created** a one-to... time) in the log file, we **create** n **VMs** each with 1-core and m/n ...
☆ Save 99 Cite Cited by 142 Related articles All 12 versions

Vmplants: Providing and managing **virtual machine** execution environments for grid computing
I Krsul, A Ganguly, J Zhang, JAB Fortes... - SC'04: Proceedings ..., 2004 - ieeexplore.ieee.org
... **configuration** and creation of flexible **VMs** that, once ... as 13 **VMs** from the VMShop and all of them **end** up being **created** on ... **another host**-only network to the client and instantiate a **VM**. ...
☆ Save 99 Cite Cited by 473 Related articles All 13 versions

VCONF: a reinforcement learning approach to **virtual machines** auto-**configuration**

J Rao, X Bu, CZ Xu, L Wang, G Yin - Proceedings of the 6th International ..., 2009 - dl.acm.org

… **host** machine with two quad-core Intel Xeon CPUs and 8 GB memory, we **created** three **VMs** … Our approach has two major benefits: **First**, we **do** not assume any domain knowledge as in […

☆ Save 99 Cite Cited by 382 Related articles All 13 versions

A case for high performance computing with **virtual machines**

W Huang, J Liu, B Abali, DK Panda - Proceedings of the 20th annual ..., 2006 - dl.acm.org

… , we **start** with more discussion on the benefits of **VM** for HPC in … daemons to **enable** physical device accesses from guest **VMs**. … To **accomplish** this, **VM** image manager **hosts** a database …

☆ Save 99 Cite Cited by 457 Related articles All 14 versions

Implementation and performance analysis of various **VM** placement strategies in CloudSim

MR Chowdhury, MR Mahmud, RM Rahman - Journal of Cloud Computing, 2015 - Springer

… migration of **VMs** from a **host**, a heuristic for **setting** an upper … will place **VMs** from **VM** migration list to **host** which has **second** … After we get our desired clusters, we will **start** allocating **VMs** …

☆ Save 99 Cite Cited by 129 Related articles All 10 versions

PE2E SEARCH - Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
L2	69	((("QOMPLX") near3 ("LLC"))).AS,AANM.	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/19 12:55 PM
L3	3	((("QOMPLX") near3 ("LLC"))).AS,AANM. AND (pipeline).clm.	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/19 12:59 PM
L4	413	((("CRABTREE") near3 ("Jason")) OR (("SELLERS") near3 ("Andrew"))).INV.	(US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT)	OR	ON	ON	2024/09/19 01:37 PM
L5	19	((("CRABTREE") near3 ("Jason")) OR (("SELLERS") near3 ("Andrew"))).INV. AND pipeline.clm.	(US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT)	OR	ON	ON	2024/09/19 01:38 PM
L6	250	("10009378" OR "10027711" OR "10038559" OR "10044675" OR "10050985" OR "10055473" OR "10061635" OR "10074052" OR "10078664" OR "10083236" OR "10102480" OR "10108907" OR "10109014" OR "10110415" OR "10120907" OR "10122687" OR "10122764" OR "10146592" OR "10152676" OR "10154049" OR "10162969" OR "10168691" OR "10180780" OR "10185832" OR "10191768" OR "10205735" OR "10210246" OR "10210470" OR "10212176" OR "10212184" OR "10216485" OR "10217348" OR "10261763" OR "10275545" OR "10277629" OR "10284570" OR "10289841" OR	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/19 01:51 PM

		"10290141" OR "10298607" OR "10305902" OR "10318739" OR "10318882" OR "10320828" OR "10321278" OR "10324773" OR "10338913" OR "10367829" OR "10380140" OR "10387124" OR "10387631" OR "10410113" OR "10410214" OR "10438001" OR "10440054" OR "10445482" OR "10452664" OR "10462112" OR "10505954" OR "10511498" OR "10515062" OR "10515366" OR "10530796" OR "10540624" OR "10579691" OR "10601854" OR "10606454" OR "10609059" OR "10628578" OR "10643144" OR "10645086" OR "10645100" OR "10673880" OR "10715534" OR "10740358" OR "10764321" OR "10776847" OR "10789367" OR "10791131" OR "10817530" OR "10861028" OR "10862916" OR "10871951" OR "10911470" OR "10944772" OR "10958667" OR "10965711" OR "10977551" OR "10985997" OR "10992698" OR "11030520" OR "11032307" OR "11113667" OR "11138514" OR "11194900" OR "11256791" OR					

		"11334831" OR "11392875" OR "11477641" OR "11539663" OR "11574206" OR "11736299" OR "20040255167" OR "20050071223" OR "20050165822" OR "20050198099" OR "20050289072" OR "20060149575" OR "20070055558" OR "20070136821" OR "20070150744" OR "20070168370" OR "20070276714" OR "20080021866" OR "20080270203" OR "20090012760" OR "20090094372" OR "20090199002" OR "20090319247" OR "20100083240" OR "20100115276" OR "20100275183" OR "20100299651" OR "20100325685" OR "20110087888" OR "20110225287" OR "20110307467" OR "20120215575" OR "20120296845" OR "20130046751" OR "20130117831" OR "20130132149" OR "20130159219" OR "20140082729" OR "20140149186" OR "20140244612" OR "20140279762" OR "20140324521" OR "20150081363" OR "20150128258" OR "20150149979" OR "20150161738" OR "20150170053" OR "20150172311" OR "20150242509" OR "20150281225" OR "20150295775" OR "20150317745" OR "20150347414" OR "20150379111" OR "20160004858" OR "20160006629" OR "20160012235" OR "20160057159" OR				

		"20160088000" OR "20160099960" OR "20160119365" OR "20160140519" OR "20160180240" OR "20160219066" OR "20160275123" OR "20160330233" OR "20160364307" OR "20160371363" OR "20170010589" OR "20170023509" OR "20170090893" OR "20170207926" OR "20170241791" OR "20180268264" OR "20180336250" OR "20190188797" OR "20200004752" OR "20200177618" OR "20200304534" OR "20200356664" OR "20200364346" OR "20200396246" OR "20210075822" OR "3370192" OR "5669000" OR "5953011" OR "6256544" OR "6477572" OR "6857073" OR "6906709" OR "7072863" OR "7139747" OR "7171515" OR "7222366" OR "7227948" OR "7266821" OR "7281125" OR "7310632" OR "7373524" OR "7437718" OR "7448046" OR "7480940" OR "7493593" OR "7530105" OR "7546207" OR "7546333" OR "7546637" OR "7603709" OR "7603714" OR "7653188" OR "7657406" OR "7660815" OR "7685296" OR "7698213" OR "7702821" OR "7739653" OR					

		"7743421" OR "7774335" OR "7818224" OR "7818417" OR "7840677" OR "7925561" OR "7933926" OR "8006303" OR "8055712" OR "8065257" OR "8069190" OR "8116450" OR "8132260" OR "8156029" OR "8205259" OR "8209274" OR "8245302" OR "8346753" OR "8352347" OR "8352412" OR "8370192" OR "8380843" OR "8386519" OR "8407800" OR "8417656" OR "8457996" OR "8495521" OR "8516594" OR "8516596" OR "8548777" OR "8566945" OR "8583639" OR "8595240" OR "8601554" OR "8601587" OR "8607197" OR "8615800" OR "8654127" OR "8677473").pn.					
L7	149	("8707275" OR "8712596" OR "8725597" OR "8726393" OR "8751867" OR "8752178" OR "8781990" OR "8782080" OR "8788306" OR "8793758" OR "8806361" OR "8813234" OR "8819772" OR "8826426" OR "8839440" OR "8897900" OR "8898442" OR "8914878" OR	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/19 01:51 PM

		"8949960" OR "8959494" OR "8990392" OR "8997233" OR "9009837" OR "9015708" OR "9021477" OR "9031870" OR "9043332" OR "9049207" OR "9069725" OR "9092616" OR "9100430" OR "9110706" OR "9129108" OR "9134966" OR "9137024" OR "9152727" OR "9166990" OR "9171079" OR "9185124" OR "9202040" OR "9203827" OR "9210185" OR "9231962" OR "9235732" OR "9253643" OR "9256735" OR "9262787" OR "9264395" OR "9276951" OR "9286103" OR "9292692" OR "9292699" OR "9294498" OR "9300682" OR "9319430" OR "9336481" OR "9338061" OR "9344444" OR "9349103" OR "9369482" OR "9384345" OR "9390376" OR "9400962" OR "9438616" OR "9461876" OR "9466041" OR "9467461" OR "9495188" OR "9501647" OR "9503467" OR "9503472" OR "9509716" OR "9515826" OR "9516053" OR "9521166" OR "9541982" OR					

		"9558220" OR "9560065" OR "9565204" OR "9571517" OR "9578046" OR "9596141" OR "9600792" OR "9602513" OR "9602529" OR "9602530" OR "9609009" OR "9609015" OR "9619291" OR "9639575" OR "9652538" OR "9652604" OR "9654495" OR "9661019" OR "9667600" OR "9667641" OR "9672283" OR "9672355" OR "9674211" OR "9674249" OR "9679125" OR "9680867" OR "9686293" OR "9690645" OR "9699205" OR "9712553" OR "9721086" OR "9729421" OR "9729538" OR "9734169" OR "9734220" OR "9736173" OR "9749343" OR "9749344" OR "9753796" OR "9756067" OR "9762443" OR "9771225" OR "9772934" OR "9774407" OR "9774522" OR "9774616" OR "9781144" OR "9807104" OR "9832213" OR "9842000" OR "9858322" OR "9860208" OR "9875360" OR "9882929" OR "9887933" OR "9910993" OR "9911088" OR "9917860" OR					

		"9928366" OR "9930058" OR "9942295" OR "9946517" OR "9952899" OR "9954879" OR "9954884" OR "9965627" OR "9967264" OR "9967265" OR "9967282" OR "9967283" OR "9967625" OR "9984129" OR "9992228").pn.					
L8	2342	(input stream) WITH first near4 pipeline AND second near4 pipeline AND (cluster system) WITH pipeline	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/19 03:03 PM
L9	1012	9 AND (@ad<"20151028")	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/19 03:04 PM
L10	11	10 AND (G06F9/5038 OR G06F9/4881).cpc.	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/19 03:05 PM
L11	69	(workflow pipeline) WITH series WITH (first second) WITH (cluster farm set) WITH (comput$3 device)	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/19 03:43 PM
L12	0	11 AND 12	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/19 03:46 PM
L13	33	12 AND (@ad<"20151028")	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/19 03:46 PM
L14	72	first near4 (pipeline workflow) WITH (output result outcome) WITH (feed$3 fed) WITH second near4 (pipeline workflow)	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/19 08:58 PM
L15	31	15 AND (@ad<"20151028")	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/19 08:58 PM
L16	987	first near4 pipeline WITH first near4 (computing device node) AND second near4 pipeline WITH second near4 (computing device node) AND ($6connect$3 communicat$3) WITH pipeline	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/19 10:21 PM
L17	138	(pipeline workflow) WITH (output result	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/19 10:21 PM

		outcome) WITH (feed$3 fed) WITH second near4 (pipeline workflow)					
L18	12	17 AND 18	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/19 10:21 PM
L19	9	19 AND (@ad<"20151028")	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/19 10:22 PM
L20	71	18 AND (@ad<"20151028")	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/19 10:25 PM
L21	0	21 AND (H04L63/20 OR G06F9/5038 OR G06F16/2477 OR G06F16/951 OR H04L63/1425 OR H04L63/1441 OR G06F9/4881).cpc.	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/19 10:25 PM
L22	0	21 AND (G06F9/5??? OR G06F9/4???).cpc.	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/19 10:27 PM
L23	16	21 AND (G06F9?????).cpc.	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/19 10:28 PM
L24	337	17 AND (@ad<"20151028")	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/19 10:51 PM
L25	27	25 AND (G06F9?????).cpc.	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/19 10:51 PM
L26	8745	(cluster set) WITH (comput$3 system device node) WITH pipeline AND ($6connect$3 communicat$3) WITH pipeline	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/20 09:13 AM
L27	2833	27 AND (@ad<"20151028")	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/20 09:18 AM
L28	58	28 AND (H04L63/20 OR G06F9/5038 OR G06F16/2477 OR G06F16/951 OR H04L63/1425 OR H04L63/1441 OR G06F9/4881).cpc.	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/20 09:18 AM
L29	305	monitor$3 WITH (workflow pipeline) AND (determin$4 evaluat$4 whether) WITH (additional new) near5 pipeline	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/20 12:05 PM
L30	111	30 AND (@ad<"20151028")	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/20 12:05 PM
L31	3	31 AND (H04L63/20 OR G06F9/5038 OR G06F16/2477 OR	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/20 12:08 PM

		G06F16/951 OR H04L63/1425 OR H04L63/1441 OR G06F9/4881).cpc.					
L32	3	monitor$3 WITH (workflow flow pipeline) WITH (cluster set) WITH (machine VM device comput$3) SAME (generat$3 creat$3 establish$3) WITH new near5 (workflow flow pipeline) WITH (second$3 other) WITH (cluster set) WITH (machine VM device comput$3)	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/20 03:36 PM
L33	611	(inter cross) WITH (cluster cloud) WITH (task $5flow pipeline) WITH (execut$3 process$4)	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/20 04:00 PM
L34	144	34 AND (@ad<"20151028")	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/20 04:01 PM
L35	4	35 AND (H04L63/20 OR G06F9/5038 OR G06F16/2477 OR G06F16/951 OR H04L63/1425 OR H04L63/1441 OR G06F9/4881).cpc.	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/20 04:01 PM
L36	0	31 AND 35	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/20 04:16 PM
L37	267	new near5 (workflow flow pipeline) WITH (second$3 other) WITH (cluster set) WITH (machine VM device comput$3)	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/20 04:16 PM
L38	2	35 AND 38	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/20 04:16 PM
L39	47	(Artificial Intelligence) Similar to: Unpublished Application # with 0 CPC Selections and 0 Text Selections Text: (US-20240195842-A1 OR US-20170124464-A1 OR WO-2017075543-A1 OR WO-2017189641-A1 OR US-11055630-B2 OR US-20210406768-	(US-PGPUB; USPAT; USOCR; FIT (CN, FR, WO))	OR	ON	ON	2024/09/20 08:12 PM

		A1 OR CN-109643285-A OR US-20240028607-A1 OR US-10055426-B2 OR US-20170371726-A1 OR US-20170139952-A1 OR US-11169959-B2 OR US-11663228-B2 OR US-20170139929-A1 OR US-20230020163-A1 OR US-20220269978-A1 OR US-10037329-B2 OR US-20230368076-A1 OR US-10127244-B2 OR US-20240231931-A1 OR US-11755957-B2 OR US-10628217-B1 OR US-20120124564-A1 OR US-20180004835-A1 OR US-11720536-B1 OR US-8402450-B2 OR WO-2012067803-A2 OR US-20200279173-A1 OR US-20230140169-A1 OR CA-3022373-C OR US-20240202602-A1 OR US-10459979-B2 OR US-9983859-B2 OR US-20180341658-A1 OR US-20220075795-A1 OR US-20180007145-A1 OR WO-2017189816-A1 OR US-20220100771-A1 OR AU-2017258180-B2 OR EP-4379568-A1 OR US-20170315791-A1 OR US-11347548-B2 OR US-11507858-B2 OR US-20230121209-A1 OR US-20210141674-A1 OR US-20230035310-A1 OR US-20200241920-A1 OR US-20190347135-A1 OR US-10901795-B2 OR US-11762701-B2).did.					
L40	3	40 AND (@ad<"20151028")	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/20 08:13 PM
L41	112	remote$2 WITH monitor$4 WITH	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/21 04:25 PM

		($6flow execution) AND (initiat$3 start$3 configur$5) WITH new near5 (thread pipeline task)					
L42	56	42 AND (@ad<"20151028")	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/21 04:25 PM
L43	3007	(multi parallel$2) WITH (process$3 execut$3) WITH (task job) WITH (first second multiple plurality) WITH cluster	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/23 04:04 PM
L44	0	new near5 pipeline WITH (task job) WITH (first second other) near5 cluster	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/23 04:06 PM
L45	100	new near5 (pipeline task job) WITH (first second other) near5 cluster	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/23 04:06 PM
L46	5	45 AND 47	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/23 04:07 PM
L47	31	47 AND (@ad<"20151028")	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/23 04:24 PM
L49	69	((("QOMPLX") near3 ("LLC"))).AS,AANM.	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/26 01:42 PM
L50	0	((("QOMPLX") near3 ("LLC"))).AS,AANM. AND (first near5 pipeline AND second near5 pipeline AND first ADJ computer AND second ADJ computer).clm.	(US-PGPUB; USPAT; USOCR; FIT (CN, FR, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2024/09/26 01:43 PM
L51	413	((("CRABTREE") near3 ("Jason")) OR (("SELLERS") near3 ("Andrew"))).INV.	(US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT)	OR	ON	ON	2024/09/26 01:43 PM
L52	0	((("CRABTREE") near3 ("Jason")) OR (("SELLERS") near3 ("Andrew"))).INV. AND (first near5 pipeline AND second near5 pipeline AND first ADJ computer AND second ADJ computer).clm.	(US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT)	OR	ON	ON	2024/09/26 01:43 PM
L53	5	((("CRABTREE") near3 ("Jason")) OR (("SELLERS") near3 ("Andrew"))).INV. AND (cluster AND pipeline).clm.	(US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT)	OR	ON	ON	2024/09/26 01:44 PM

L54	0	((("QOMPLX") near3 ("LLC"))).AS,AANM. AND (cluster AND pipeline).clm.	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/26 01:51 PM
L55	130	(pipeline AND first ADJ computer AND second ADJ computer).clm.	(US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT)	OR	ON	ON	2024/09/26 01:52 PM
L56	1	55 AND (G06F9/5038 OR G06F9/4881).cpc.	(US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT)	OR	ON	ON	2024/09/26 01:54 PM
L60	6	fail$3 NEAR over WITH (pipeline workflow) WITH cluster	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/26 04:10 PM

PE2E SEARCH - Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
N1	3	((("QOMPLX") near3 ("LLC"))).AS,AANM. AND (pipeline).clm.	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/24 12:06 PM
N2	19	((("CRABTREE") near3 ("Jason")) OR (("SELLERS") near3 ("Andrew"))).INV. AND pipeline.clm.	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/24 12:07 PM
N3	1998	(G06F9/5038 AND G06F9/4881).cpc.	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/24 12:08 PM
N4	7	N3 AND ((workflow pipeline) WITH cluster).clm.	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/24 12:09 PM
N5	21	(first near5 pipeline AND second near5 pipeline AND first ADJ computer AND second ADJ computer).clm.	(US-PGPUB; USPAT)	OR	ON	ON	2024/09/24 12:20 PM

Pipeline architecture

CV Ramamoorthy, HF Li - ACM Computing Surveys (CSUR), 1977 - dl.acm.org

... After the analytical evaluation of a **pipeline's** performance, the ... requires internal looping, ie **results** being **fed** back as inputs. ... called shortstop is established between the **output** (**transmit** ...

☆ Save 99 Cite Cited by 354 Related articles All 7 versions

A novel continuous time representation for the scheduling of **pipeline** systems with pumping yield rate constraints

R Rejowski Jr, JM Pinto - Computers & Chemical Engineering, 2008 - Elsevier

... segments into packs of equal size, whereas the **second** one divides them into packs of ... consider the **pipeline feed** as well as the timing constraints imposed by the refinery **production**. ...

☆ Save 99 Cite Cited by 175 Related articles All 3 versions

Reducing the branch penalty in pipelined processors

DJ Lilja - Computer, 1988 - ieeexplore.ieee.org

... **pipeline** and prefetching begins from the new path. If the instruction is not in the BTB, however, the **pipeline** is ... If you are interested in exploring **employment** opportunities with us, **send** a ...

☆ Save 99 Cite Cited by 158 Related articles All 7 versions

Pipeline vectorization

M Weinhardt, W Luk - ... on Computer-Aided Design of Integrated ..., 2001 - ieeexplore.ieee.org

... Since most **pipelines** are **fed** by data from external memory, ... of the **second** loop. The alignment parameter is chosen as the ... synthesized for the selected candidates and **output** to a **file**. ...

☆ Save 99 Cite Cited by 252 Related articles All 10 versions

The optimal logic depth per **pipeline** stage is 6 to 8 FO4 inverter delays

MS Hrishikesh, D Burger, NP Jouppi ... - ACM SIGARCH ..., 2002 - dl.acm.org

... The **results** of this study have significant implications for ... gate followed by an inverter and a **feed**-back path. Data values are ... For example, the access latency of the register **file** is 0.39ns ...

☆ Save 99 Cite Cited by 367 Related articles All 43 versions

FlumeJava: easy, efficient data-parallel **pipelines**

C Chambers, A Raniwala, F Perry, S Adams... - ACM Sigplan ..., 2010 - dl.acm.org

... determines which **output file** shard will hold that key's **results**. ... of these two operations are flatten()ed together and **fed** into ... of operations, the optimizer takes less than a **second** or two. ...

☆ Save 99 Cite Cited by 813 Related articles All 20 versions

Pipelined asynchronous circuits

AM Lines - Master's thesis, California Institute of ..., 1995 - authors.library.caltech.edu

... A **\pipeline"** is a linear sequence of bu ers where the **output** ... 4 or 8 bits) , and **send** bu ered copies of the control tokens to ... no new circuits, is to **feed** an **output** of a normal pipelined cell ...

☆ Save 99 Cite Cited by 307 Related articles All 5 versions

Micropipelines

IE Sutherland - Communications of the ACM, 1989 - dl.acm.org

… I will describe **pipelines** in their complete form later, but **first** I will focus on … **output** rate of an inelastic **pipeline** must match exactly. Stripped of any processing logic, an inelastic **pipeline** …

☆ Save 99 Cite Cited by 2077 Related articles All 8 versions

On pipelining dynamic instruction scheduling logic

J Stark, MD Brown, YN Patt - Proceedings of the 33rd annual ACM/IEEE …, 2000 - dl.acm.org

… of a baseline machine that **pipelines** the scheduling logic but … **file** or bypassed from earlier instructions in the **pipeline**. After … the tag bus, which is then **fed** to the tag comparators of the …

☆ Save 99 Cite Cited by 185 Related articles All 14 versions

Time dependent processing in a parallel **pipeline** architecture

J Biddiscombe, B Geveci, K Martin, … - … on Visualization and …, 2007 - ieeexplore.ieee.org

… , it also has the novel ability to **send** the algorithm to the data, … their inputs and **feed** the result to their **outputs**. However, … all we need do is add a **second** data path where the the DATA …

☆ Save 99 Cite Cited by 61 Related articles All 19 versions

CodePipe: An opportunistic **feeding** and routing protocol for reliable multicast with pipelined network coding
P Li, S Guo, S Yu, AV Vasilakos - 2012 Proceedings IEEE ..., 2012 - ieeexplore.ieee.org
... start to **send** the coded packets of a new batch. Multiple batches form as a **pipeline** for data
... In all cases, our **results** are averaged over 20 randomly generated network examples. We ...
☆ Save 99 Cite Cited by 200 Related articles All 9 versions

Planning and sequencing product distribution in a real-world **pipeline** network: an MILP decomposition approach
SNB Magatão, L Magatão, H Luis Polli... - Industrial & ..., 2013 - ACS Publications
... Each refinery and depot can also receive/**send** different ... These **results** are **fed** into the
next block, the assignment and ... The **second** route is defined from the area of the surge tank ...
☆ Save 99 Cite Cited by 35 Related articles All 5 versions

VISTA data flow system: **pipeline** processing for WFCAM and VISTA
MJ Irwin, J Lewis, S Hodgkin... - ... scientific return for ..., 2004 - spiedigitallibrary.org
... typically gives an overall factor of 3–4 reduction in **file** size. ... that in normal use and then **fed**
back to the controlling system for ... Interleaving **results** in a new confidence map for the **output** ...
☆ Save 99 Cite Cited by 307 Related articles All 9 versions

An industrial big data **pipeline** for data-driven analytics maintenance applications in large-scale smart manufacturing facilities
P O'Donovan, K Leahy, K Bruton, DTJ O'Sullivan - Journal of big data, 2015 - Springer
... of a relational database or flat log **file**. Once data is persisted in ... The qualitative **results** in
this research include a set of ... we plan to use the data **pipeline** to **feed** predictive maintenance ...
☆ Save 99 Cite Cited by 373 Related articles All 13 versions

Feeding the **pipeline**: Gender, occupational plans, and college major selection
SL Morgan, D Gelbgiser, KA Weeden - Social science research, 2013 - Elsevier
... are quite large, would be larger still if our **first** category was limited to STEM majors. In the ...
pathway toward medical **careers**. **Second**, as we noted above, our **results** are weighted by the ...
☆ Save 99 Cite Cited by 422 Related articles All 14 versions

Reliable multicast with pipelined network coding using opportunistic **feeding** and routing
P Li, S Guo, S Yu, AV Vasilakos - IEEE Transactions on Parallel ..., 2014 - ieeexplore.ieee.org
... source can start to **send** coded packets of the next batch, which forms as a **pipeline** for batch
... inter-batch coding are created by using the **results** of linear programming LP3, which can be ...
☆ Save 99 Cite Cited by 201 Related articles All 10 versions

A feedback-driven proportion allocator for real-rate scheduling

DC Steere, A Goel, J Gruenberg, D McNamee, C Pu, ... - OSDI, 1999 - usenix.org

... by **feed**back is necessary because the needs of **jobs**, and the ... appears to **work** well for our video **pipeline** application, we ... **Jobs submit** units of **work** to be scheduled, and the scheduler ...

☆ Save 99 Cite Cited by 442 Related articles All 28 versions »

CloudMap: a cloud-based **pipeline** for analysis of mutant genome sequences

G Minevich, DS Park, D Blankenberg, RJ Poole, ... - Genetics, 2012 - academic.oup.com

... reads generated by **second** generation sequencing platforms... VCF **output file** as an input to the parameter "**Output** variants ... , we **feed** an alignment (BAM) **file** and a VCF reference **file** of ...

☆ Save 99 Cite Cited by 342 Related articles All 12 versions

Subsea **pipeline** design, analysis, and installation

Q Bai, Y Bai - 2014 - books.google.com

... Alternatively you can **submit** your request online by visiting the ... Better quality control in **pipeline production** allows more ... The **second** high stress area is in the sag bend, where the ...

☆ Save 99 Cite Cited by 275 Related articles »

Decoupling algorithms from schedules for easy optimization of image processing **pipelines**

J Ragan-Kelley, A Adams, S Paris, M Levoy... - ACM Transactions on ..., 2012 - dl.acm.org

... a representation for **feed**-forward imaging **pipelines** that ... , starting from the **output** of the full **pipeline**. Formally, when a ... it can be compiled to an object **file** and header to be used by ...

☆ Save 99 Cite Cited by 344 Related articles All 11 versions

Doc code: IDS
Doc description: Information Disclosure Statement (IDS) Filed

PTO/SB/08a (01-22)
Approved for use through 05/31/2024. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(Not for submission under 37 CFR 1.99)

Application Number	
Filing Date	2024-07-21
First Named Inventor	Jason Crabtree
Art Unit	
Examiner Name	
Attorney Docket Number	15305FTNK

U.S.PATENTS Remove

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1					

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS Remove

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1					

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2] i	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1							☐

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	1	Excerpts of raw documentation source located in the Apache Logging Flume GitHub "apache/logging-flume" repository tagged as Apache Flume Release 1.6.0, dated May 5, 2015, full release available at https://github.com/apache/logging-flume/tree/release-1.6.0, 200 pages	☐
	2	Excerpts of raw documentation source located in the Apache Airflow GitHub "apache/airflow" repository tagged as Apache Airflow Release v1.5.1, dated September 4, 2015, full release available at https://github.com/apache/airflow/tree/1.5.1, 75 pages	☐
	3	Excerpts of raw documentation source located in the Apache Apex core GitHub "apache/apex-core" repository tagged as Apache Apex Core Release v3.1.1, dated October 8, 2015, full release available at https://github.com/apache/apex-core/tree/v3.1.1, 158 pages	☐
	4	Excerpts of raw documentation source located in the Apache Beam GitHub "apache/beam" repository tagged as Apache Beam Release v1.2.0, dated October 5, 2015, full release available at https://github.com/apache/beam/tree/v1.2.0, 31 pages	☐
	5	Excerpts of raw documentation source located in the Apache Flink GitHub "apache/flink" repository tagged as Apache Flink Release 0.9.7, dated August 27, 2015, full release available at https://github.com/apache/flink/tree/release-0.9.1, 712 pages	☐
	6	Excerpts of raw documentation source located in the Apache Hadoop GiHub "apache/hadoop" repository tagged as Apache Hadoop Release 2.6.1, dated September 23, 2015, full release available at https://github.com/apache/hadoop/tree/release-2.6.1, 1640 pages	☐
	7	Excerpts of raw documentation source located in the Apache Ignite GitHub "apache/ignite" repository tagged as Apache Ignite Release 1.4.1, dated September 25, 2015, full release available at https://github.com/apache/ignite/tree/1.4.1, 103 pages	☐
	8	Excerpts of raw documentation source located in the Apache Kafka GitHub "apache/kafka" repository tagged as Apache Kafka Release 0.8.2.2, dated September 2, 2015, full release available at https://github.com/apache/kafka/tree/0.8.2.2, 13 pages	☐
	9	Excerpts of raw documentation source located in the Apache NiFi GitHub "apache/nifi" repository tagged as Apache NiFi Release nifi-0.3.0-RC1, dated September 14, 2015, full release available at https://github.com/apache/nifi/tree/nifi-0.3.0-RC1, 286 pages	☐
	10	Excerpts of raw documentation source located in the Apache Samza GitHub "apache/samza" repository tagged as Apache Samza Release 0.9.1-rc1, date June 23, 2015, full release available at https://github.com/apache/samza/tree/release-0.9.1-rc1, 221 pages	☐
	11	Excerpts of raw documentation source located in the Apache Spark GitHub "apache/spark" repository tagged as Apache Spark Release v.15.1-rc1, dated September 23, 2015, full release available at https://github.com/apache/spark/tree/v1.5.1, 2122 pages	☐

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	12	Excerpts of raw documentation source located in the Apache Storm GitHub "apache/storm" repository tagged as Apache Storm Release v0.10.0-beta1, dated June 19, 2015, full release available at https://github.com/apache/storm/tree/v0.10.0-beta1, 589 pages	☐

If you wish to add additional non-patent literature document citation information please click the Add button Add

EXAMINER SIGNATURE

Examiner Signature	/Catherine Thiaw/	Date Considered	09/26/2024

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(**Not for submission under 37 CFR 1.99)**

Application Number	
Filing Date	2024-07-21
First Named Inventor	Jason Crabtree
Art Unit	
Examiner Name	
Attorney Docket Number	15305FTNK

CERTIFICATION STATEMENT

Please see 37 CFR 1.97(e) to make the appropriate selection(s):

☐ That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

☐ See attached certification statement.

☒ A certification statement is not submitted herewith.

FEE

☐ The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

COPIES

☐ An identification of an earlier application pursuant to 37 CFR 1.98 (d) (1) is attached.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/BRIAN R. GALVIN/	Date (YYYY-MM-DD)	2024-07-19
Name/Print	Brian R. Galvin	Registration Number	50966

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these record s.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

Doc code: IDS
Doc description: Information Disclosure Statement (IDS) Filed

PTO/SB/08a (01-22)
Approved for use through 05/31/2024. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (**Not for submission under 37 CFR 1.99)**		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

U.S.PATENTS Remove

Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	3370192	A	1968-02-20	SCHWARTZ JAMES W et al.	Entire Document
	2	5669000	A	1997-09-16	Jessen et al.	Entire Document
	3	5953011	A	1999-09-14	Matsuoka	Entire Document
	4	6256544	B1	2001-07-03	Weissinger	Entire Document
	5	6477572	B1	2002-11-05	Elderton et al.	Entire Document
	6	6857073	B2	2005-02-15	French et al.	Entire Document
	7	6906709	B1	2005-06-14	Larkin et al.	Entire Document
	8	7072863	B1	2006-07-04	Phillips et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	9	7139747	B1	2006-11-21	Najork	Entire Document
	10	7171515	B2	2007-01-30	Ohta et al.	Entire Document
	11	7222366	B2	2007-05-22	Bruton, III et al.	Entire Document
	12	7227948	B2	2007-06-05	Ohkuma et al.	Entire Document
	13	7266821	B2	2007-09-04	Polizzi et al.	Entire Document
	14	7281125	B2	2007-10-09	Challener et al.	Entire Document
	15	7310632	B2	2007-12-18	Meek et al.	Entire Document
	16	7373524	B2	2008-05-13	Motsinger et al.	Entire Document
	17	7437718	B2	2008-10-14	Fournet et al.	Entire Document
	18	7448046	B2	2008-11-04	Navani et al.	Entire Document
	19	7480940	B1	2009-01-20	Agbabian et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	20	7493593	B2	2009-02-17	Koehler	Entire Document
	21	7530105	B2	2009-05-05	Gilbert et al.	Entire Document
	22	7546207	B2	2009-06-09	Nix et al.	Entire Document
	23	7546333	B2	2009-06-09	Alon et al.	Entire Document
	24	7546637	B1	2009-06-09	Agbabian et al.	Entire Document
	25	7603709	B2	2009-10-13	Lewis et al.	Entire Document
	26	7603714	B2	2009-10-13	Johnson et al.	Entire Document
	27	7653188	B2	2010-01-26	Kloberdans et al.	Entire Document
	28	7657406	B2	2010-02-02	Tolone et al.	Entire Document
	29	7660815	B1	2010-02-09	Scofield et al.	Entire Document
	30	7685296	B2	2010-03-23	Brill et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	31	7698213	B2	2010-04-13	Lancaster	Entire Document
	32	7702821	B2	2010-04-20	Feinberg et al.	Entire Document
	33	7739653	B2	2010-06-15	Venolia	Entire Document
	34	7743421	B2	2010-06-22	Cosquer et al.	Entire Document
	35	7774335	B1	2010-08-10	Scofield et al.	Entire Document
	36	7818224	B2	2010-10-19	Boerner	Entire Document
	37	7818417	B2	2010-10-19	Ginis et al.	Entire Document
	38	7840677	B2	2010-11-23	Li et al.	Entire Document
	39	7925561	B2	2011-04-12	Xu	Entire Document
	40	7933926	B2	2011-04-26	Ebert	Entire Document
	41	8006303	B1	2011-08-23	Dennerline et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	42	8055712	B2	2011-11-08	Kagawa et al.	Entire Document
	43	8065257	B2	2011-11-22	Kuecuekyan	Entire Document
	44	8069190	B2	2011-11-29	McColl et al.	Entire Document
	45	8116450	B2	2012-02-14	Agrawal et al.	Entire Document
	46	8132260	B1	2012-03-06	Mayer et al.	Entire Document
	47	8156029	B2	2012-04-10	Szydlo	Entire Document
	48	8205259	B2	2012-06-19	Stute	Entire Document
	49	8209274	B1	2012-06-26	Lin et al.	Entire Document
	50	8245302	B2	2012-08-14	Evans et al.	Entire Document
	51	8346753	B2	2013-01-01	Hayes	Entire Document
	52	8352347	B2	2013-01-08	Howard et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	53	8352412	B2	2013-01-08	Alba et al.	Entire Document
	54	8370192	B2	2013-02-05	Deo et al.	Entire Document
	55	8380843	B2	2013-02-19	Loizeaux et al.	Entire Document
	56	8386519	B2	2013-02-26	Kenedy et al.	Entire Document
	57	8407800	B2	2013-03-26	Schloegel et al.	Entire Document
	58	8417656	B2	2013-04-09	Beg et al.	Entire Document
	59	8457996	B2	2013-06-04	Winkler et al.	Entire Document
	60	8495521	B2	2013-07-23	Fried	Entire Document
	61	8516594	B2	2013-08-20	Bennett et al.	Entire Document
	62	8516596	B2	2013-08-20	Sandoval et al.	Entire Document
	63	8548777	B2	2013-10-01	Sturrock et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	64	8566945	B2	2013-10-22	Sima	Entire Document
	65	8583639	B2	2013-11-12	Chitnis et al.	Entire Document
	66	8595240	B1	2013-11-26	Otey et al.	Entire Document
	67	8601554	B2	2013-12-03	Gordon et al.	Entire Document
	68	8601587	B1	2013-12-03	Powell et al.	Entire Document
	69	8607197	B2	2013-12-10	Barcia et al.	Entire Document
	70	8615800	B2	2013-12-24	Baddour et al.	Entire Document
	71	8654127	B2	2014-02-18	Kenttala et al.	Entire Document
	72	8677473	B2	2014-03-18	Dennerline et al.	Entire Document
	73	8707275	B2	2014-04-22	Mascaro et al.	Entire Document
	74	8712596	B2	2014-04-29	Scott	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	75	8725597	B2	2014-05-13	Mauseth et al.	Entire Document
	76	8726393	B2	2014-05-13	Macy et al.	Entire Document
	77	8751867	B2	2014-06-10	Marvasti et al.	Entire Document
	78	8752178	B2	2014-06-10	Coates et al.	Entire Document
	79	8781990	B1	2014-07-15	de Alfaro et al.	Entire Document
	80	8782080	B2	2014-07-15	Lee et al.	Entire Document
	81	8788306	B2	2014-07-22	Delurgio et al.	Entire Document
	82	8793758	B2	2014-07-29	Raleigh et al.	Entire Document
	83	8806361	B1	2014-08-12	Noel et al.	Entire Document
	84	8813234	B1	2014-08-19	Bowers et al.	Entire Document
	85	8819772	B2	2014-08-26	Bettini et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	86	8826426	B1	2014-09-02	Dubey	Entire Document
	87	8839440	B2	2014-09-16	Yun et al.	Entire Document
	88	8897900	B2	2014-11-25	Smith et al.	Entire Document
	89	8898442	B2	2014-11-25	Stoitsev	Entire Document
	90	8914878	B2	2014-12-16	Burns et al.	Entire Document
	91	8949960	B2	2015-02-03	Berkman et al.	Entire Document
	92	8959494	B2	2015-02-17	Howard	Entire Document
	93	8990392	B1	2015-03-24	Stamos	Entire Document
	94	8997233	B2	2015-03-31	Green et al.	Entire Document
	95	9009837	B2	2015-04-14	Nunez Di Croce	Entire Document
	96	9015708	B2	2015-04-21	Choudhury et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	97	9021477	B2	2015-04-28	Choudhury et al.	Entire Document
	98	9031870	B2	2015-05-12	Kenedy et al.	Entire Document
	99	9043332	B2	2015-05-26	Noel et al.	Entire Document
	100	9049207	B2	2015-06-02	Hugard, IV et al.	Entire Document
	101	9069725	B2	2015-06-30	Jones	Entire Document
	102	9092616	B2	2015-07-28	Kumar et al.	Entire Document
	103	9100430	B1	2015-08-04	Seiver et al.	Entire Document
	104	9110706	B2	2015-08-18	Yu et al.	Entire Document
	105	9129108	B2	2015-09-08	Drissi et al.	Entire Document
	106	9134966	B2	2015-09-15	Brock et al.	Entire Document
	107	9137024	B2	2015-09-15	Swingler et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	108	9152727	B1	2015-10-06	Balducci et al.	Entire Document
	109	9166990	B2	2015-10-20	Eswaran et al.	Entire Document
	110	9171079	B2	2015-10-27	Banka et al.	Entire Document
	111	9185124	B2	2015-11-10	Chakraborty	Entire Document
	112	9202040	B2	2015-12-01	Rosenblatt et al.	Entire Document
	113	9203827	B2	2015-12-01	Srinivasan et al.	Entire Document
	114	9210185	B1	2015-12-08	Pinney Wood et al.	Entire Document
	115	9231962	B1	2016-01-05	Yen et al.	Entire Document
	116	9235732	B2	2016-01-12	Eynon et al.	Entire Document
	117	9253643	B2	2016-02-02	Pattar et al.	Entire Document
	118	9256735	B2	2016-02-09	Stute	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	119	9262787	B2	2016-02-16	Binion et al.	Entire Document
	120	9264395	B1	2016-02-16	Stamos	Entire Document
	121	9276951	B2	2016-03-01	Choi et al.	Entire Document
	122	9286103	B2	2016-03-15	Acharya et al.	Entire Document
	123	9292692	B2	2016-03-22	Wallrabenstein	Entire Document
	124	9292699	B1	2016-03-22	Stuntebeck et al.	Entire Document
	125	9294498	B1	2016-03-22	Yampolskiy et al.	Entire Document
	126	9300682	B2	2016-03-29	Burnham et al.	Entire Document
	127	9319430	B2	2016-04-19	Bell, Jr. et al.	Entire Document
	128	9336481	B1	2016-05-10	Ionson	Entire Document
	129	9338061	B2	2016-05-10	Chen et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	130	9344444	B2	2016-05-17	Lippmann et al.	Entire Document
	131	9349103	B2	2016-05-24	Eberhardt, III et al.	Entire Document
	132	9369482	B2	2016-06-14	Borohovski et al.	Entire Document
	133	9384345	B2	2016-07-05	Dixon et al.	Entire Document
	134	9390376	B2	2016-07-12	Harrison et al.	Entire Document
	135	9400962	B2	2016-07-26	Prasad	Entire Document
	136	9438616	B2	2016-09-06	Singla et al.	Entire Document
	137	9461876	B2	2016-10-04	Van Dusen et al.	Entire Document
	138	9466041	B2	2016-10-11	Simitsis et al.	Entire Document
	139	9467461	B2	2016-10-11	Balderas	Entire Document
	140	9495188	B1	2016-11-15	Ettema et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	141	9501647	B2	2016-11-22	Yampolskiy et al.	Entire Document
	142	9503467	B2	2016-11-22	Lefebvre et al.	Entire Document
	143	9503472	B2	2016-11-22	Laidlaw et al.	Entire Document
	144	9509716	B2	2016-11-29	Shabtai et al.	Entire Document
	145	9515826	B2	2016-12-06	Whelan et al.	Entire Document
	146	9516053	B1	2016-12-06	Muddu et al.	Entire Document
	147	9521166	B2	2016-12-13	Wilson	Entire Document
	148	9541982	B2	2017-01-10	Lipasti et al.	Entire Document
	149	9558220	B2	2017-01-31	Nixon et al.	Entire Document
	150	9560065	B2	2017-01-31	Neil et al.	Entire Document
	151	9565204	B2	2017-02-07	Chesla	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	152	9571517	B2	2017-02-14	Vallone et al.	Entire Document
	153	9578046	B2	2017-02-21	Baker	Entire Document
	154	9596141	B2	2017-03-14	McDowall	Entire Document
	155	9600792	B2	2017-03-21	Foehr et al.	Entire Document
	156	9602513	B2	2017-03-21	Gamage et al.	Entire Document
	157	9602529	B2	2017-03-21	Jones et al.	Entire Document
	158	9602530	B2	2017-03-21	Ellis et al.	Entire Document
	159	9609009	B2	2017-03-28	Muddu et al.	Entire Document
	160	9609015	B2	2017-03-28	Natarajan et al.	Entire Document
	161	9619291	B2	2017-04-11	Pueyo et al.	Entire Document
	162	9639575	B2	2017-05-02	Leida et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	163	9652538	B2	2017-05-16	Shivaswamy et al.	Entire Document
	164	9652604	B1	2017-05-16	Johansson et al.	Entire Document
	165	9654495	B2	2017-05-16	Hubbard et al.	Entire Document
	166	9661019	B2	2017-05-23	Liu	Entire Document
	167	9667600	B2	2017-05-30	Piqueras Jover et al.	Entire Document
	168	9667641	B2	2017-05-30	Muddu et al.	Entire Document
	169	9672283	B2	2017-06-06	Pappas et al.	Entire Document
	170	9672355	B2	2017-06-06	Titonis et al.	Entire Document
	171	9674211	B2	2017-06-06	Curcic et al.	Entire Document
	172	9674249	B1	2017-06-06	Kekre et al.	Entire Document
	173	9679125	B2	2017-06-13	Bailor et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	174	9680867	B2	2017-06-13	Hughes et al.	Entire Document
	175	9686293	B2	2017-06-20	Golshan et al.	Entire Document
	176	9690645	B2	2017-06-27	Samuni et al.	Entire Document
	177	9699205	B2	2017-07-04	Muddu et al.	Entire Document
	178	9712553	B2	2017-07-18	Nguyen et al.	Entire Document
	179	9721086	B2	2017-08-01	Shear et al.	Entire Document
	180	9729421	B2	2017-08-08	Brech et al.	Entire Document
	181	9729538	B2	2017-08-08	Plotnik et al.	Entire Document
	182	9734169	B2	2017-08-15	Redlich et al.	Entire Document
	183	9734220	B2	2017-08-15	Karpi{hacek over (s)}t{hacek over (s)}enko et al.	Entire Document
	184	9736173	B2	2017-08-15	Li et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	185	9749343	B2	2017-08-29	Watters et al.	Entire Document
	186	9749344	B2	2017-08-29	Watters et al.	Entire Document
	187	9753796	B2	2017-09-05	Mahaffey et al.	Entire Document
	188	9756067	B2	2017-09-05	Boyadjiev et al.	Entire Document
	189	9762443	B2	2017-09-12	Dickey	Entire Document
	190	9771225	B2	2017-09-26	Stone et al.	Entire Document
	191	9772934	B2	2017-09-26	Maag et al.	Entire Document
	192	9774407	B2	2017-09-26	Hudson et al.	Entire Document
	193	9774522	B2	2017-09-26	Vasseur et al.	Entire Document
	194	9774616	B2	2017-09-26	Flores et al.	Entire Document
	195	9781144	B1	2017-10-03	Otvagin et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	196	9807104	B1	2017-10-31	Sarra	Entire Document
	197	9832213	B2	2017-11-28	Underwood et al.	Entire Document
	198	9842000	B2	2017-12-12	Bishop et al.	Entire Document
	199	9858322	B2	2018-01-02	Theimer et al.	Entire Document
	200	9860208	B1	2018-01-02	Ettema et al.	Entire Document
	201	9875360	B1	2018-01-23	Grossman et al.	Entire Document
	202	9882929	B1	2018-01-30	Ettema et al.	Entire Document
	203	9887933	B2	2018-02-06	Lawrence, III	Entire Document
	204	9910993	B2	2018-03-06	Grossman et al.	Entire Document
	205	9911088	B2	2018-03-06	Nath et al.	Entire Document
	206	9917860	B2	2018-03-13	Senanayake et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	207	9928366	B2	2018-03-27	Ladnai et al.	Entire Document
	208	9930058	B2	2018-03-27	Carpenter et al.	Entire Document
	209	9942295	B2	2018-04-10	Rider et al.	Entire Document
	210	9946517	B2	2018-04-17	Talby et al.	Entire Document
	211	9952899	B2	2018-04-24	Novaes	Entire Document
	212	9954879	B1	2018-04-24	Sadaghiani et al.	Entire Document
	213	9954884	B2	2018-04-24	Hassell et al.	Entire Document
	214	9965627	B2	2018-05-08	Ray et al.	Entire Document
	215	9967264	B2	2018-05-08	Harris et al.	Entire Document
	216	9967265	B1	2018-05-08	Peer et al.	Entire Document
	217	9967282	B2	2018-05-08	Thomas et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	218	9967283	B2	2018-05-08	Ray et al.	Entire Document
	219	9967625	B2	2018-05-08	Korst et al.	Entire Document
	220	9984129	B2	2018-05-29	Patel et al.	Entire Document
	221	9992228	B2	2018-06-05	Ray et al.	Entire Document
	222	10009378	B2	2018-06-26	Chiviendacz et al.	Entire Document
	223	10027711	B2	2018-07-17	Gill et al.	Entire Document
	224	10038559	B2	2018-07-31	Burrows et al.	Entire Document
	225	10044675	B1	2018-08-07	Ettema et al.	Entire Document
	226	10050985	B2	2018-08-14	Mhatre et al.	Entire Document
	227	10055473	B2	2018-08-21	Allen et al.	Entire Document
	228	10061635	B2	2018-08-28	Ellwein	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	229	10074052		2018-09-11	Banerjee et al.	Entire Document
	230	10078664	B2	2018-09-18	Gustafson et al.	Entire Document
	231	10083236	B2	2018-09-25	Crosby	Entire Document
	232	10102480	B2	2018-10-16	Dirac et al.	Entire Document
	233	10108907	B2	2018-10-23	Bugay et al.	Entire Document
	234	10109014	B1	2018-10-23	Bischoff et al.	Entire Document
	235	10110415	B2	2018-10-23	Radivojevic et al.	Entire Document
	236	10120907	B2	2018-11-06	de Castro Alves et al.	Entire Document
	237	10122687	B2	2018-11-06	Thomas et al.	Entire Document
	238	10122764	B1	2018-11-06	Obaidi	Entire Document
	239	10146592	B2	2018-12-04	Bishop et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	240	10152676	B1	2018-12-11	Strom	Entire Document
	241	10154049	B2	2018-12-11	Sancheti et al.	Entire Document
	242	10162969	B2	2018-12-25	Knapp	Entire Document
	243	10168691	B2	2019-01-01	Zornio et al.	Entire Document
	244	10180780	B2	2019-01-15	Ainalem	Entire Document
	245	10185832	B2	2019-01-22	Cam	Entire Document
	246	10191768	B2	2019-01-29	Bishop et al.	Entire Document
	247	10205735	B2	2019-02-12	Apostolopoulos	Entire Document
	248	10210246	B2	2019-02-19	Stojanovic et al.	Entire Document
	249	10210470	B2	2019-02-19	Partha Ray	Entire Document
	250	10212176	B2	2019-02-19	Wang	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	251	10212184	B2	2019-02-19	Sweeney et al.	Entire Document
	252	10216485	B2	2019-02-26	Misra et al.	Entire Document
	253	10217348	B2	2019-02-26	Poder et al.	Entire Document
	254	10261763	B2	2019-04-16	Fink et al.	Entire Document
	255	10275545	B2	2019-04-30	Yeager et al.	Entire Document
	256	10277629	B1	2019-04-30	Guntur	Entire Document
	257	10284570	B2	2019-05-07	Schmidtler et al.	Entire Document
	258	10289841	B2	2019-05-14	Tang et al.	Entire Document
	259	10290141	B2	2019-05-14	Kennedy et al.	Entire Document
	260	10298607	B2	2019-05-21	Tang et al.	Entire Document
	261	10305902	B2	2019-05-28	Kim	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	262	10318739	B2	2019-06-11	Brucker et al.	Entire Document
	263	10318882	B2	2019-06-11	Brueckner et al.	Entire Document
	264	10320828	B1	2019-06-11	Derbeko et al.	Entire Document
	265	10321278	B2	2019-06-11	Proctor	Entire Document
	266	10324773	B2	2019-06-18	Wing et al.	Entire Document
	267	10338913	B2	2019-07-02	Franchitti	Entire Document
	268	10367829	B2	2019-07-30	Huang et al.	Entire Document
	269	10380140	B2	2019-08-13	Sherman	Entire Document
	270	10387124	B2	2019-08-20	Chaudhuri et al.	Entire Document
	271	10387631	B2	2019-08-20	Duggal et al.	Entire Document
	272	10410113	B2	2019-09-10	Clayton et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	273	10410214	B2	2019-09-10	Doyle	Entire Document
	274	10438001	B1	2019-10-08	Hariprasad	Entire Document
	275	10440054	B2	2019-10-08	Robertson	Entire Document
	276	10445482	B2	2019-10-15	Ren	Entire Document
	277	10452664	B2	2019-10-22	Le Mouel et al.	Entire Document
	278	10462112	B1	2019-10-29	Makmel et al.	Entire Document
	279	10505954	B2	2019-12-10	Stokes, III et al.	Entire Document
	280	10511498	B1	2019-12-17	Narayan et al.	Entire Document
	281	10515062	B2	2019-12-24	Tidwell et al.	Entire Document
	282	10515366	B1	2019-12-24	Gorelik et al.	Entire Document
	283	10530796	B2	2020-01-07	Patterson et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	284	10540624	B2	2020-01-21	Hui et al.	Entire Document
	285	10579691	B2	2020-03-03	Levine et al.	Entire Document
	286	10601854	B2	2020-03-24	Lokamathe et al.	Entire Document
	287	10606454	B2	2020-03-31	Pani	Entire Document
	288	10609059	B2	2020-03-31	Apostolopoulos	Entire Document
	289	10628578	B2	2020-04-21	Eksten et al.	Entire Document
	290	10643144	B2	2020-05-05	Bowers et al.	Entire Document
	291	10645086	B1	2020-05-05	Hadler	Entire Document
	292	10645100	B1	2020-05-05	Wang et al.	Entire Document
	293	10673880	B1	2020-06-02	Pratt et al.	Entire Document
	294	10715534	B2	2020-07-14	Sander et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	295	10740358	B2	2020-08-11	Chan et al.	Entire Document
	296	10764321	B2	2020-09-01	Bower, III et al.	Entire Document
	297	10776847	B1	2020-09-15	Comar et al.	Entire Document
	298	10789367	B2	2020-09-29	Joseph Durairaj et al.	Entire Document
	299	10791131	B2	2020-09-29	Nor et al.	Entire Document
	300	10817530	B2	2020-10-27	Siebel et al.	Entire Document

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS Remove

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1					

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2] ¡	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	1	2930026	CA	A1	2015-05-14	Theimer et al.	Entire Document	☐
	2	2014159150	WO	A1	2014-10-02	Jubran et al.	Entire Document	☐
	3	2015089463	WO	A1	2015-06-18	Gupta et al.	Entire Document	☐
	4	2020079685	WO	A1	2020-04-23	Atsmon et al.	Entire Document	☐

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]
	1	BOUKHTOUTA, et al, "Graph-theoretic characterization of cyber-threat infrastructures", Digital Investigation, 2015, vol 14, pg. S3-S15, USA	☐
	2	CUI, et al, "Non-intrusive process-based monitoring system to mitigate and prevent VM vulnerability explorations", Collaboratecom, 2013, Austin, USA	☐
	3	EKELHART, et al, "Integrating attacker behavior in IT security analysis: a discrete-event simulation approach", ResearchGate, 2015	☐
	4	FISK, VARGHESE, "Agile and Scalable Analysis of Network Events"	☐
	5	GEDIK et al, "Elastic Scaling for Data Stream Processing", IEEE Transactions on Parallel and Distributed Systems, June 2014, vol. 25, no. 6, pg. 1447-1463	☐

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	6	JAJODIA, et al, "Advanced Cyber Attack Modeling, Analysis, and Visualization", AFRL Final Technical Report, March 2010, USA	☐
	7	JPCERT-CC, "Detecting Lateral Movement through Tracking Event Logs (Version 2)", 2017, pg. 1-16, Japan	☐
	8	JUNGLES et al, "Mitigating Pass-the-Hash (PtH) Attacks and Other Credential Theft Techniques", TwC Next, 2012	☐
	9	KBAR, "Wireless Network Token-Based Fast Authentication", 17th International Conference on Telecommunications, 2010, pg. 227-233	☐
	10	KIESLING et al, "Selecting security control portfolios: a multi-objective simulation-optimization approach", EURO Journal on Decision Processes, April 2016	☐
	11	KOTENKO, "A Cyber Attack Modeling and Impact Assessment framework", Conference Paper, January 2013	☐
	12	KUMAR, et al, "DFuse: A Framework for Distributed Data Fusion", Georgia Institute of Technology	☐
	13	LU, et al, "Sybil Attack Detection through Global Topology Pattern Visualization", 2011	☐
	14	PASQUALETTI et al, "Attack Detection and Identification in Cyber-Physical Systems", IEEE Transactions on Automatic Control, vol. 58, no. 11, pg. 2715-2729	☐
	15	Patapanchala, "Exploring Security Metrics for Electric Grid Infrastructure Leveraging Attack Graphs", Oregon State Thesis, 2016	☐
	16	SHANDILYA et al, "Use of Attack Graphs in Security Systems", Journal of Computer Networks and Communications, vol. 2014	☐

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	17	YANG et al, "Attack Projection", Advances in Information Security 62, 2014, pg. 239-261	☐
	18	ZARGAR et al, "XABA: A Zero-Knowledge Anomaly-Based Behavioral Analysis Method to Detect Insider Threats", 2016	☐

If you wish to add additional non-patent literature document citation information please click the Add button Add

EXAMINER SIGNATURE

Examiner Signature	/Catherine Thiaw/	Date Considered	09/26/2024

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT

(**Not for submission under 37 CFR 1.99)**

Application Number	
Filing Date	2024-07-21
First Named Inventor	Jason Crabtree
Art Unit	
Examiner Name	
Attorney Docket Number	15305FTNK

CERTIFICATION STATEMENT

Please see 37 CFR 1.97(e) to make the appropriate selection(s):

☐ That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

☐ See attached certification statement.

☒ A certification statement is not submitted herewith.

FEE

☐ The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

COPIES

☐ An identification of an earlier application pursuant to 37 CFR 1.98 (d) (1) is attached.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/BRIAN R. GALVIN/	Date (YYYY-MM-DD)	2024-07-19
Name/Print	Brian R. Galvin	Registration Number	50966

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these record s.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

Bibliographic Data

Application No: 18/779,029

Foreign Priority claimed: ○ Yes ◉ No

35 USC 119 (a-d) conditions met: ☐ Yes ☐ No ☐ Met After Allowance

Verified and Acknowledged: /Catherine Thiaw/ (Examiner's Signature) (Initials)

Title: RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH

FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
07/21/2024	718	2493	15305FT
RULE			

APPLICANTS

QOMPLX LLC, Reston, VA, UNITED STATES

INVENTORS

Jason Crabtree, Vienna, VA, UNITED STATES

Andrew Sellers, Monument, CO, UNITED STATES

CONTINUING DATA

This application is a CON of 18581375 02/20/2024

18581375 is a CON of 17189161 03/01/2021

17189161 is a CIP of 17061195 10/01/2020 PAT 11570214

17061195 is a CIP of 17035029 09/28/2020 PAT 11546380

17035029 is a CIP of 17008276 08/31/2020 PAT 11323484

17008276 is a CIP of 17000504 08/24/2020 PAT 11477245

17000504 is a CIP of 16855724 04/22/2020 PAT 11218510

16855724 is a CIP of 16836717 03/31/2020 PAT 10917428

16855724 is a CIP of 16777270 01/30/2020 PAT 11025674

16777270 is a CIP of 16720383 12/19/2019 PAT 10944795

17189161 is a CIP of 16709598 12/10/2019 PAT 11507858

17000504 is a CIP of 16412340 05/14/2019 PAT 11539663

16412340 is a CIP of 16267893 02/05/2019ABN

16267893 is a CIP of 16248133 01/15/2019ABN

16836717 is a CIP of 15887496 02/02/2018 PAT 10783241

17061195 is a CIP of 15879801 01/25/2018ABN

16248133 is a CIP of 15849901 12/21/2017 PAT 11023284

15849901 is a CIP of 15835436 12/07/2017 PAT 10572828

15849901 is a CIP of 15835312 12/07/2017 PAT 11055451

16720383 is a CON of 15823363 11/27/2017 PAT 10560483

15887496 is a CIP of 15823285 11/27/2017 PAT 10740096

15887496 is a CIP of 15818733 11/20/2017 PAT 10673887

16248133 is a CIP of 15813097 11/14/2017ABN

16248133 is a CIP of 15806697 11/08/2017ABN

15835436 is a CIP of 15790457 10/23/2017 PAT 10884999

15790457 is a CIP of 15790327 10/23/2017 PAT 10860951

15823285 is a CIP of 15788718 10/19/2017 PAT 10861014

15788718 is a CIP of 15788002 10/19/2017ABN

15788002 is a CIP of 15787601 10/18/2017 PAT 10860660

15790457 has PRO of 62568298 10/04/2017

15787601 has PRO of 62568312 10/04/2017

15788002 has PRO of 62568305 10/04/2017

15818733 is a CIP of 15725274 10/04/2017 PAT 10609079

15823363 is a CIP of 15725274 10/04/2017 PAT 10609079

15790327 has PRO of 62568291 10/04/2017

15788718 has PRO of 62568307 10/04/2017

16248133 is a CIP of 15673368 08/09/2017ABN

15725274 is a CIP of 15655113 07/20/2017 PAT 10735456

15813097 is a CIP of 15616427 06/07/2017ABN

15655113 is a CIP of 15616427 06/07/2017ABN

15790327 is a CIP of 15616427 06/07/2017ABN

15787601 is a CIP of 15616427 06/07/2017ABN

15879801 is a CIP of 15379899 12/15/2016ABN

15673368 is a CIP of 15376657 12/13/2016 PAT 10402906

15806697 is a CIP of 15376657 12/13/2016 PAT 10402906

15379899 is a CIP of 15376657 12/13/2016 PAT 10402906

15806697 is a CIP of 15343209 11/04/2016 PAT 11087403

15376657 is a CIP of 15237625 08/15/2016 PAT 10248910

15655113 is a CIP of 15237625 08/15/2016 PAT 10248910

15343209 is a CIP of 15237625 08/15/2016 PAT 10248910

15343209 is a CIP of 15229476 08/05/2016 PAT 10454791

15237625 is a CIP of 15206195 07/08/2016ABN

15229476 is a CIP of 15206195 07/08/2016ABN

15206195 is a CIP of 15186453 06/18/2016ABN

15835312 is a CIP of 15186453 06/18/2016ABN

15186453 is a CIP of 15166158 05/26/2016ABN

15790327 is a CIP of 15141752 04/28/2016 PAT 10860962

15166158 is a CIP of 15141752 04/28/2016 PAT 10860962

15141752 is a CIP of 15091563 04/05/2016 PAT 10204147

15141752 is a CIP of 14986536 12/31/2015 PAT 10210255

16709598 is a CIP of 14925974 10/28/2015ABN

15616427 is a CIP of 14925974 10/28/2015ABN

15141752 is a CIP of 14925974 10/28/2015ABN

FOREIGN APPLICATIONS

IF REQUIRED, FOREIGN LICENSE GRANTED**

08/20/2024

STATE OR COUNTRY

UNITED STATES

ADDRESS

Brian Galvin
Galvin Patent Law LLC
2916 NW Bucklin Hill Road
Suite 485
Silverdale, WA 98383
UNITED STATES

FILING FEE RECEIVED

$6,160

Doc code: IDS
Doc description: Information Disclosure Statement (IDS) Filed

PTO/SB/08a (01-22)
Approved for use through 05/31/2024. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it contains a valid OMB control number.

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (**Not for submission under 37 CFR 1.99)**		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

U.S.PATENTS						Remove
Examiner Initial*	Cite No	Patent Number	Kind Code[1]	Issue Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	10861028	B2	2020-12-08	Silberman et al.	Entire Document
	2	10862916	B2	2020-12-08	Hittel et al.	Entire Document
	3	10871951	B2	2020-12-22	Ding et al.	Entire Document
	4	10911470	B2	2021-02-02	Muddu et al.	Entire Document
	5	10944772	B2	2021-03-09	Mulchandani et al.	Entire Document
	6	10958667	B1	2021-03-23	Maida et al.	Entire Document
	7	10965711	B2	2021-03-30	Schiappa et al.	Entire Document
	8	10977551	B2	2021-04-13	Van Seijen et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	9	10985997	B2	2021-04-20	Duggal et al.	Entire Document
	10	10992698	B2	2021-04-27	Patel et al.	Entire Document
	11	11030520	B2	2021-06-08	Mankovskii et al.	Entire Document
	12	11032307	B2	2021-06-08	Tsironis	Entire Document
	13	11113667	B1	2021-09-07	Jiang et al.	Entire Document
	14	11138514	B2	2021-10-05	Hu et al.	Entire Document
	15	11194900	B2	2021-12-07	Loman et al.	Entire Document
	16	11256791	B2	2022-02-22	Douglas et al.	Entire Document
	17	11334831	B2	2022-05-17	Abu El Ata et al.	Entire Document
	18	11392875	B2	2022-07-19	Carstens et al.	Entire Document
	19	11477641	B2	2022-10-18	Damlaj et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	20	11539663	B2	2022-12-27	Chasman et al.	Entire Document
	21	11574206	B2	2023-02-07	Butler, Jr. et al.	Entire Document
	22	11736299	B2	2023-08-22	Cerna, Jr.	Entire Document

If you wish to add additional U.S. Patent citation information please click the Add button. Add

U.S.PATENT APPLICATION PUBLICATIONS Remove

Examiner Initial*	Cite No	Publication Number	Kind Code[1]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear
	1	20040255167	A1	2004-12-16	Knight, James Michael	Entire Document
	2	20050071223	A1	2005-03-31	Jain, Vivek et al.	Entire Document
	3	20050165822	A1	2005-07-28	Yeung, Simon D. et al.	Entire Document
	4	20050198099	A1	2005-09-08	Motsinger, David Lee et al.	Entire Document
	5	20050289072	A1	2005-12-29	Sabharwal, Vinay	Entire Document
	6	20060149575	A1	2006-07-06	Varadarajan et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	7	20070055558	A1	2007-03-08	Shanahan et al.	Entire Document
	8	20070136821	A1	2007-06-14	Hershaft et al.	Entire Document
	9	20070150744	A1	2007-06-28	Cheng et al.	Entire Document
	10	20070168370	A1	2007-07-19	Hardy	Entire Document
	11	20070276714	A1	2007-11-29	Beringer	Entire Document
	12	20080021866	A1	2008-01-24	HINTON et al.	Entire Document
	13	20080270203	A1	2008-10-30	Holmes et al.	Entire Document
	14	20090012760	A1	2009-01-08	SCHUNEMANN	Entire Document
	15	20090094372	A1	2009-04-09	Nyang et al.	Entire Document
	16	20090199002	A1	2009-08-06	Erickson	Entire Document
	17	20090319247	A1	2009-12-24	Ratcliffe, III et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	18	20100083240	A1	2010-04-01	Siman	Entire Document
	19	20100115276	A1	2010-05-06	Betouin et al.	Entire Document
	20	20100275183	A1	2010-10-28	Panicker et al.	Entire Document
	21	20100299651	A1	2010-11-25	FAINEKOS et al.	Entire Document
	22	20100325685	A1	2010-12-23	Sanbower	Entire Document
	23	20110087888	A1	2011-04-14	Rennie	Entire Document
	24	20110225287	A1	2011-09-15	DALAL et al.	Entire Document
	25	20110307467	A1	2011-12-15	Severance	Entire Document
	26	20120215575	A1	2012-08-23	Deb et al.	Entire Document
	27	20120296845	A1	2012-11-22	Andrews et al.	Entire Document
	28	20130046751	A1	2013-02-21	Tsiatsis et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	29	20130117831	A1	2013-05-09	Hook et al.	Entire Document
	30	20130132149	A1	2013-05-23	Wei et al.	Entire Document
	31	20130159219	A1	2013-06-20	Pantel et al.	Entire Document
	32	20140082729	A1	2014-03-20	Shim et al.	Entire Document
	33	20140149186	A1	2014-05-29	Flaxer et al.	Entire Document
	34	20140244612	A1	2014-08-28	Bhasin et al.	Entire Document
	35	20140279762	A1	2014-09-18	Xaypanya et al.	Entire Document
	36	20140324521	A1	2014-10-30	Mun	Entire Document
	37	20150081363	A1	2015-03-19	Taylor et al.	Entire Document
	38	20150128258	A1	2015-05-07	Novozhenets	Entire Document
	39	20150149979	A1	2015-05-28	Talby et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	40	20150161738	A1	2015-06-11	Stempora	Entire Document
	41	20150170053	A1	2015-06-18	Miao	Entire Document
	42	20150172311	A1	2015-06-18	Freedman et al.	Entire Document
	43	20150242509	A1	2015-08-27	Pall et al.	Entire Document
	44	20150281225	A1	2015-10-01	Schoen et al.	Entire Document
	45	20150295775	A1	2015-10-15	Dickey	Entire Document
	46	20150317745	A1	2015-11-05	Collins et al.	Entire Document
	47	20150347414	A1	2015-12-03	Xiao et al.	Entire Document
	48	20150379111	A1	2015-12-31	Hwang	Entire Document
	49	20160004858	A1	2016-01-07	Chen et al.	Entire Document
	50	20160006629	A1	2016-01-07	Ianakiev et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	51	20160012235	A1	2016-01-14	Lee et al.	Entire Document
	52	20160057159	A1	2016-02-25	Yin et al.	Entire Document
	53	20160088000	A1	2016-03-24	Siva Kumar et al.	Entire Document
	54	20160099960	A1	2016-04-07	Gerritz et al.	Entire Document
	55	20160119365	A1	2016-04-28	Barel	Entire Document
	56	20160140519	A1	2016-05-19	Trepca et al.	Entire Document
	57	20160180240	A1	2016-06-23	MAJUMDAR et al.	Entire Document
	58	20160219066	A1	2016-07-28	Vasseur et al.	Entire Document
	59	20160275123	A1	2016-09-22	LIN et al.	Entire Document
	60	20160330233	A1	2016-11-10	Hart	Entire Document
	61	20160364307	A1	2016-12-15	Garg et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	62	20160371363	A1	2016-12-22	MURO et al.	Entire Document
	63	20170010589	A1	2017-01-12	de Anda Fast	Entire Document
	64	20170023509	A1	2017-01-26	KIM et al.	Entire Document
	65	20170090893	A1	2017-03-30	Aditya et al.	Entire Document
	66	20170207926	A1	2017-07-20	GIL et al.	Entire Document
	67	20170241791	A1	2017-08-24	Madigan et al.	Entire Document
	68	20180268264	A1	2018-09-20	Marwah et al.	Entire Document
	69	20180336250	A1	2018-11-22	LLAVES et al.	Entire Document
	70	20190188797	A1	2019-06-20	Przechocki et al.	Entire Document
	71	20200004752	A1	2020-01-02	MAJUMDAR et al.	Entire Document
	72	20200177618	A1	2020-06-04	Hassanzadeh et al.	Entire Document

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

	73	20200304534	A1	2020-09-24	RAKESH et al.	Entire Document
	74	20200356664	A1	2020-11-12	Maor	Entire Document
	75	20200364346	A1	2020-11-19	Gourisetti et al.	Entire Document
	76	20200396246	A1	2020-12-17	Zoldi et al.	Entire Document
	77	20210075822	A1	2021-03-11	CHUNG et al.	Entire Document

If you wish to add additional U.S. Published Application citation information please click the Add button. Add

FOREIGN PATENT DOCUMENTS Remove

Examiner Initial*	Cite No	Foreign Document Number[3]	Country Code[2]	Kind Code[4]	Publication Date	Name of Patentee or Applicant of cited Document	Pages,Columns,Lines where Relevant Passages or Relevant Figures Appear	T[5]
	1							☐

If you wish to add additional Foreign Patent Document citation information please click the Add button Add

NON-PATENT LITERATURE DOCUMENTS Remove

Examiner Initials*	Cite No	Include name of the author (in CAPITAL LETTERS), title of the article (when appropriate), title of the item (book, magazine, journal, serial, symposium, catalog, etc), date, pages(s), volume-issue number(s), publisher, city and/or country where published.	T[5]
	1		☐

If you wish to add additional non-patent literature document citation information please click the Add button Add

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

EXAMINER SIGNATURE			
Examiner Signature	/Catherine Thiaw/	Date Considered	09/26/2024

*EXAMINER: Initial if reference considered, whether or not citation is in conformance with MPEP 609. Draw line through a citation if not in conformance and not considered. Include copy of this form with next communication to applicant.

[1] See Kind Codes of USPTO Patent Documents at www.USPTO.GOV or MPEP 901.04. [2] Enter office that issued the document, by the two-letter code (WIPO Standard ST.3). [3] For Japanese patent documents, the indication of the year of the reign of the Emperor must precede the serial number of the patent document. [4] Kind of document by the appropriate symbols as indicated on the document under WIPO Standard ST.16 if possible. [5] Applicant is to place a check mark here if English language translation is attached.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	
	Filing Date	2024-07-21
	First Named Inventor	Jason Crabtree
	Art Unit	
	Examiner Name	
	Attorney Docket Number	15305FTNK

CERTIFICATION STATEMENT

Please see 37 CFR 1.97(e) to make the appropriate selection(s):

☐ That each item of information contained in the information disclosure statement was first cited in any communication from a foreign patent office in a counterpart foreign application not more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(1).

OR

☐ That no item of information contained in the information disclosure statement was cited in a communication from a foreign patent office in a counterpart foreign application, and, to the knowledge of the person signing the certification after making reasonable inquiry, no item of information contained in the information disclosure statement was known to any individual designated in 37 CFR 1.56(c) more than three months prior to the filing of the information disclosure statement. See 37 CFR 1.97(e)(2).

☐ See attached certification statement.

☒ A certification statement is not submitted herewith.

FEE

☐ The fee set forth in 37 CFR 1.17 (p) has been submitted herewith.

COPIES

☐ An identification of an earlier application pursuant to 37 CFR 1.98 (d) (1) is attached.

SIGNATURE

A signature of the applicant or representative is required in accordance with CFR 1.33, 10.18. Please see CFR 1.4(d) for the form of the signature.

Signature	/BRIAN R. GALVIN/	Date (YYYY-MM-DD)	2024-07-19
Name/Print	Brian R. Galvin	Registration Number	50966

This collection of information is required by 37 CFR 1.97 and 1.98. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 1 hour to complete, including gathering, preparing and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.**

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. Accordingly, pursuant to the requirements of the Act, please be advised that: (1) the general authority for the collection of this information is 35 U.S.C. 2(b)(2); (2) furnishing of the information solicited is voluntary; and (3) the principal purpose for which the information is used by the U.S. Patent and Trademark Office is to process and/or examine your submission related to a patent application or patent. If you do not furnish the requested information, the U.S. Patent and Trademark Office may not be able to process and/or examine your submission, which may result in termination of proceedings or abandonment of the application or expiration of the patent.

The information provided by you in this form will be subject to the following routine uses:

1. The information on this form will be treated confidentially to the extent allowed under the Freedom of Information Act (5 U.S.C. 552) and the Privacy Act (5 U.S.C. 552a). Records from this system of records may be disclosed to the Department of Justice to determine whether the Freedom of Information Act requires disclosure of these record s.

2. A record from this system of records may be disclosed, as a routine use, in the course of presenting evidence to a court, magistrate, or administrative tribunal, including disclosures to opposing counsel in the course of settlement negotiations.

3. A record in this system of records may be disclosed, as a routine use, to a Member of Congress submitting a request involving an individual, to whom the record pertains, when the individual has requested assistance from the Member with respect to the subject matter of the record.

4. A record in this system of records may be disclosed, as a routine use, to a contractor of the Agency having need for the information in order to perform a contract. Recipients of information shall be required to comply with the requirements of the Privacy Act of 1974, as amended, pursuant to 5 U.S.C. 552a(m).

5. A record related to an International Application filed under the Patent Cooperation Treaty in this system of records may be disclosed, as a routine use, to the International Bureau of the World Intellectual Property Organization, pursuant to the Patent Cooperation Treaty.

6. A record in this system of records may be disclosed, as a routine use, to another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c)).

7. A record from this system of records may be disclosed, as a routine use, to the Administrator, General Services, or his/her designee, during an inspection of records conducted by GSA as part of that agency's responsibility to recommend improvements in records management practices and programs, under authority of 44 U.S.C. 2904 and 2906. Such disclosure shall be made in accordance with the GSA regulations governing inspection of records for this purpose, and any other relevant (i.e., GSA or Commerce) directive. Such disclosure shall not be used to make determinations about individuals.

8. A record from this system of records may be disclosed, as a routine use, to the public after either publication of the application pursuant to 35 U.S.C. 122(b) or issuance of a patent pursuant to 35 U.S.C. 151. Further, a record may be disclosed, subject to the limitations of 37 CFR 1.14, as a routine use, to the public if the record was filed in an application which became abandoned or in which the proceedings were terminated and which application is referenced by either a published application, an application open to public inspections or an issued patent.

9. A record from this system of records may be disclosed, as a routine use, to a Federal, State, or local law enforcement agency, if the USPTO becomes aware of a violation or potential violation of law or regulation.

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.T/

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

NOTICE OF ALLOWANCE AND FEE(S) DUE

82318 7590 09/29/2024

Brian Galvin
Galvin Patent Law LLC
2916 NW Bucklin Hill Road
Suite 485
Silverdale, WA 98383

EXAMINER	
THIAW, CATHERINE B	
ART UNIT	**PAPER NUMBER**
2493	

DATE MAILED: 09/29/2024

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
18/779,029	07/21/2024	Jason Crabtree	15305FT	4119

TITLE OF INVENTION: RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	UNDISCOUNTED	$1200	$0.00	$0.00	$1200	12/30/2024

THE APPLICATION IDENTIFIED ABOVE HAS BEEN EXAMINED AND IS ALLOWED FOR ISSUANCE AS A PATENT. PROSECUTION ON THE MERITS IS CLOSED. THIS NOTICE OF ALLOWANCE IS NOT A GRANT OF PATENT RIGHTS. THIS APPLICATION IS SUBJECT TO WITHDRAWAL FROM ISSUE AT THE INITIATIVE OF THE OFFICE OR UPON PETITION BY THE APPLICANT. SEE 37 CFR 1.313 AND MPEP 1308.

THE ISSUE FEE AND PUBLICATION FEE (IF REQUIRED) MUST BE PAID WITHIN THREE MONTHS FROM THE MAILING DATE OF THIS NOTICE OR THIS APPLICATION SHALL BE REGARDED AS ABANDONED. THIS STATUTORY PERIOD CANNOT BE EXTENDED. SEE 35 U.S.C. 151. THE ISSUE FEE DUE INDICATED ABOVE DOES NOT REFLECT A CREDIT FOR ANY PREVIOUSLY PAID ISSUE FEE IN THIS APPLICATION. IF AN ISSUE FEE HAS PREVIOUSLY BEEN PAID IN THIS APPLICATION (AS SHOWN ABOVE), THE RETURN OF PART B OF THIS FORM WILL BE CONSIDERED A REQUEST TO REAPPLY THE PREVIOUSLY PAID ISSUE FEE TOWARD THE ISSUE FEE NOW DUE.

HOW TO REPLY TO THIS NOTICE:

I. Review the ENTITY STATUS shown above. If the ENTITY STATUS is shown as SMALL or MICRO, verify whether entitlement to that entity status still applies.

If the ENTITY STATUS is the same as shown above, pay the TOTAL FEE(S) DUE shown above.

If the ENTITY STATUS is changed from that shown above, on PART B - FEE(S) TRANSMITTAL, complete section number 5 titled "Change in Entity Status (from status indicated above)".

For purposes of this notice, small entity fees are 40% the amount of undiscounted fees, and micro entity fees are 20% the amount of undiscounted fees.

II. PART B - FEE(S) TRANSMITTAL, or its equivalent, must be completed and returned to the United States Patent and Trademark Office (USPTO) with your ISSUE FEE and PUBLICATION FEE (if required). If you are charging the fee(s) to your deposit account, section "4b" of Part B - Fee(s) Transmittal should be completed. If an equivalent of Part B is filed, a request to reapply a previously paid issue fee must be clearly made, and delays in processing may occur due to the difficulty in recognizing the paper as an equivalent of Part B.

III. All communications regarding this application must give the application number. Please direct all communications prior to issuance to Mail Stop ISSUE FEE unless advised to the contrary.

IMPORTANT REMINDER: Maintenance fees are due in utility patents issuing on applications filed on or after Dec. 12, 1980. It is patentee's responsibility to ensure timely payment of maintenance fees when due. More information is available at www.uspto.gov/PatentMaintenanceFees.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), by mail or fax, or via the USPTO patent electronic filing system.

By mail, send to: Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

By fax, send to: (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications. **Because electronic patent issuance may occur shortly after issue fee payment, any desired continuing application should preferably be filed prior to payment of this issue fee in order not to jeopardize copendency.**

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

82318 7590 09/29/2024

Brian Galvin
Galvin Patent Law LLC
2916 NW Bucklin Hill Road
Suite 485
Silverdale, WA 98383

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being transmitted to the USPTO via the USPTO patent electronic filing system or by facsimile to (571) 273-2885, on the date below.

(Typed or printed name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
18/779,029	07/21/2024	Jason Crabtree	15305FT	4119

TITLE OF INVENTION: RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	UNDISCOUNTED	$1200	$0.00	$0.00	$1200	12/30/2024

EXAMINER	ART UNIT	CLASS-SUBCLASS
THIAW, CATHERINE B	2493	718-102000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

❑ Change of correspondence address (or Change of Correspondence Address form PTO/AIA/122 or PTO/SB/122) attached.

❑ "Fee Address" indication (or "Fee Address" Indication form PTO/AIA/47 or PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
(1) The names of up to 3 registered patent attorneys or agents OR, alternatively,
(2) The name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 _____
2 _____
3 _____

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document must have been previously recorded, or filed for recordation, as set forth in 37 CFR 3.11 and 37 CFR 3.81(a). Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE (B) RESIDENCE: (CITY and STATE OR COUNTRY)

Please check the appropriate assignee category or categories (will not be printed on the patent) : ❑ Individual ❑ Corporation or other private group entity ❑ Government

4a. Fees submitted: ❑ Issue Fee ❑ Publication Fee (if required)

4b. Method of Payment: *(Please first reapply any previously paid fee shown above)*

❑ Electronic Payment via the USPTO patent electronic filing system ❑ Enclosed check ❑ Non-electronic payment by credit card (Attach form PTO-2038)

❑ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment to Deposit Account No. _____

5. **Change in Entity Status** (from status indicated above)

❑ Applicant certifying micro entity status. See 37 CFR 1.29

NOTE: Absent a valid certification of Micro Entity Status (see forms PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.

❑ Applicant asserting small entity status. See 37 CFR 1.27

NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.

❑ Applicant changing to regular undiscounted fee status.

NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: This form must be signed in accordance with 37 CFR 1.31 and 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Authorized Signature _____ Date _____

Typed or printed name _____ Registration No. _____

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
18/779,029	07/21/2024	Jason Crabtree	15305FT	4119

82318 7590 09/29/2024

Brian Galvin
Galvin Patent Law LLC
2916 NW Bucklin Hill Road
Suite 485
Silverdale, WA 98383

EXAMINER	
THIAW, CATHERINE B	

ART UNIT	PAPER NUMBER
2493	

DATE MAILED: 09/29/2024

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(Applications filed on or after May 29, 2000)

The Office has discontinued providing a Patent Term Adjustment (PTA) calculation with the Notice of Allowance.

Section 1(h)(2) of the AIA Technical Corrections Act amended 35 U.S.C. 154(b)(3)(B)(i) to eliminate the requirement that the Office provide a patent term adjustment determination with the notice of allowance. See Revisions to Patent Term Adjustment, 78 Fed. Reg. 19416, 19417 (Apr. 1, 2013). Therefore, the Office is no longer providing an initial patent term adjustment determination with the notice of allowance. The Office will continue to provide a patent term adjustment determination with the Issue Notification Letter that is mailed to applicant approximately three weeks prior to the issue date of the patent, and will include the patent term adjustment on the patent. Any request for reconsideration of the patent term adjustment determination (or reinstatement of patent term adjustment) should follow the process outlined in 37 CFR 1.705.

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Customer Service Center of the Office of Patent Publication at 1-(888)-786-0101 or (571)-272-4200.

OMB Clearance and PRA Burden Statement for PTOL-85 Part B

The Paperwork Reduction Act (PRA) of 1995 requires Federal agencies to obtain Office of Management and Budget approval before requesting most types of information from the public. When OMB approves an agency request to collect information from the public, OMB (i) provides a valid OMB Control Number and expiration date for the agency to display on the instrument that will be used to collect the information and (ii) requires the agency to inform the public about the OMB Control Number's legal significance in accordance with 5 CFR 1320.5(b).

The information collected by PTOL-85 Part B is required by 37 CFR 1.311. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 30 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, Virginia 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, Virginia 22313-1450. Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Privacy Act Statement

The Privacy Act of 1974 (P.L. 93-579) requires that you be given certain information in connection with your submission of the attached form related to a patent application or patent. The United States Patent and Trademark Office (USPTO) collects the information in this record under authority of 35 U.S.C. 2. The USPTO's system of records is used to manage all applicant and owner information including name, citizenship, residence, post office address, and other information with respect to inventors and their legal representatives pertaining to the applicant's/ owner's activities in connection with the invention for which a patent is sought or has been granted. The applicable Privacy Act System of Records Notice for the information collected in this form is COMMERCE/PAT-TM-7 Patent Application Files, available in the Federal Register at 78 FR 19243 (March 29, 2013).
https://www.govinfo.gov/content/pkg/FR-2013-03-29/pdf/2013-07341.pdf

Routine uses of the information in this record may include disclosure to:

1) law enforcement, in the event that the system of records indicates a violation or potential violation of law;
2) a federal, state, local, or international agency, in response to its request;
3) a contractor of the USPTO having need for the information in order to perform a contract;
4) the Department of Justice for determination of whether the Freedom of Information Act (FOIA) requires disclosure of the record;
5) a Member of Congress submitting a request involving an individual to whom the record pertains, when the individual has requested the Member's assistance with respect to the subject matter of the record;
6) a court, magistrate, or administrative tribunal, in the course of presenting evidence, including disclosures to opposing counsel in the course of settlement negotiations;
7) the Administrator, General Services Administration (GSA), or their designee, during an inspection of records conducted by GSA under authority of 44 U.S.C. 2904 and 2906, in accordance with the GSA regulations and any other relevant (i.e., GSA or Commerce) directive, where such disclosure shall not be used to make determinations about individuals;
8) another federal agency for purposes of National Security review (35 U.S.C. 181) and for review pursuant to the Atomic Energy Act (42 U.S.C. 218(c));
9) the Office of Personnel Management (OPM) for personnel research purposes; and
10) the Office of Management and Budget (OMB) for legislative coordination and clearance.

If you do not furnish the information requested on this form, the USPTO may not be able to process and/or examine your submission, which may result in termination of proceedings, abandonment of the application, and/or expiration of the patent.

Notice of Allowability	Application No. 18/779,029	Applicant(s) Crabtree et al.	
	Examiner CATHERINE B THIAW	Art Unit 2493	AIA (FITF) Status Yes

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address--

All claims being allowable, PROSECUTION ON THE MERITS IS (OR REMAINS) CLOSED in this application. If not included herewith (or previously mailed), a Notice of Allowance (PTOL-85) or other appropriate communication will be mailed in due course. **THIS NOTICE OF ALLOWABILITY IS NOT A GRANT OF PATENT RIGHTS.** This application is subject to withdrawal from issue at the initiative of the Office or upon petition by the applicant. See 37 CFR 1.313 and MPEP 1308.

1. ☑ This communication is responsive to communication received 7/21/2024 and interview with Attorney.
   ☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2. ☐ An election was made by the applicant in response to a restriction requirement set forth during the interview on _____; the restriction requirement and election have been incorporated into this action.

3. ☑ The allowed claim(s) is/are 1-8 and 11-15 . As a result of the allowed claim(s), you may be eligible to benefit from the **Patent Prosecution Highway** program at a participating intellectual property office for the corresponding application. For more information , please see **http://www.uspto.gov/patents/init_events/pph/index.jsp** or send an inquiry to **PPHfeedback@uspto.gov.**

4. ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

   Certified copies:

   a) ☐ All b) ☐ Some* c) ☐ None of the:

   1. ☐ Certified copies of the priority documents have been received.
   2. ☐ Certified copies of the priority documents have been received in Application No. _____ .
   3. ☐ Copies of the certified copies of the priority documents have been received in this national stage application from the International Bureau (PCT Rule 17.2(a)).

   * Certified copies not received: _____ .

Applicant has THREE MONTHS FROM THE "MAILING DATE" of this communication to file a reply complying with the requirements noted below. Failure to timely comply will result in ABANDONMENT of this application.
THIS THREE-MONTH PERIOD IS NOT EXTENDABLE.

5. ☐ CORRECTED DRAWINGS (as "replacement sheets") must be submitted.
   ☐ including changes required by the attached Examiner's Amendment / Comment or in the Office action of Paper No./Mail Date _____ .

   Identifying indicia such as the application number (see 37 CFR 1.84(c)) should be written on the drawings in the front (not the back) of each sheet. Replacement sheet(s) should be labeled as such in the header according to 37 CFR 1.121(d).

6. ☐ DEPOSIT OF and/or INFORMATION about the deposit of BIOLOGICAL MATERIAL must be submitted. Note the attached Examiner's comment regarding REQUIREMENT FOR THE DEPOSIT OF BIOLOGICAL MATERIAL.

Attachment(s)

1. ☑ Notice of References Cited (PTO-892)
2. ☑ Information Disclosure Statements (PTO/SB/08), Paper No./Mail Date _____.
3. ☐ Examiner's Comment Regarding Requirement for Deposit of Biological Material _____.
4. ☑ Interview Summary (PTO-413), Paper No./Mail Date. _____.
5. ☑ Examiner's Amendment/Comment
6. ☑ Examiner's Statement of Reasons for Allowance
7. ☐ Other _____.

/Catherine Thiaw/
Primary Examiner, Art Unit 2493

Notice of Pre-AIA or AIA Status

The present application, filed on or after March 16, 2013, is being examined under the first inventor to file provisions of the AIA.

1. The present Office Action is responsive to communication received 7/21/2024.

Information Disclosure Statement

2. The information disclosure statement (IDS) submitted 7/21/2024 is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statement are being considered by the examiner.

Examiner's Amendments

3. An examiner's amendment to the record appears below. Should the changes and/or additions be unacceptable to applicant, an amendment may be filed as provided by 37 CFR 1.312. To ensure consideration of such an amendment, it MUST be submitted no later than the payment of the issue fee.

4. Authorization for this examiner's amendment was given in a communication with Attorney of Record Brian Galvin on 9/25/2024.

5. Please replace the claim listing with the following one. Claims 1-8, 11-15 are amended, claims 9-10 and 16-18 are cancelled.

1. (Currently Amended) A ~~system~~ distributed computing cluster comprising:

~~a distributed computing cluster comprising~~ a first plurality of computer systems,

wherein each respective computer system of the first plurality of computer systems comprises a memory that stores a respective first data,

wherein the respective first data represents a respective portion of a distributed computational graph,

and wherein the distributed computational graph describes a flow of output data of a first transformation pipeline to an input of a second transformation pipeline,

wherein a first computer system of the first plurality of computer systems is configured to:

receive a first stream of input data from a[[n]] first input feed,

process the first stream of input data substantially in real time by executing software instructions that apply the first transformation pipeline to the first stream of input data to generate first pipeline output messages,

process the respective first data stored in the memory of the first computer system to determine information about the second transformation pipeline,

and transmit the first pipeline output messages to a second computer system of the first plurality of computer systems in accordance with the determined information,

wherein the second computer system is configured to:

receive the first pipeline output messages,

and process the first pipeline output messages substantially in real time by executing software instructions that apply the second transformation pipeline to the first pipeline output messages to generate second pipeline output messages,

wherein the first and second computer systems are distinct; and

a second plurality of computer systems;

wherein a third computer system of the first plurality of computer systems is configured to execute software instructions that cause a fourth computer system of the second plurality of computer systems to execute software instructions that apply at least one of the first transformation pipeline and the second transformation pipeline.

2. (Currently Amended) The ~~system~~ distributed computing cluster of claim 1, wherein at least one of the first transformation pipeline and the second transformation pipeline is non-linear.

3. (Currently Amended) The ~~system~~ distributed computing cluster of claim 1,

wherein the first computer system is further configured to transmit the first pipeline output messages to ~~a third~~ the fourth computer system of the ~~first~~ second plurality of computer systems,

wherein the first computer system is distinct from the ~~third~~ fourth computer system,

and wherein the second computer system is distinct from the ~~third~~ fourth computer system.

4. (Currently Amended) The ~~system~~ distributed computing cluster of claim [[2]] 1,

wherein the first computer system is further configured to transmit the first pipeline output messages to another ~~third~~ computer system of the ~~first~~ second plurality of computer systems[[,]]

~~wherein the first computer system is distinct from the third computer system,~~

~~and wherein the second computer system is distinct from the third computer system~~.

5. (Currently Amended) The ~~system~~ distributed computing cluster of claim 2,

wherein [[a]] the third computer system of the first plurality of computer systems is configured to:

identify a ~~fourth~~ fifth computer system of the first plurality of computer systems,

wherein the ~~fourth~~ fifth computer system is not configured to apply the first transformation pipeline to any stream of input data,

and cause the ~~fourth~~ fifth computer system to process a second stream of input data received from a[[n]] second input feed substantially in real time by executing software instructions that apply the first transformation pipeline to the second stream of input data.

6. (Currently Amended) The ~~system~~ distributed computing cluster of claim 5, wherein the third computer system is further configured to execute software instructions that:

identify one or more of the respective memories that store respective first data representing at least a respective portion of the distributed computational graph,

wherein the respective first data includes information about the first transformation pipeline,

and store in the identified one or more memories information identifying the ~~fourth~~ fifth computer system.

7. (Currently Amended) The ~~system~~ distributed computing cluster of claim 2,

wherein [[a]] the third computer system of the first plurality of computer systems is configured to:

identify a ~~fourth~~ fifth computer system of the first plurality of computer systems,

wherein the ~~fourth~~ fifth computer system is not configured to apply the second transformation pipeline to any pipeline output message,

and cause the ~~fourth~~ fifth computer system to process third pipeline output messages substantially in real time by executing software instructions that apply the second transformation pipeline to the third pipeline output messages.

8. (Currently Amended) The ~~system~~ distributed computing cluster of claim 7, wherein the third computer system is further configured to execute software instructions that:

identify one or more of the respective memories that store respective first data representing at least a respective portion of the distributed computational graph,

wherein the respective first data includes information about the second transformation pipeline,

and store in the identified one or more memories information identifying the ~~fourth~~ fifth computer system.

9. (Canceled)

10. (Canceled)

11. (Currently Amended) A ~~system~~ distributed computing cluster comprising:

~~a distributed computing cluster comprising~~ a first plurality of computer systems,

wherein each respective computer system of the first plurality of computer systems comprises a respective memory that stores a respective first data that:

describes at least a respective portion of a configuration of a first transformation pipeline,

and identifies at least a second transformation pipeline,

wherein a first computer system of the first plurality of computer systems is configured to:

receive a first stream of input data from a[[n]] first input feed,

process the first stream of input data substantially in real time by executing software instructions that apply the first transformation pipeline to the first stream of input data to generate first pipeline output messages,

process the respective first data to determine an identification of the second transformation pipeline,

and transmit the first pipeline output messages to a second computer system of the first plurality of computer systems in accordance with the determined identification,

wherein the second computer system is configured to:

receive the first pipeline output messages,

and process the first pipeline output messages substantially in real time by executing software instructions that apply the second transformation pipeline to the first pipeline output messages,

wherein the first and second computer systems are distinct; and

a second plurality of computer systems;

wherein a third computer system of the first plurality of computer systems is configured to execute software instructions that cause the second plurality of computer systems to execute at least one of the first transformation pipeline and the second transformation pipeline.

12. (Currently Amended) The ~~system~~ distributed computing cluster of claim 11, wherein one or more of the first transformation pipeline and the second transformation pipeline is non-linear.

13. (Currently Amended) The ~~system~~ distributed computing cluster of claim 11,

wherein the first computer system is further configured to transmit the first pipeline output messages to a ~~third~~ fourth computer system of the ~~first~~ second plurality of computer systems,

wherein the first computer system is distinct from the ~~third~~ fourth computer system,

and wherein the second computer system is distinct from the ~~third~~ fourth computer system.

14. (Currently Amended) The ~~system~~ distributed computing cluster of claim 12,

wherein [[a]] the third computer system of the first plurality of computer systems is configured to:

identify a fourth computer system of the first plurality of computer systems,

wherein the fourth computer system is not configured to apply the first transformation pipeline to any stream of input data,

and cause the fourth computer system to process a second stream of input data received from a[[n]] second input feed substantially in real time by executing software instructions that apply the first transformation pipeline to the second stream of input data.

15. (Currently Amended) The ~~system~~ distributed computing cluster of claim 14, wherein the third computer system is further configured to execute software instructions that:

identify one or more of the respective memories that stores respective data that describes a respective portion of a configuration of the first transformation pipeline,

and store in the identified one or more memories information identifying the fourth computer system.

16. (Canceled)

17. (Canceled)

18. (Canceled)

Allowed Claims

6. Claims 1-8, 11-15 are allowed, in view of the examiner's amendments above.

Terminal Disclaimer

7. The terminal disclaimer filed on 9/27/2024 disclaiming the terminal portion of any patent granted on this application which would extend beyond the expiration date of a patent that would be granted for application 18/779043 is accepted.

Reasons for Allowance

8. The closest prior arts of the record are *US 20090327668 to Sudzilouski, hereinafter Sudzilouski, and Publication "FlumeJava: easy, efficient data-parallel pipelines." by Chambers, et al. ACM Sigplan Notices 45.6 (2010): 363-375, hereinafter Chambers.*

Sudzilouski discloses:

A distributed computing cluster comprising: a first plurality of computer systems, wherein each respective computer system of the first plurality of computer systems comprises a memory that stores a respective first data *(Fig. 1 [0017][0019], cluster of computers 102A … 102N, each comprising a processing units 104, to read code/data from memory 112 using a process spawning an execution thread ([0031])*, **wherein the respective first data represents a respective portion of a distributed computation**[al graph] *([0031]: although no computational graph is taught*

explicitly, each system executes a part of the process, see Fig. 2, step executed by 104 A ,…, 104N), **and wherein the distributed computation**[al graph] **describes a flow of output data of a first transformation pipeline to an input of a second transformation pipeline** *([0042][0044]: the threads executes tasks arranged in a pipeline, where the output of one pipeline becomes input for the other pipeline)*, **wherein a first computer system of the first plurality of computer systems is configured to: receive a first stream of input data from a first input feed, process the first stream of input data substantially in real time by executing software instructions that apply the first transformation pipeline to the first stream of input data to generate first pipeline output messages** *([0042]: load data into pipeline "I/O, decompression, parsing", generate tokens 220 (Fig.2, 3))*, **process the respective first data stored in the memory of the first computer system to determine information about the second transformation pipeline** *([0061]:number of tasks executed into a sequence of execution, organized through the pipeline; [0063][0066]: set of one or more tokens generated as output)*, **and transmit the first pipeline output messages to a second computer system of the first plurality of computer systems in accordance with the determined information** *([Fig. 3: token 220s i.e output of load pipeline is fed to pipeline for application thread)*, **wherein the second computer system is configured to: receive the first pipeline output messages, and process the first pipeline output messages substantially in real time by executing software instructions that apply the second transformation pipeline to the first pipeline output messages to generate second pipeline output messages** *([0049]: another*

set of tokens is output at the completion of the second pipeline), **wherein the first and second computer systems are distinct** (Fig. 2: 104a different from 104n).

Although Sudzilouski discloses a distributed computation of processes in pipelines, Sudzilouski does not explicitly teach **a distributed computational graph**, *or the other limitations of the claim.*

Chambers describes **computational graph** *called "execution plan graph structure" (p.363, on right). The graph comprises instructions to execute stages of a pipeline (Fig.re 1, p.366). It would have been obvious to a skilled artisan before the instant application was filed to implement computational graph to implement the pipelines as taught by Chambers, because it provides "a high-level abstraction for parallel data and computation … to yield and -easy-to-use system" (see Abstract).*

Neither Sudzilouski alone or in combination with Chambers, Lin or any other prior art of the record discloses:

a second plurality of computer systems; wherein a third computer system of the first plurality of computer systems is configured to execute software instructions that cause a fourth computer system of the second plurality of computer systems to execute software instructions that apply at least one of the first transformation pipeline and the second transformation pipeline, *as recited in claim 1 and substantially in claim 11. Claims 2-8, 12-15, respectively dependent from claim 1 and claim 11, are equally found allowable.*

Other relevant prior arts of the records are:

Lin 20160275123 discloses a cluster of nodes comprising a plurality of nodes and a pipeline manager; each node also includes a job tracker. The pipeline manager monitor success of the individual pipelines, and may add additional pipelines if needed The pipeline manager is connected to the cluster by a network 204, which could be the Internet, a WAN, LAN, a direct wired connection. That means the monitoring can be local to the cluster (direct wired connection) or remote to the cluster (thru the Internet).

Llmas et al 20140351827 disclose the execution of pipestages included in a pipeline; output of the execution can be consumed by a downstream pipestage.

Eldar et al 9886273 disclose a pipeline architecture comprising multiple pipelines, each including a plurality of stages, the output of the last stage of a first pipeline feeds the first stage of a second pipeline.

Huang et al 20130073573 disclose processing a query using a pipeline of sequential operations

Undy et al 6629167 disclose decoupling at least two multi-stage pipelines using buffers to stage the result of pipelines stages.

Alapati et al 9348602 disclose an allocation unit that determines the number of execution pipelines that are active any point in time, based on computing resources.

Markam et al 20130067558 disclose a first and second pipelines, each comprising virtual machines communicating from one pipeline to the other.

Le Vasseur et al 20160323216 disclose a monitor service that instantiate and manage a pipeline comprising multiple workflows executing on a plurality of devices.

The monitor performs failover procedures to keep instances of workflow running in case of a device's failure.

Hegar et al 9479720 disclose a device including an active input pipeline that feeds an output pipeline; a standby pipeline processes other input from other redundant sources when the active pipeline fails.

Shau et al 20150261580 disclose determining dependencies between a plurality of nodes in a first cluster, determining an order of tasks for the nodes, creating an execution plan that divides the cluster management operation into the plurality of nodes in the determined order.

Snell et al 20080288621 disclose a system for defining, executing, and managing network workflow. One or more agents execute on one or more devices in the network, and execute a set of instructions. A workflow manager selects which agents evaluate and execute subsets of the instructions. In one embodiment, the property of "clustering" specifies whether all or specific agents are to run a workflow object, and under what circumstances. In a round-robin cluster, each subsequent run of the workflow object executes on the next agent in the group. In a fail-over cluster, a next agent takes over in the group, and so on, until the workflow object executes successfully or there are no more agents to try.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to CATHERINE B THIAW whose telephone number is (571)270-1138. The examiner can normally be reached Monday-Friday 7am-4pm.

Examiner interviews are available via telephone, in-person, and video conferencing using a USPTO supplied web-based collaboration tool. To schedule an interview, applicant is encouraged to use the USPTO Automated Interview Request (AIR) at http://www.uspto.gov/interviewpractice.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, CARL G COLIN can be reached on 571-272-3862. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of published or unpublished applications may be obtained from Patent Center. Unpublished application information in Patent Center is available to registered users. To file and manage patent submissions in Patent Center, visit: https://patentcenter.uspto.gov. Visit https://www.uspto.gov/patents/apply/patent-center for more information about Patent Center and https://www.uspto.gov/patents/docx for information about filing in DOCX format. For additional questions, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

/Catherine Thiaw/
Primary Examiner, Art Unit 2493
9/26/2024

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), by mail or fax, or via the USPTO patent electronic filing system.

By mail, send to: Mail Stop ISSUE FEE
Commissioner for Patents
P.O. Box 1450
Alexandria, Virginia 22313-1450

By fax, send to: (571)-273-2885

INSTRUCTIONS: This form should be used for transmitting the ISSUE FEE and PUBLICATION FEE (if required). Blocks 1 through 5 should be completed where appropriate. All further correspondence will be mailed to the current correspondence address as indicated unless corrected below or directed otherwise in Block 1, by (a) specifying a new correspondence address; and/or (b) indicating a separate "FEE ADDRESS" for maintenance fee notifications. **Because electronic patent issuance may occur shortly after issue fee payment, any desired continuing application should preferably be filed prior to payment of this issue fee in order not to jeopardize copendency.**

CURRENT CORRESPONDENCE ADDRESS (Note: Use Block 1 for any change of address)

82318 7590 09/29/2024

Brian Galvin
Galvin Patent Law LLC
2916 NW Bucklin Hill Road
Suite 485
Silverdale, WA 98383

Note: A certificate of mailing can only be used for domestic mailings of the Fee(s) Transmittal. This certificate cannot be used for any other accompanying papers. Each additional paper, such as an assignment or formal drawing, must have its own certificate of mailing or transmission.

Certificate of Mailing or Transmission

I hereby certify that this Fee(s) Transmittal is being deposited with the United States Postal Service with sufficient postage for first class mail in an envelope addressed to the Mail Stop ISSUE FEE address above, or being transmitted to the USPTO via the USPTO patent electronic filing system or by facsimile to (571) 273-2885, on the date below.

(Typed or printed name)
(Signature)
(Date)

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
18/779,029	07/21/2024	Jason Crabtree	15305FT	4119

TITLE OF INVENTION: RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	UNDISCOUNTED	$1200	$0.00	$0.00	$1200	12/30/2024

EXAMINER	ART UNIT	CLASS-SUBCLASS
THIAW, CATHERINE B	2493	718-102000

1. Change of correspondence address or indication of "Fee Address" (37 CFR 1.363).

☐ Change of correspondence address (or Change of Correspondence Address form PTO/AIA/122 or PTO/SB/122) attached.

☐ "Fee Address" indication (or "Fee Address" Indication form PTO/AIA/47 or PTO/SB/47; Rev 03-02 or more recent) attached. **Use of a Customer Number is required.**

2. For printing on the patent front page, list
(1) The names of up to 3 registered patent attorneys or agents OR, alternatively,
(2) The name of a single firm (having as a member a registered attorney or agent) and the names of up to 2 registered patent attorneys or agents. If no name is listed, no name will be printed.

1 Galvin Patent Law LLC
2 Brian R. Galvin
3

3. ASSIGNEE NAME AND RESIDENCE DATA TO BE PRINTED ON THE PATENT (print or type)

PLEASE NOTE: Unless an assignee is identified below, no assignee data will appear on the patent. If an assignee is identified below, the document must have been previously recorded, or filed for recordation, as set forth in 37 CFR 3.11 and 37 CFR 3.81(a). Completion of this form is NOT a substitute for filing an assignment.

(A) NAME OF ASSIGNEE: QOMPLX LLC

(B) RESIDENCE: (CITY and STATE OR COUNTRY): Reston, VA

Please check the appropriate assignee category or categories (will not be printed on the patent) : ☐ Individual ☒ Corporation or other private group entity ☐ Government

4a. Fees submitted: ☒ Issue Fee ☐ Publication Fee (if required)

4b. Method of Payment: *(Please first reapply any previously paid fee shown above)*

☒ Electronic Payment via the USPTO patent electronic filing system ☐ Enclosed check ☐ Non-electronic payment by credit card (Attach form PTO-2038)

☐ The Director is hereby authorized to charge the required fee(s), any deficiency, or credit any overpayment to Deposit Account No. _____

5. **Change in Entity Status** (from status indicated above)

☐ Applicant certifying micro entity status. See 37 CFR 1.29

NOTE: Absent a valid certification of Micro Entity Status (see forms PTO/SB/15A and 15B), issue fee payment in the micro entity amount will not be accepted at the risk of application abandonment.

☐ Applicant asserting small entity status. See 37 CFR 1.27

NOTE: If the application was previously under micro entity status, checking this box will be taken to be a notification of loss of entitlement to micro entity status.

☐ Applicant changing to regular undiscounted fee status.

NOTE: Checking this box will be taken to be a notification of loss of entitlement to small or micro entity status, as applicable.

NOTE: This form must be signed in accordance with 37 CFR 1.31 and 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Authorized Signature /BRIAN R. GALVIN/ Date October 3, 2024

Typed or printed name Brian R. Galvin Registration No. 50,966

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC PAYMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
18/779,029	**10/03/2024 08:06:01 PM Z ET**	**15305FT**

Title of Invention

RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	-
CONFIRMATION #	4119	FILED BY	Aura Morrey
PATENT CENTER #	67432297	AUTHORIZED BY	Brian Galvin
CUSTOMER #	82318	FILING DATE	07/21/2024
CORRESPONDENCE ADDRESS	-	FIRST NAMED INVENTOR	Jason Crabtree

Payment Information

PAYMENT METHOD	PAYMENT TRANSACTION ID	PAYMENT AUTHORIZED BY
CARD / 1009	**E202403K07346430**	**Aura Morrey**

FEE CODE	DESCRIPTION	ITEM PRICE($)	QUANTITY	ITEM TOTAL($)
1501	UTILITY ISSUE FEE	1200.00	1	1200.00
			TOTAL AMOUNT:	**$1,200.00**

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage

submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES
PATENT AND TRADEMARK OFFICE

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC ACKNOWLEDGEMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
18/779,029	**10/03/2024 08:06:01 PM Z ET**	**15305FT**

Title of Invention

RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	-
CONFIRMATION #	4119	FILED BY	Aura Morrey
PATENT CENTER #	67432297	FILING DATE	07/21/2024
CUSTOMER #	82318	FIRST NAMED INVENTOR	Jason Crabtree
CORRESPONDENCE ADDRESS	-	AUTHORIZED BY	Brian Galvin

Documents

TOTAL DOCUMENTS: 1

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
241003_15305FT_Issue Fee_AS_FILED.pdf	1	Issue Fee Payment (PTO-85B)	320 KB

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)
241003_15305FT_Issue Fee_AS_FILED.pdf	53D07B67B477A6E7CC80F5DC158EDDA4EA1ED7BDE89041D86352FC00446E28727FBAC0F0777E3F19D6D1C674690EB7097FF2760D37546492E517AB89F626715F

This Acknowledgement Receipt evidences receipt on the noted date by the USPTO of the indicated documents, characterized by the applicant, and including page counts, where applicable. It serves as evidence of receipt similar to a Post Card, as described in MPEP 503.

New Applications Under 35 U.S.C. 111

If a new application is being filed and the application includes the necessary components for filing date (see 37 CFR 1.53(b)-(d) and MPEP 506), a Filing Receipt (37 CFR 1.54) will be issued in due course and the date shown on this Acknowledgement Receipt will establish the filing date of the application

National Stage of an International Application under 35 U.S.C. 371

If a timely submission to enter the national stage of an international application is compliant with the conditions of 35 U.S.C. 371 and other applicable requirements a Form PCT/DO/EO/903 indicating acceptance of the application as a national stage submission under 35 U.S.C. 371 will be issued in addition to the Filing Receipt, in due course.

New International Application Filed with the USPTO as a Receiving Office

If a new international application is being filed and the international application includes the necessary components for an international filing date (see PCT Article 11 and MPEP 1810), a Notification of the International Application Number and of the International Filing Date (Form PCT/RO/105) will be issued in due course, subject to prescriptions concerning national security, and the date shown on this Acknowledgement Receipt will establish the international filing date of the application.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	ISSUE DATE	PATENT NO.	ATTORNEY DOCKET NO.	CONFIRMATION NO.
18/779,029	11/12/2024	12143424	15305FT	4119

82318 7590 10/23/2024

Brian Galvin
Galvin Patent Law LLC
2916 NW Bucklin Hill Road
Suite 485
Silverdale, WA 98383

ISSUE NOTIFICATION

The projected patent number and issue date are specified above. The patent will issue electronically. The electronically issued patent is the official patent grant pursuant to 35 U.S.C. § 153. The patent may be accessed on or after the issue date through Patent Center at https://patentcenter.uspto.gov/. The patent will be available in both the public and the private sides of Patent Center. Further assistance in electronically accessing the patent, or about Patent Center, is available by calling the Patent Electronic Business Center at 1-888-217-9197.

The USPTO is implementing electronic patent issuance with a transition period, during which period the USPTO will mail a ceremonial paper copy of the electronic patent grant to the correspondence address of record. Additional copies of the patent (i.e., certified and presentation copies) may be ordered for a fee from the USPTO's Certified Copy Center at https://certifiedcopycenter.uspto.gov/index.html. The Certified Copy Center may be reached at (800)972-6382.

Determination of Patent Term Adjustment under 35 U.S.C. 154 (b)
(application filed on or after May 29, 2000)

The Patent Term Adjustment is 0 day(s). Any patent to issue from the above-identified application will include an indication of the adjustment on the front page.

If a Continued Prosecution Application (CPA) was filed in the above-identified application, the filing date that determines Patent Term Adjustment is the filing date of the most recent CPA.

Applicant will be able to obtain more detailed information by accessing the Patent Center (https://patentcenter.uspto.gov).

Any questions regarding the Patent Term Extension or Adjustment determination should be directed to the Office of Patent Legal Administration at (571)-272-7702. Questions relating to issue and publication fee payments should be directed to the Application Assistance Unit (AAU) of the Office of Patents Stakeholder Experience (OPSE), Stakeholder Support Division (SSD) at (571)-272-4200.

INVENTOR(s) (Please see PATENT CENTER site https://patentcenter.uspto.gov for additional inventors):

Jason Crabtree, Vienna, VA;
Andrew Sellers, Monument, CO;

APPLICANT(s) (Please see PATENT CENTER site https://patentcenter.uspto.gov for additional applicants):

QOMPLX LLC, Reston, VA;

The United States represents the largest, most dynamic marketplace in the world and is an unparalleled location for business investment, innovation, and commercialization of new technologies. The USA offers tremendous resources and advantages for those who invest and manufacture goods here. Through SelectUSA, our nation works to encourage and facilitate business investment. To learn more about why the USA is the best country in the world to develop technology, manufacture products, and grow your business, visit SelectUSA.gov.

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
18/779,029	07/21/2024	Jason Crabtree	15305FT	4119

82318 7590 11/12/2024

Brian Galvin
Galvin Patent Law LLC
2916 NW Bucklin Hill Road
Suite 485
Silverdale, WA 98383

EXAMINER	
THIAW, CATHERINE B	

ART UNIT	PAPER NUMBER
2407	

NOTIFICATION DATE	DELIVERY MODE
11/12/2024	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

USPTO@GALVINPATENTLAW.COM
eofficeaction@appcoll.com

APPLICATION NO.	ISSUE DATE	PATENT NO.
18/779,029	12-Nov-2024	12143424

Brian Galvin
Galvin Patent Law LLC
2916 NW Bucklin Hill Road
Silverdale, WA 98383

EGRANT NOTIFICATION

Your electronic patent grant (eGrant) is now available, which can be accessed via Patent Center at https://patentcenter.uspto.gov

The electronic patent grant is the official patent grant under 35 U.S.C. 153. For more information, please visit https://www.uspto.gov/electronicgrants

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
18/779,029	07/21/2024	Jason Crabtree	15305FT

CONFIRMATION NO. 4119

82318
Brian Galvin
Galvin Patent Law LLC
2916 NW Bucklin Hill Road
Suite 485
Silverdale, WA 98383

PUBLICATION NOTICE

OC000000081465394

Date Mailed: 12/05/2024

Title:RAPID PREDICTIVE ANALYSIS OF VERY LARGE DATA SETS USING THE DISTRIBUTED COMPUTATIONAL GRAPH

Publication No.US-2024-0396942-A1
Publication Date:11/28/2024

NOTICE OF PUBLICATION OF APPLICATION

The above-identified application will be electronically published as a patent application publication pursuant to 37 CFR 1.211, et seq. The patent application publication number and publication date are set forth above.

The publication may be viewed using the USPTO's publicly available Searchable Databases via the Patent Public Search tool at www.uspto.gov. The direct link to access the Patent Public Search tool is currently https://ppubs.uspto.gov/pubwebapp/static/pages/ppubsbasic.html.

The publication process established by the Office does not provide for mailing a copy of the publication to applicant. A copy of the publication may be obtained from the Office upon payment of the appropriate fee set forth in 37 CFR 1.19(a)(1). Orders for copies of patent application publications are handled by the USPTO's Public Records Division. The Public Records Division can be reached by telephone at (571) 272-3150 or (800) 972-6382, by facsimile at (571) 273-3250, by mail addressed to the United States Patent and Trademark Office, Public Records Division, Alexandria, VA 22313-1450 or via the Internet.

In addition, information on the status of the application, including the mailing date of Office actions and the dates of receipt of correspondence filed in the Office, may also be accessed via the Internet through Patent Center, the USPTO's electronic patent application filing and management system. The direct link to access this status information is currently https://patentcenter.uspto.gov/. Prior to publication, such status information is confidential and may only be obtained by applicant using the private side of Patent Center.

Further assistance in electronically accessing the publication, or about Patent Center, is available by calling the Patent Electronic Business Center at 1-866-217-9197.

Office of Data Managment, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101