PTO/AIA/15 (10-17)
Approved for use through 11/30/2020. 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.

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No.	FERM0003CON200564319
First Named Inventor	Robert Bismuth
Title	Analytics, Algorithm Architecture, and Data Processing System and Method
Priority Mail Express® Label No.	

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

1. [✓] **Fee Transmittal Form** (PTO/SB/17 or equivalent)
2. [✓] **Applicant asserts small entity status.** See 37 CFR 1.27
3. [ ] **Applicant certifies micro entity status.** See 37 CFR 1.29. Applicant must attach form PTO/SB/15A or B or equivalent.
4. [✓] **Specification** [*Total Pages* 73] Both the claims and abstract must start on a new page. *(See MPEP § 608.01(a) for information on the preferred arrangement)*
5. [✓] **Drawing(s)** (35 U.S.C. 113) [*Total Sheets* 29]
6. **Inventor's Oath or Declaration** [*Total Pages* ____] *(including substitute statements under 37 CFR 1.64 and assignments serving as an oath or declaration under 37 CFR 1.63(e))*
   a. [ ] Newly executed (original or copy)
   b. [✓] A copy from a prior application (37 CFR 1.63(d))
7. [✓] **Application Data Sheet** ** See note below.* See 37 CFR 1.76 (PTO/AIA/14 or equivalent)
8. **CD-ROM or CD-R** in duplicate, large table, or Computer Program (*Appendix*)
   [ ] Landscape Table on CD
9. **Nucleotide and/or Amino Acid Sequence Submission** (*if applicable, items a. – c. are required*)
   a. [ ] Computer Readable Form (CRF)
   b. [ ] Specification Sequence Listing on:
      i. [ ] CD-ROM or CD-R (2 copies); or
      ii. [ ] Paper
   c. [ ] Statements verifying identity of above copies

ADDRESS TO:

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

ACCOMPANYING APPLICATION PAPERS

10. [ ] **Assignment Papers** (cover sheet & document(s))
    Name of Assignee _____
11. [ ] **37 CFR 3.73(c) Statement** (*when there is an assignee*) [ ] **Power of Attorney**
12. [ ] **English Translation Document** (*if applicable*)
13. [ ] **Information Disclosure Statement** (PTO/SB/08 or PTO-1449)
    [ ] Copies of citations attached
14. [ ] **Preliminary Amendment**
15. [ ] **Return Receipt Postcard** (*MPEP § 503*) (*Should be specifically itemized*)
16. [ ] **Certified Copy of Priority Document(s)** (*if foreign priority is claimed*)
17. [ ] **Nonpublication Request** Under 35 U.S.C. 122(b)(2)(B)(i). Applicant must attach form PTO/SB/35 or equivalent.
18. [ ] **Other:** _____

***Note**: (1) Benefit claims under 37 CFR 1.78 and foreign priority claims under 1.55 **must** be included in an Application Data Sheet (ADS).
(2) For applications filed under 35 U.S.C. 111, the application must contain an ADS specifying the applicant if the applicant is an assignee, person to whom the inventor is under an obligation to assign, or person who otherwise shows sufficient proprietary interest in the matter. See 37 CFR 1.46(b).

19. CORRESPONDENCE ADDRESS

[✓] The address associated with Customer Number: 27799 **OR** [ ] Correspondence address below

Name					
Address					
City		State		Zip Code	
Country		Telephone		Email	

Signature	/Victor J. Castellucci/	Date	2022-12-06
Name (Print/Type)	Victor J. Castellucci	Registration No. (Attorney/Agent)	43535

This collection of information is required by 37 CFR 1.53(b). 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 12 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.

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.

PTO/AIA/14 (07-14)
Approved for use through 04/30/2017. 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	FERM0003CON200564319
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

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
	Robert		Bismuth	

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

City	Seattle	State/Province	WA	Country of Residence	US

Mailing Address of Inventor:

Address 1	2113 West Wheeler St.
Address 2	

City	Seattle	State/Province	WA
Postal Code	98199	Country	US

Inventor 2 Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Mike		Stengle	

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

City	Seattle	State/Province		Country of Residence	WA

Mailing Address of Inventor:

Address 1	Ackerstrasse 30
Address 2	4057

City	Basel	State/Province	
Postal Code		Country	CH

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).

PTO/AIA/14 (07-14)
Approved for use through 04/30/2017. 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	FERM0003CON200564319
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

☐ An Address is being provided for the correspondence Information of this application.			
Customer Number	27799		
Email Address	NYIPDocketing@cozen.com	Add Email	Remove Email

Application Information:

Title of the Invention	Analytics, Algorithm Architecture, and Data Processing System and Method		
Attorney Docket Number	FERM0003CON200564319	**Small Entity Status Claimed**	☒
Application Type	Nonprovisional		
Subject Matter	Utility		
Total Number of Drawing Sheets (if any)	29	**Suggested Figure for Publication (if any)**	

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	27799		

Application Data Sheet 37 CFR 1.76	Attorney Docket Number	FERM0003CON200564319
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

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 this 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 blank.

Prior Application Status	Pending		Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
	Continuation of	17666049	2022-02-07

Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
	Continuation of	16729727	2019-12-30	11249690	2022-02-15

Prior Application Status	Expired		Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
	Claims benefit of provisional	62882044	2019-08-02
Prior Application Status	Expired		Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
	Claims benefit of provisional	62853764	2019-05-29
Prior Application Status	Expired		Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
	Claims benefit of provisional	62801839	2019-02-06

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

Foreign Priority Information:

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.

PTO/AIA/14 (07-14)
Approved for use through 04/30/2017. 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	FERM0003CON200564319
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

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.

Authorization to Permit Access:

☐ Authorization to Permit Access to the Instant Application by the Participating Offices

If 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 World Intellectual Property Office (WIPO), and any other intellectual property offices in which a foreign application claiming priority to the instant patent application is filed access to the instant patent application. See 37 CFR 1.14(c) and (h). This box should not be checked if the applicant does not wish the EPO, JPO, KIPO, WIPO, or other intellectual property office in which a foreign application claiming priority to the instant patent application is filed to have access to the instant patent application.

In accordance with 37 CFR 1.14(h)(3), access will be provided to a copy of the instant patent application with respect to: 1) the instant patent application-as-filed; 2) any foreign application to which the instant patent application claims priority under 35 U.S.C. 119(a)-(d) if a copy of the foreign application that satisfies the certified copy requirement of 37 CFR 1.55 has been filed in the instant patent application; and 3) any U.S. application-as-filed from which benefit is sought in the instant patent application.

In accordance with 37 CFR 1.14(c), access may be provided to information concerning the date o f filing this Authorization.

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.

PTO/AIA/14 (07-14)
Approved for use through 04/30/2017. 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	FERM0003CON200564319
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

Applicant 1

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	Fermat International, Inc.

Mailing Address Information For Applicant:

Address 1	27/29 rue Raffet		
Address 2			
City	Paris	**State/Province**	
Country	FR	Postal Code	75016
Phone Number		Fax Number	
Email Address			

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

Assignee Information including Non-Applicant Assignee Information:

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

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.

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

PTO/AIA/14 (07-14)
Approved for use through 04/30/2017. 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	FERM0003CON200564319
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

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.

Signature:

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Signature	/Victor Castellucci/			Date (YYYY-MM-DD)	2022-12-06
First Name	Victor	Last Name	Castellucci	Registration Number	43535

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

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 C o o p eration 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.

DocuSign Envelope ID: 94CD8E88-A59E-4B7E-9F50-7CE1A06664EA

PTO/AIA/01 (06-12)
Approved for use through 11/30/2020. 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	Analytics, Algorithm Architecture, and Data Processing System and Method

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 16729727

filed on DECEMBER 30, 2019.

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: Mike Stengle Date (Optional): 1/1/2020

Signature: DocuSigned by: Mike Stengle 7E27D578E5154CD...

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.

DocuSign Envelope ID: C9D0DA58-534D-403C-B01E-7ED843CD675B

PTO/AIA/01 (06-12)
Approved for use through 11/30/2020. 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	Analytics, Algorithm Architecture, and Data Processing System and Method

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 16729727

filed on December 30, 2019.

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: Robert Bismuth DocuSigned by: Date (Optional): 12/31/2019

Signature: Robert Bismuth E6DB403AF2F349E...

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.

FIG. 1

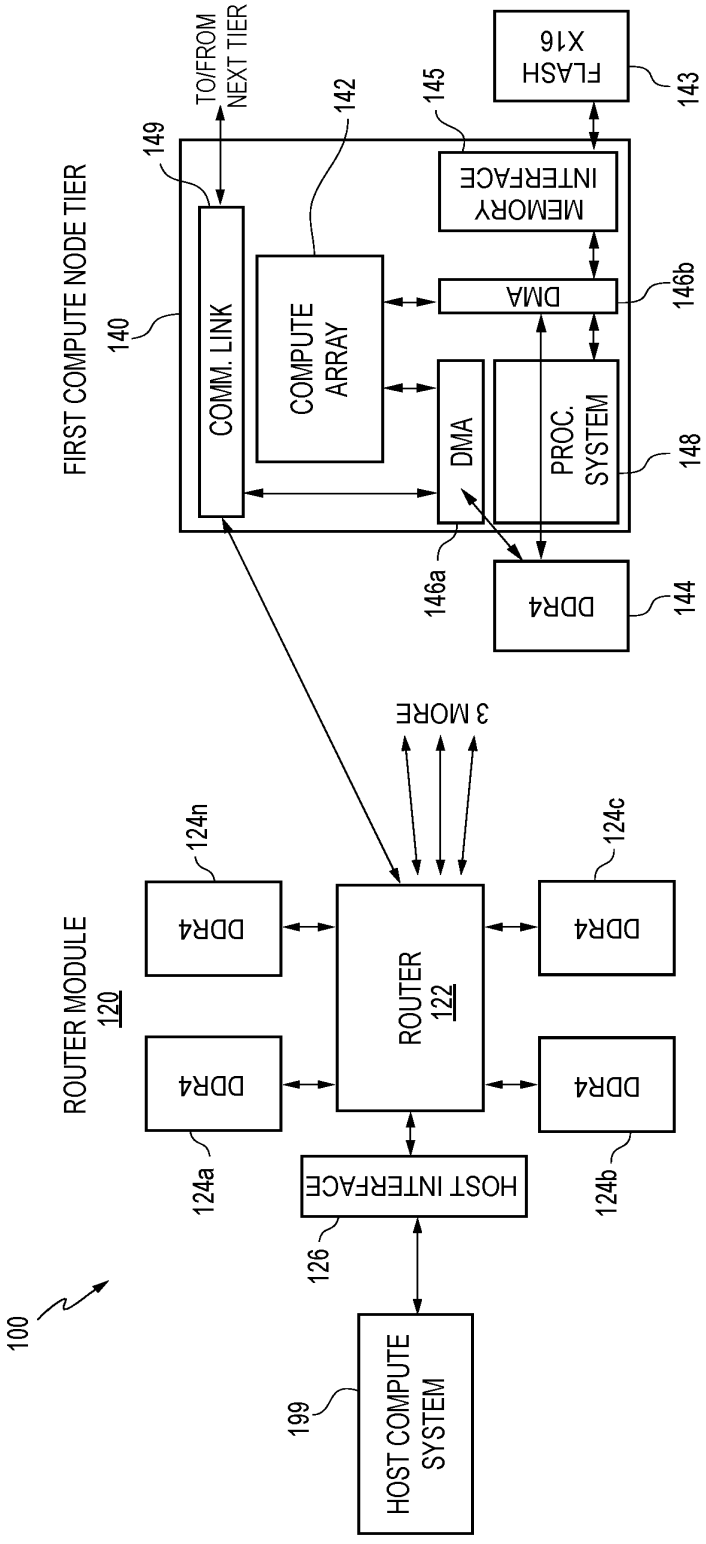

100
HOST COMPUTE SYSTEM
199
HOST INTERFACE
126
ROUTER MODULE
120
ROUTER
122
DDR4
124a
DDR4
124b
DDR4
124c
DDR4
124n
3 MORE
FIRST COMPUTE NODE TIER
140
COMM. LINK
149
TO/FROM NEXT TIER
COMPUTE ARRAY
142
DMA
146a
PROC. SYSTEM
148
DMA
146b
MEMORY INTERFACE
145
FLASH X16
143
DDR4
144

FIG. 2A

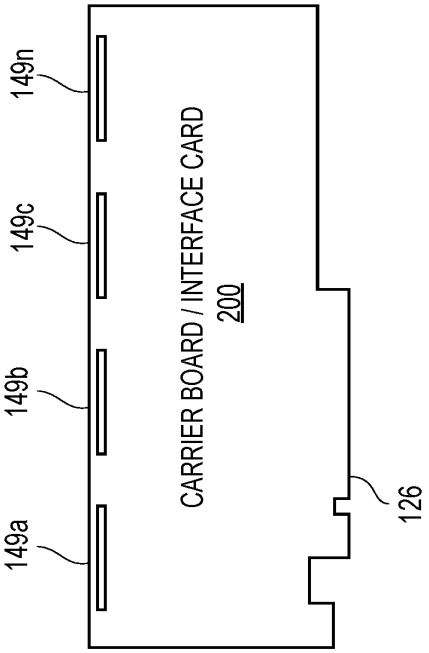


FIG. 2C

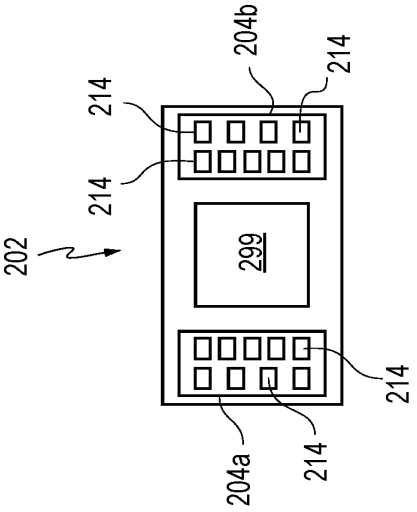
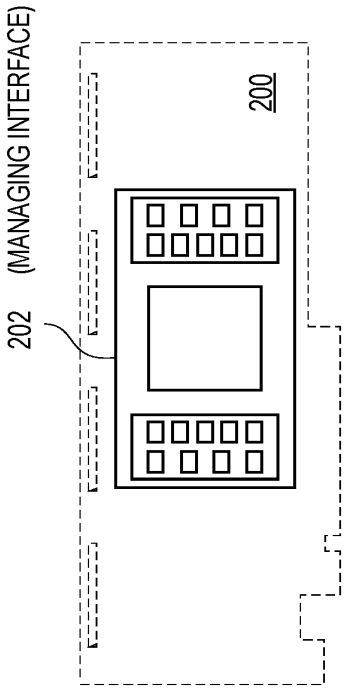


FIG. 2B



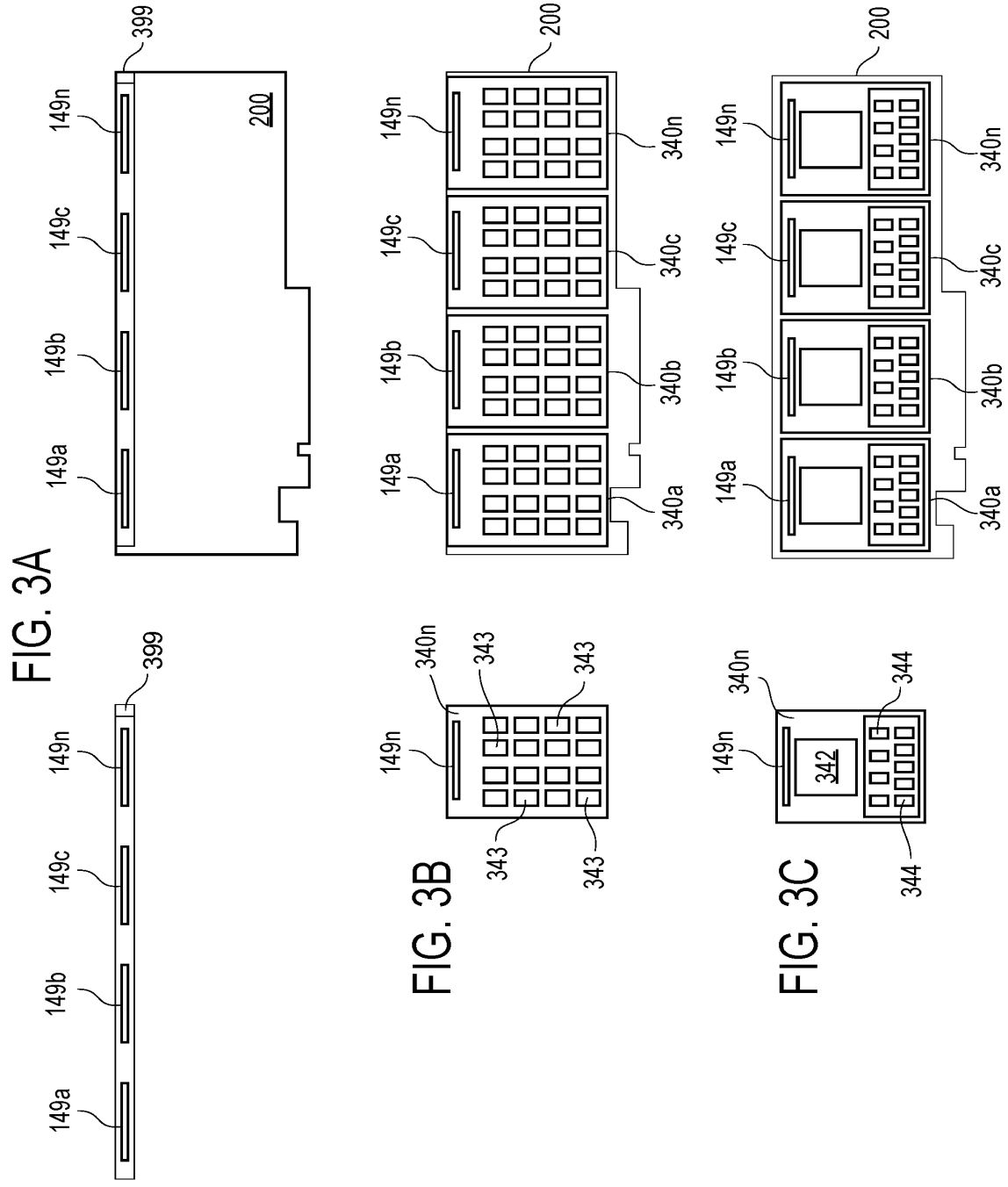


FIG. 3A

FIG. 3B

FIG. 3C

FIG. 4

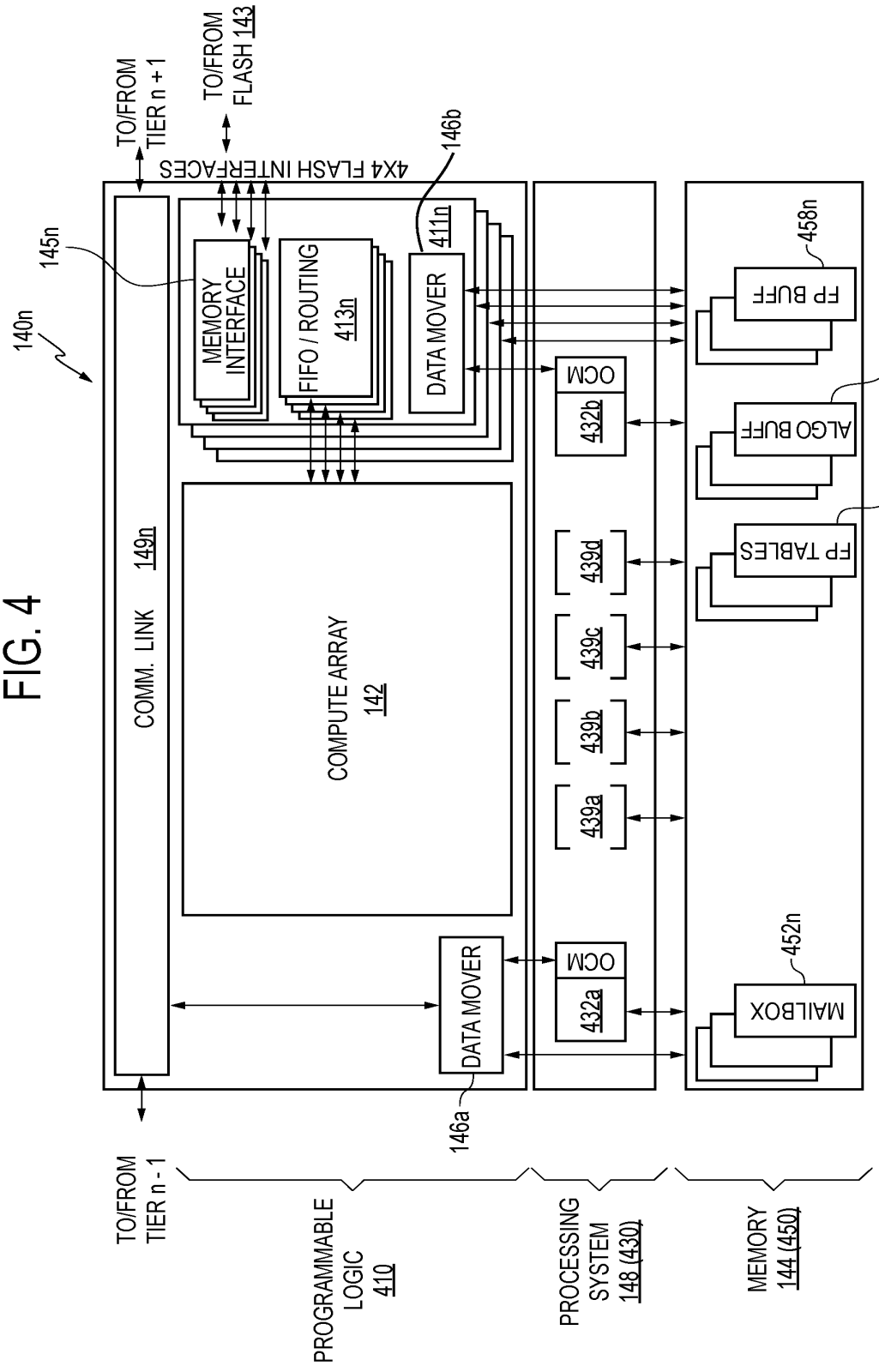

TO/FROM TIER n - 1
TO/FROM TIER n + 1
TO/FROM FLASH 143
4X4 FLASH INTERFACES
140n
145n
146b
146a
COMM. LINK 149n
MEMORY INTERFACE
FIFO / ROUTING 413n
DATA MOVER 411n
COMPUTE ARRAY 142
DATA MOVER
OCM 432a
OCM 432b
439a
439b
439c
439d
FP BUFF 458n
ALGO BUFF 456n
FP TABLES 454n
MAILBOX 452n
PROGRAMMABLE LOGIC 410
PROCESSING SYSTEM 148 (430)
MEMORY 144 (450)

FIG. 5

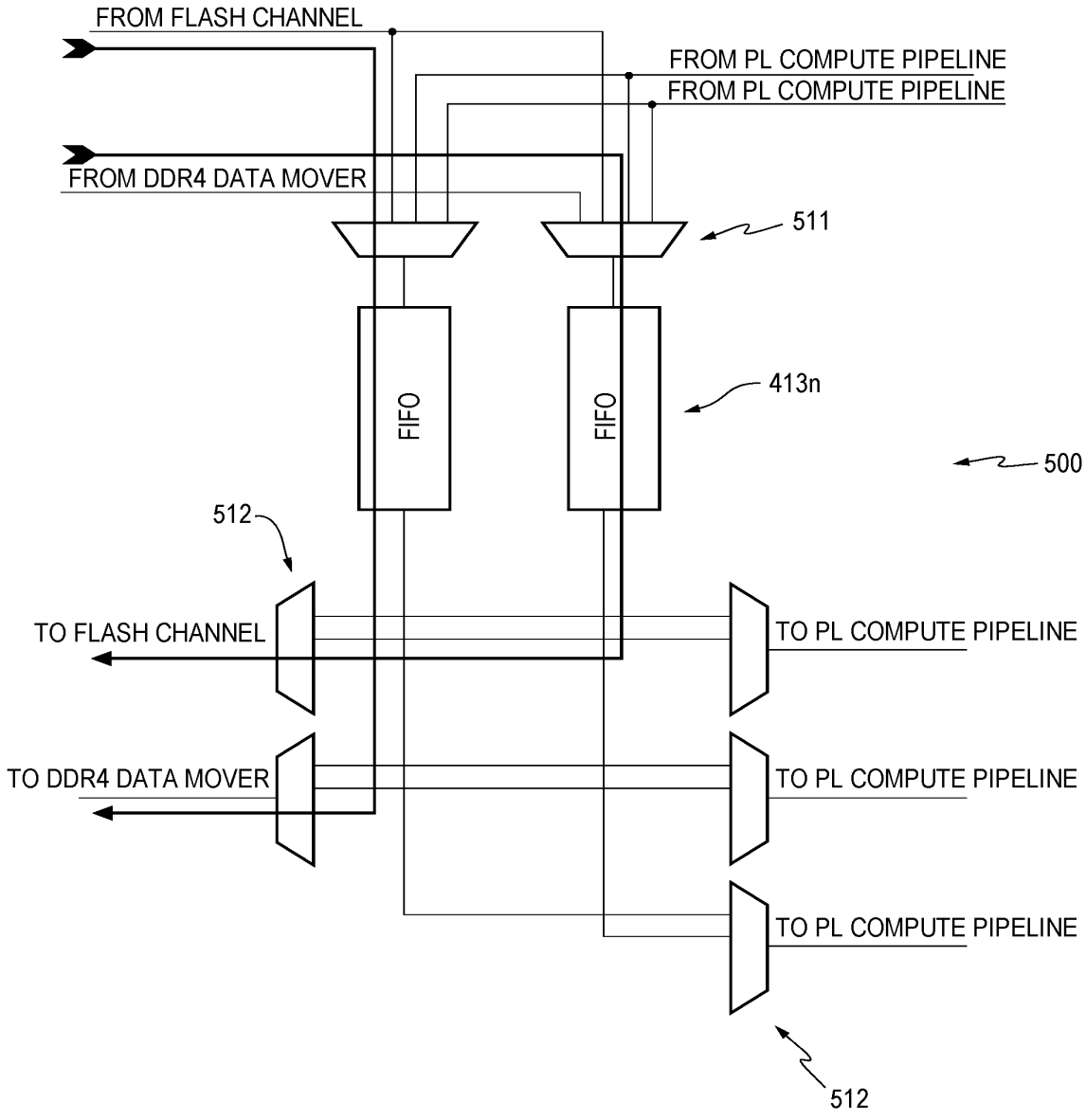

FROM FLASH CHANNEL
FROM PL COMPUTE PIPELINE
FROM PL COMPUTE PIPELINE
FROM DDR4 DATA MOVER
511
FIFO
FIFO
413n
500
512
TO FLASH CHANNEL
TO PL COMPUTE PIPELINE
TO DDR4 DATA MOVER
TO PL COMPUTE PIPELINE
TO PL COMPUTE PIPELINE
512

FIG. 6

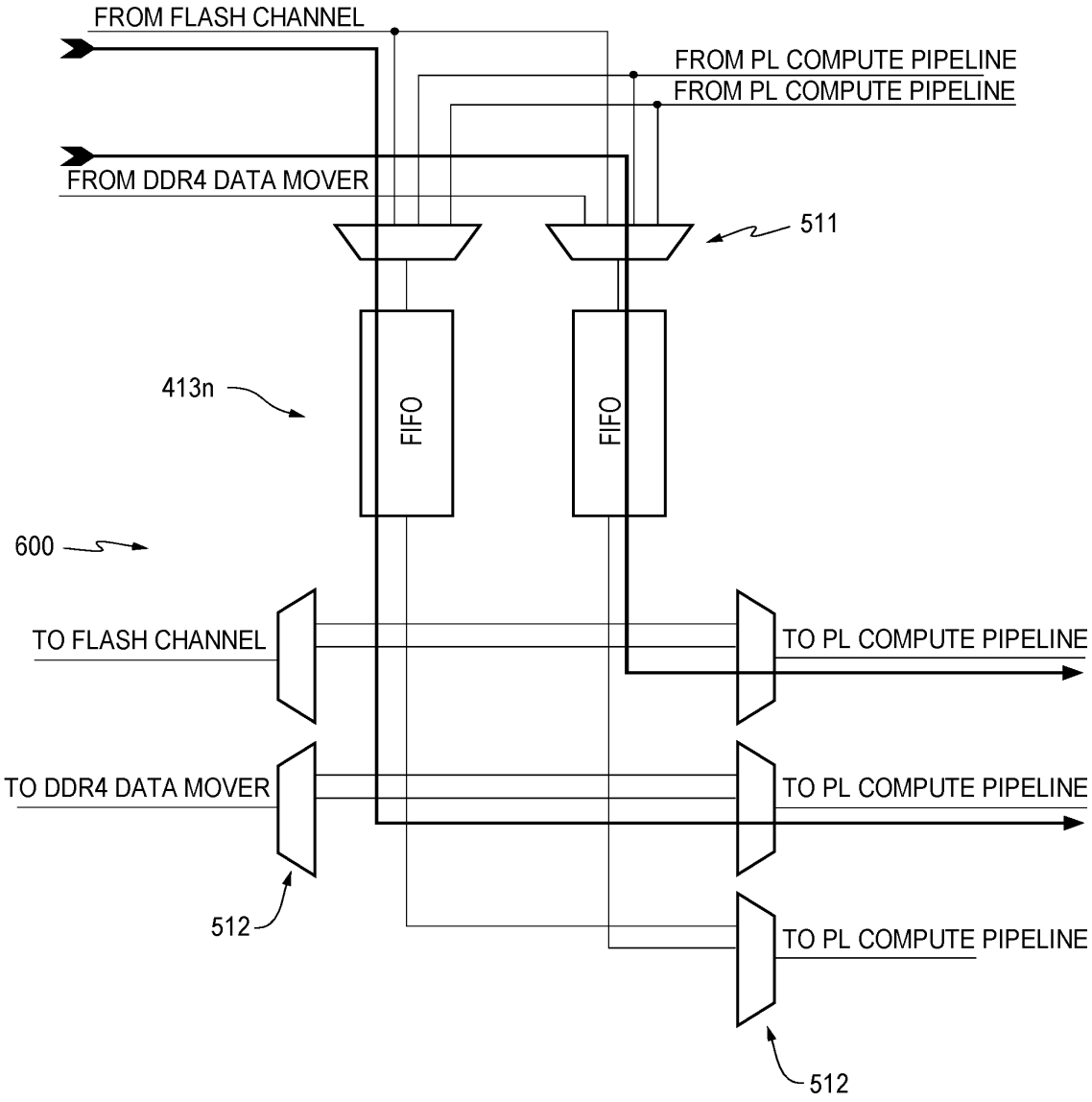

FROM FLASH CHANNEL
FROM PL COMPUTE PIPELINE
FROM PL COMPUTE PIPELINE
FROM DDR4 DATA MOVER
511
413n
FIFO
FIFO
600
TO FLASH CHANNEL
TO PL COMPUTE PIPELINE
TO DDR4 DATA MOVER
TO PL COMPUTE PIPELINE
512
TO PL COMPUTE PIPELINE
512

FIG. 7

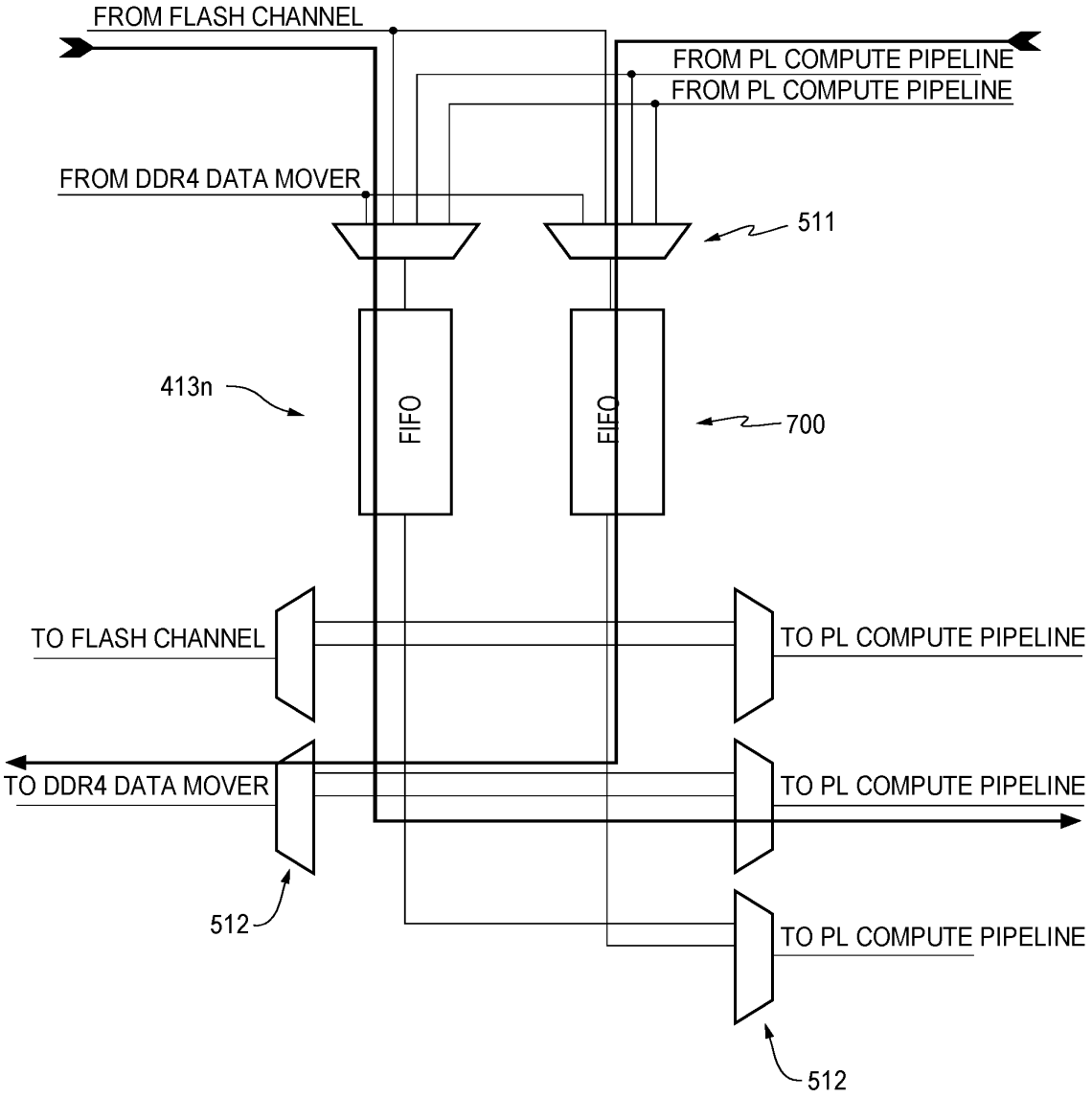

FROM FLASH CHANNEL
FROM PL COMPUTE PIPELINE
FROM PL COMPUTE PIPELINE
FROM DDR4 DATA MOVER
511
413n
FIFO
FIFO
700
TO FLASH CHANNEL
TO PL COMPUTE PIPELINE
TO DDR4 DATA MOVER
TO PL COMPUTE PIPELINE
512
TO PL COMPUTE PIPELINE
512

FIG. 8

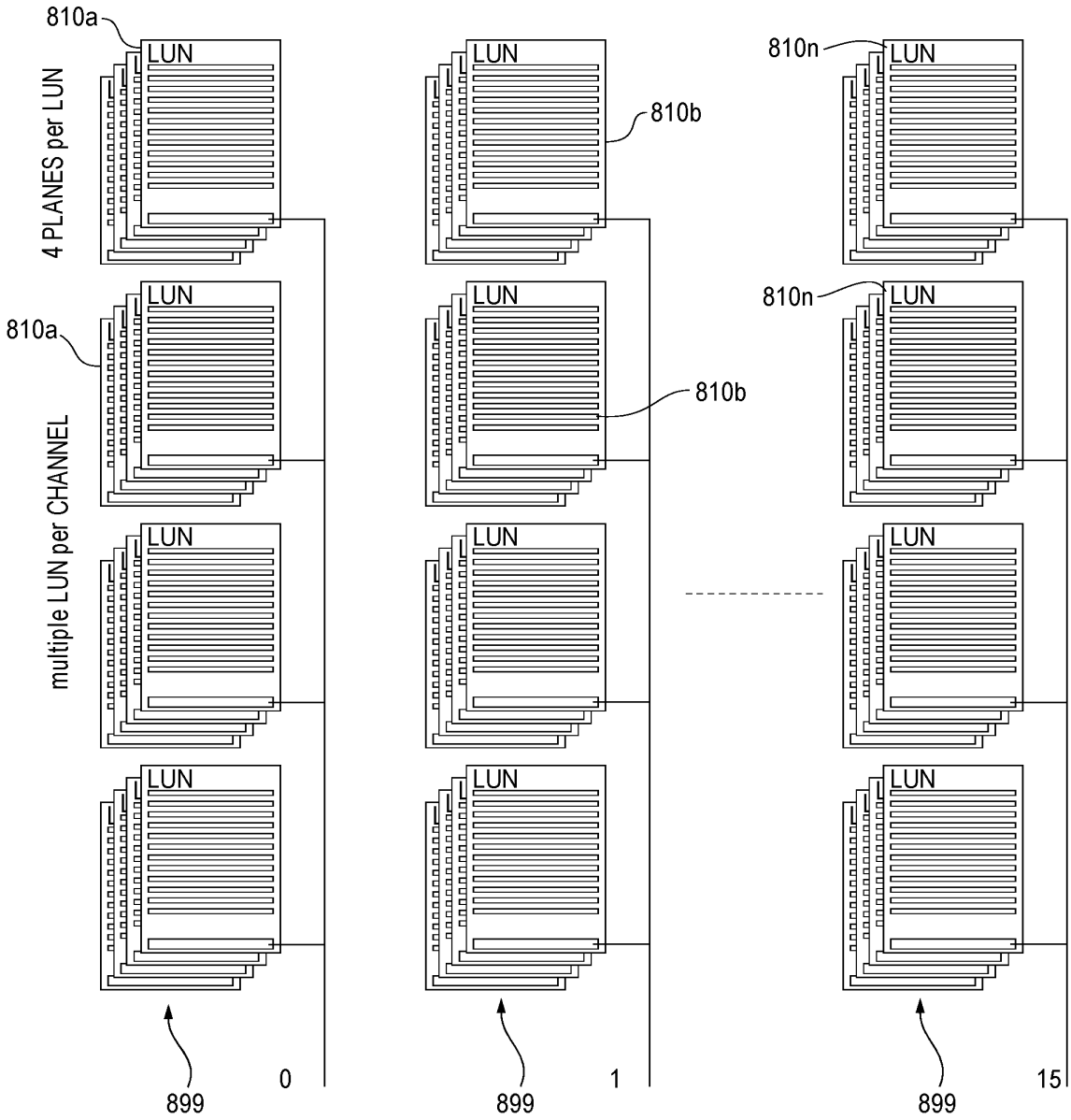

810a
LUN
4 PLANES per LUN
multiple LUN per CHANNEL
810b
810n
899
0
1
15

FIG. 9

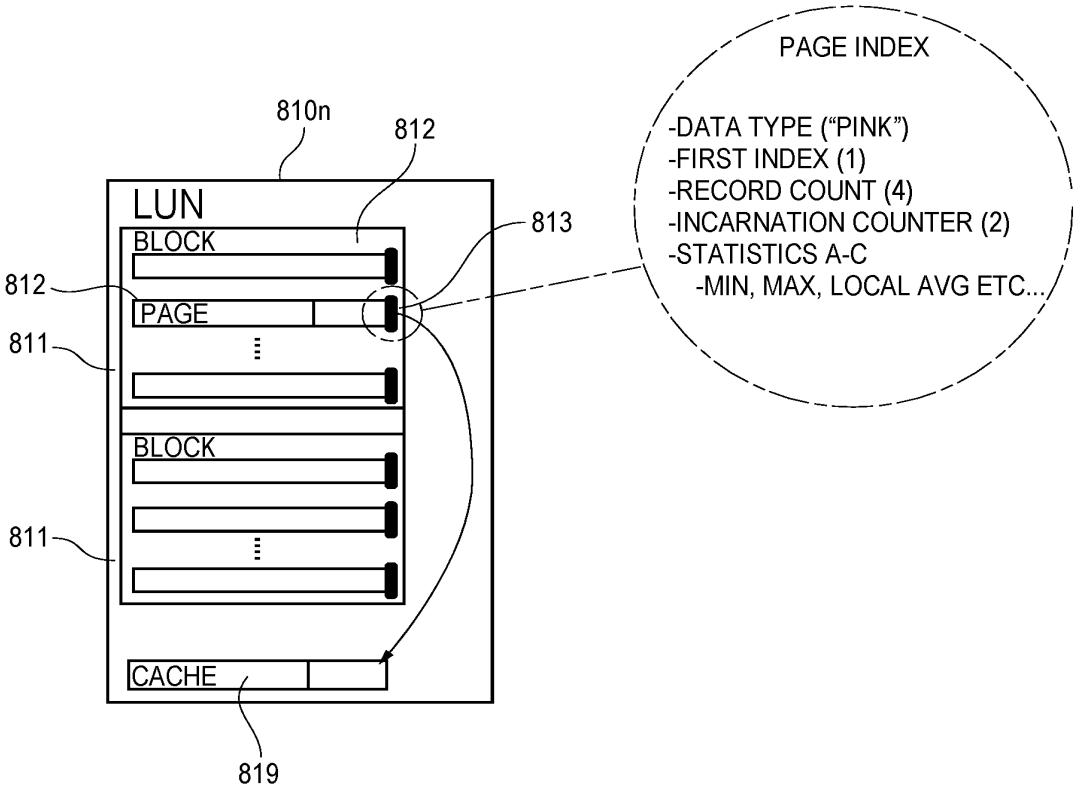

PAGE INDEX
-DATA TYPE ("PINK")
-FIRST INDEX (1)
-RECORD COUNT (4)
-INCARNATION COUNTER (2)
-STATISTICS A-C
-MIN, MAX, LOCAL AVG ETC...
810n
812
LUN
BLOCK
813
812
PAGE
811
BLOCK
811
CACHE
819

FIG. 10
PRIOR ART

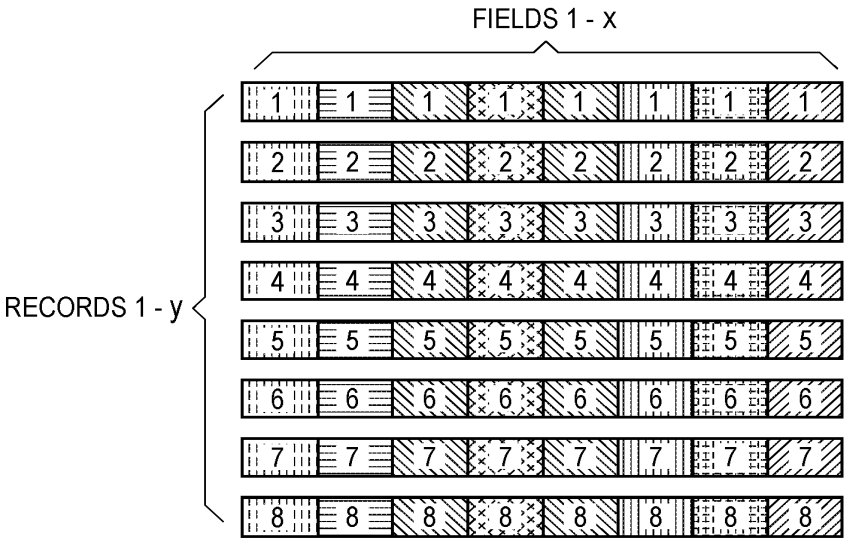


FIG. 11

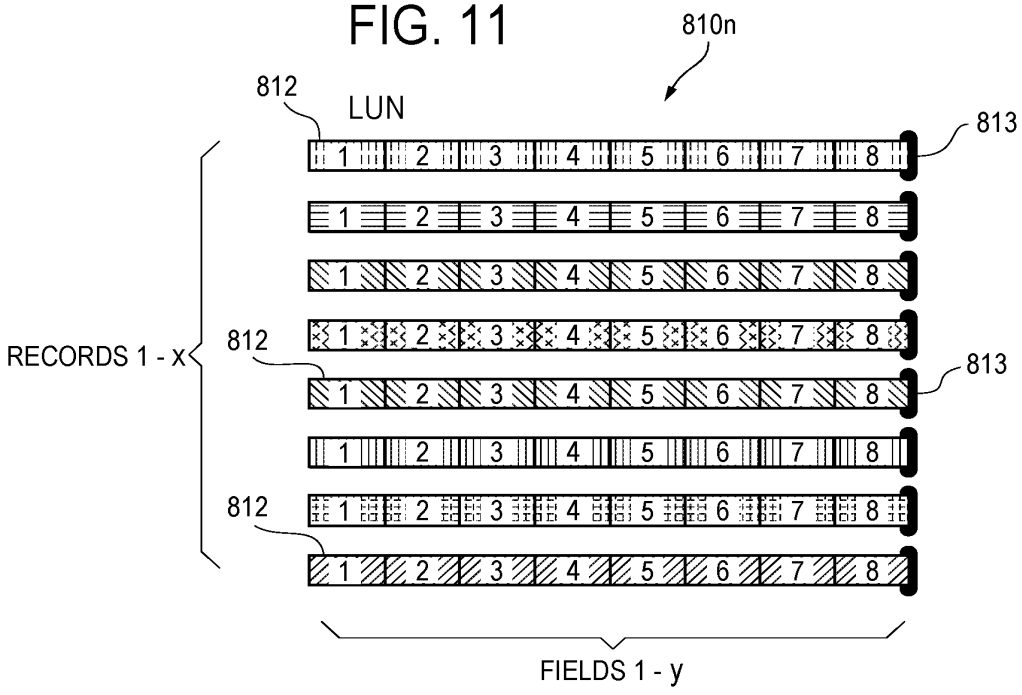


FIG. 12

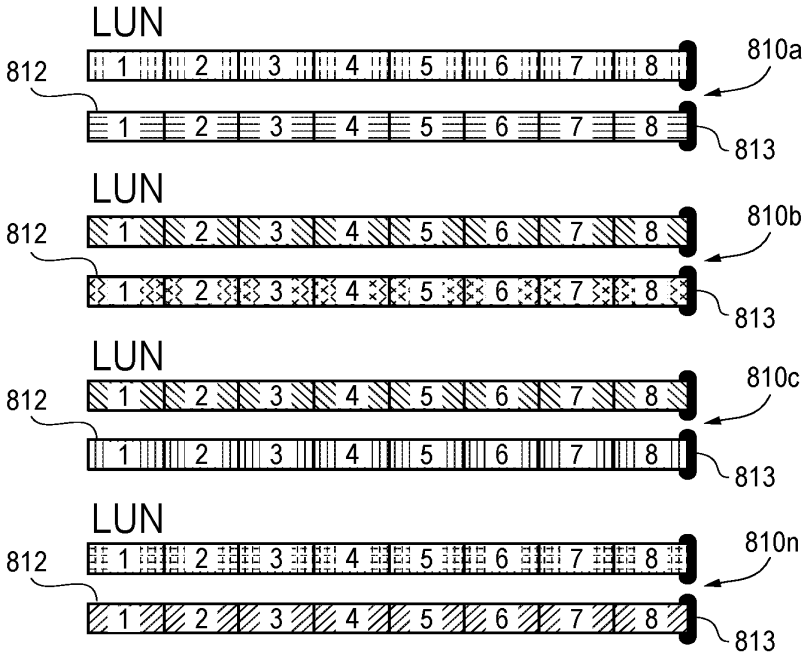

LUN
812
1 2 3 4 5 6 7 8
810a
1 2 3 4 5 6 7 8
813
LUN
812
1 2 3 4 5 6 7 8
810b
1 2 3 4 5 6 7 8
813
LUN
812
1 2 3 4 5 6 7 8
810c
1 2 3 4 5 6 7 8
813
LUN
812
1 2 3 4 5 6 7 8
810n
1 2 3 4 5 6 7 8
813

FIG. 13

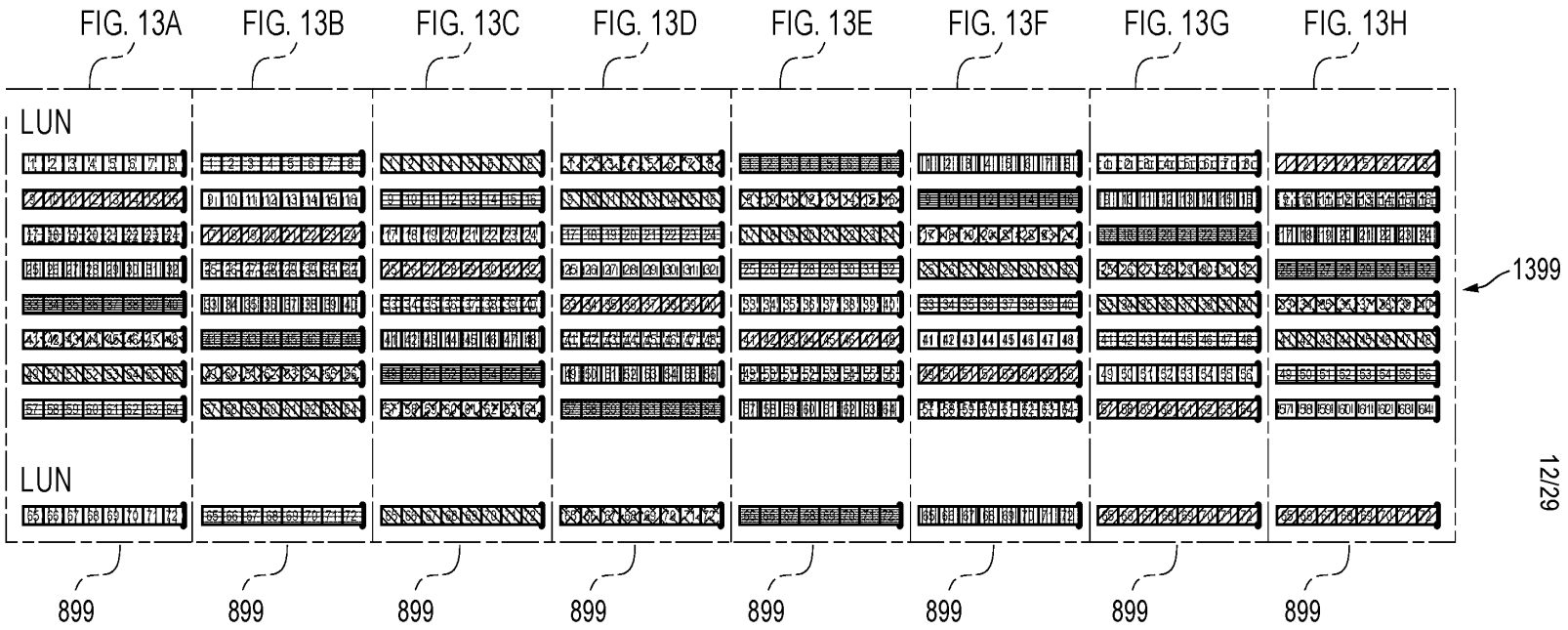

FIG. 13A
FIG. 13B
FIG. 13C
FIG. 13D
FIG. 13E
FIG. 13F
FIG. 13G
FIG. 13H
LUN
LUN
1399
899
899
899
899
899
899
899
899

FIG. 14

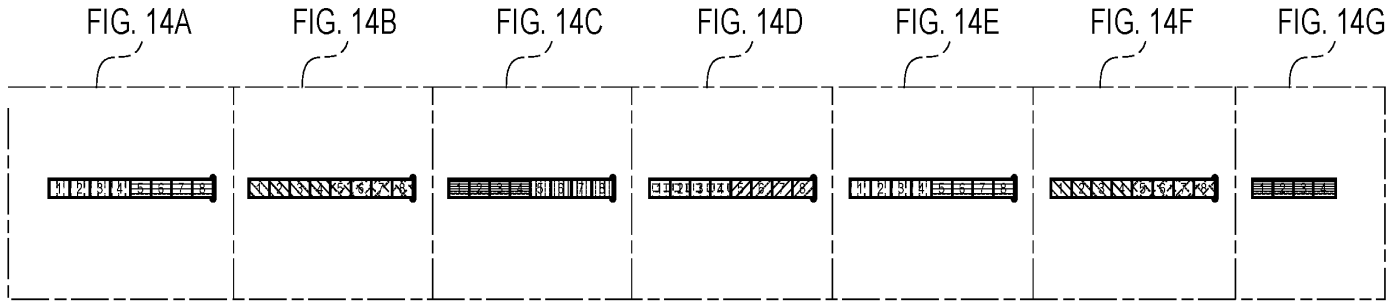

FIG. 14A
FIG. 14B
FIG. 14C
FIG. 14D
FIG. 14E
FIG. 14F
FIG. 14G

FIG. 13A

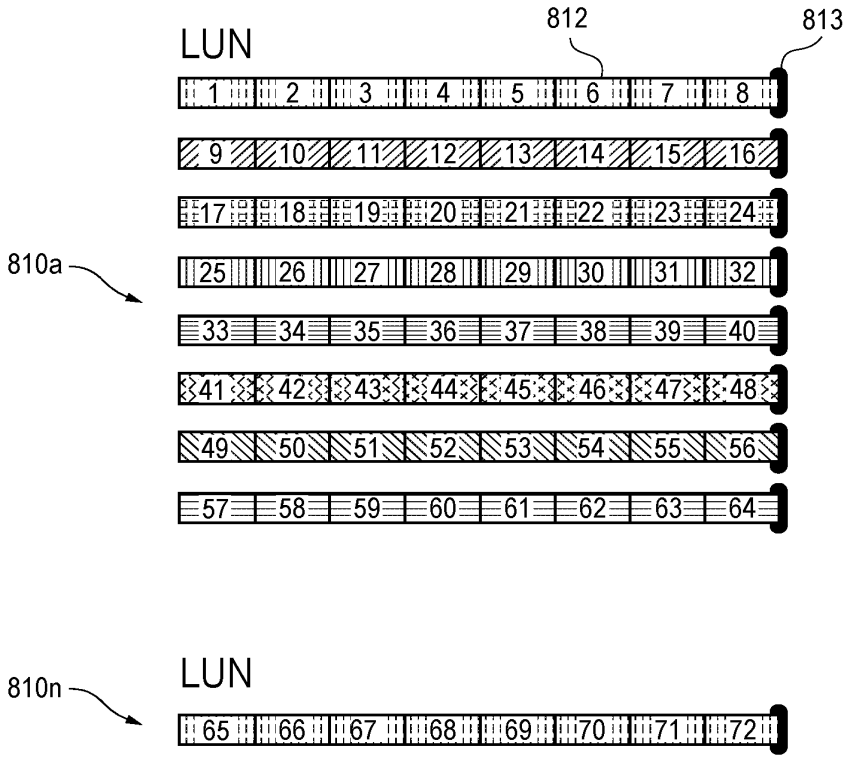

812
813
LUN
1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
810a
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56
57 58 59 60 61 62 63 64
LUN
810n
65 66 67 68 69 70 71 72

FIG. 13B

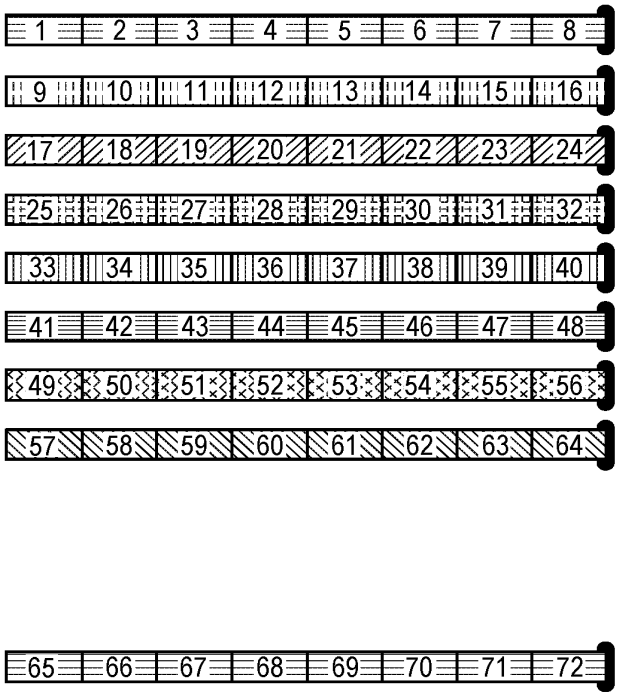

1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56
57 58 59 60 61 62 63 64
65 66 67 68 69 70 71 72

FIG. 13C

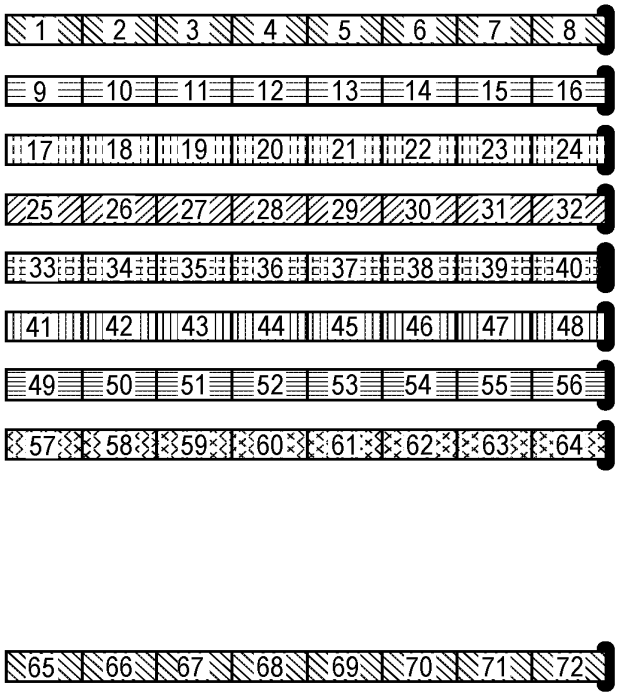

1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56
57 58 59 60 61 62 63 64
65 66 67 68 69 70 71 72

FIG. 13D

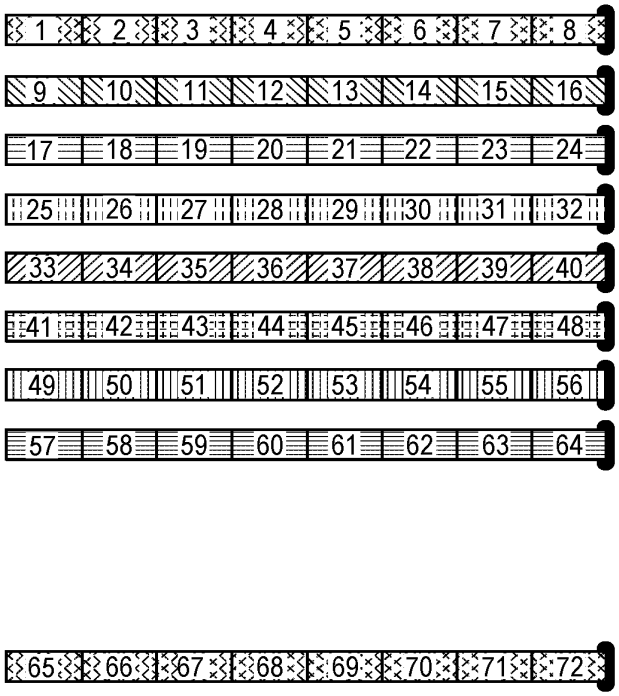

1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56
57 58 59 60 61 62 63 64
65 66 67 68 69 70 71 72

FIG. 13E

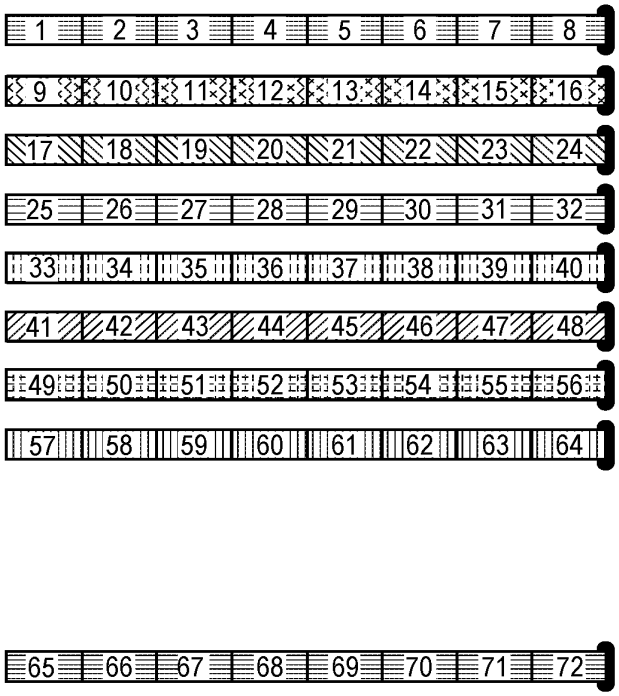

1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56
57 58 59 60 61 62 63 64
65 66 67 68 69 70 71 72

FIG. 13F

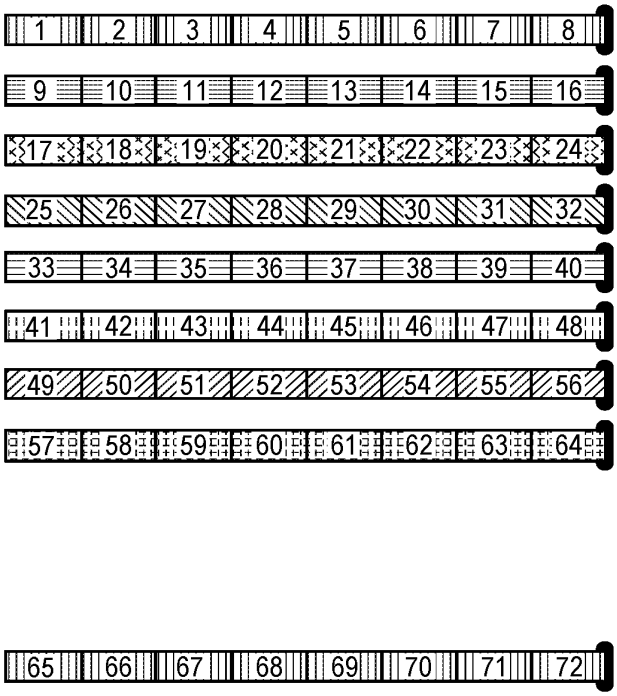

1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56
57 58 59 60 61 62 63 64
65 66 67 68 69 70 71 72

FIG. 13G

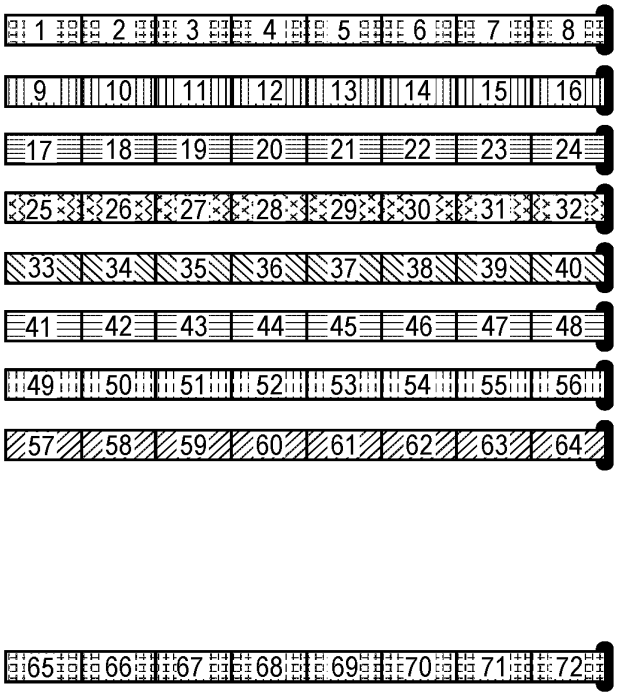

1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56
57 58 59 60 61 62 63 64
65 66 67 68 69 70 71 72

FIG. 13H

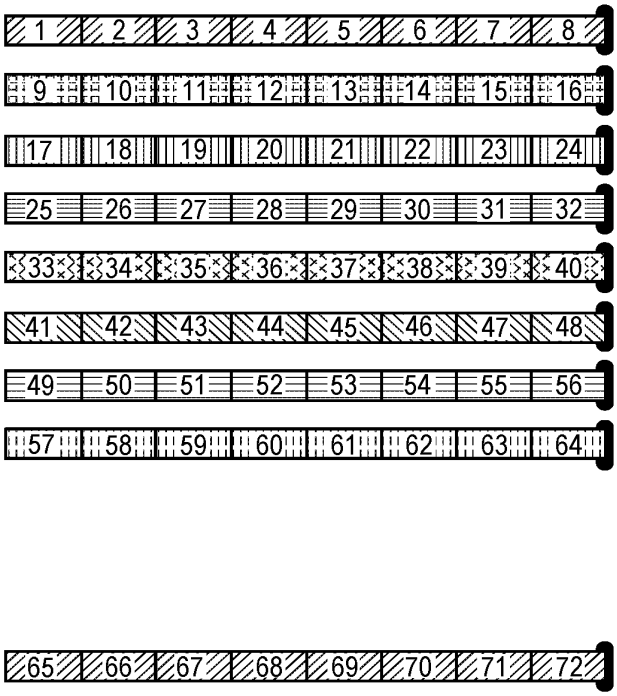

1 2 3 4 5 6 7 8
9 10 11 12 13 14 15 16
17 18 19 20 21 22 23 24
25 26 27 28 29 30 31 32
33 34 35 36 37 38 39 40
41 42 43 44 45 46 47 48
49 50 51 52 53 54 55 56
57 58 59 60 61 62 63 64
65 66 67 68 69 70 71 72

FIG. 14A

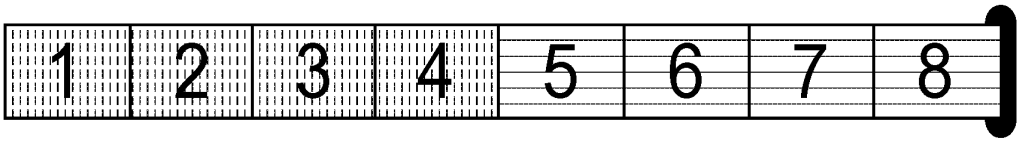


FIG. 14B



FIG. 14C

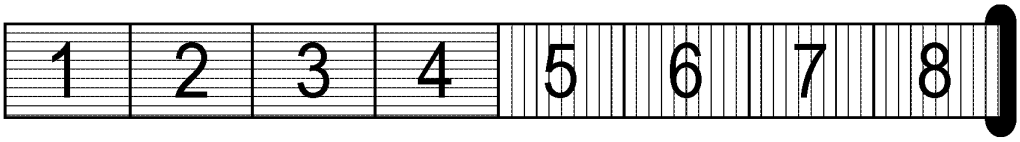


FIG. 14D

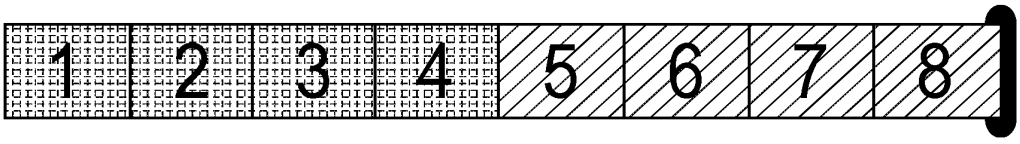


FIG. 14E

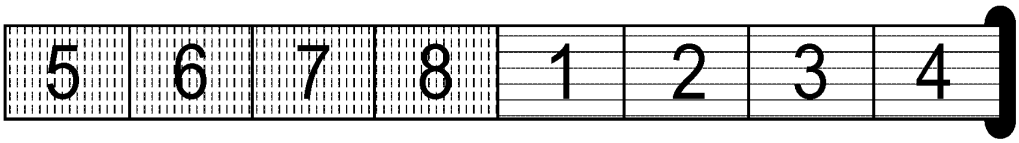


FIG. 14F

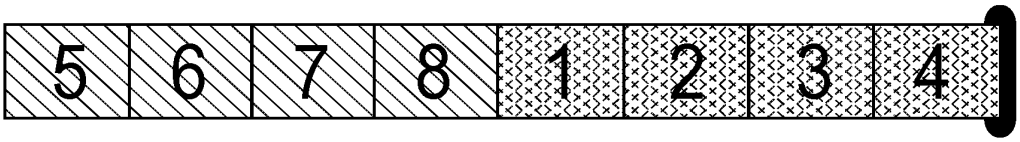


FIG. 14G
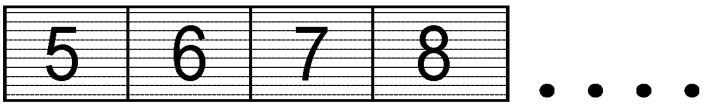


FIG. 15

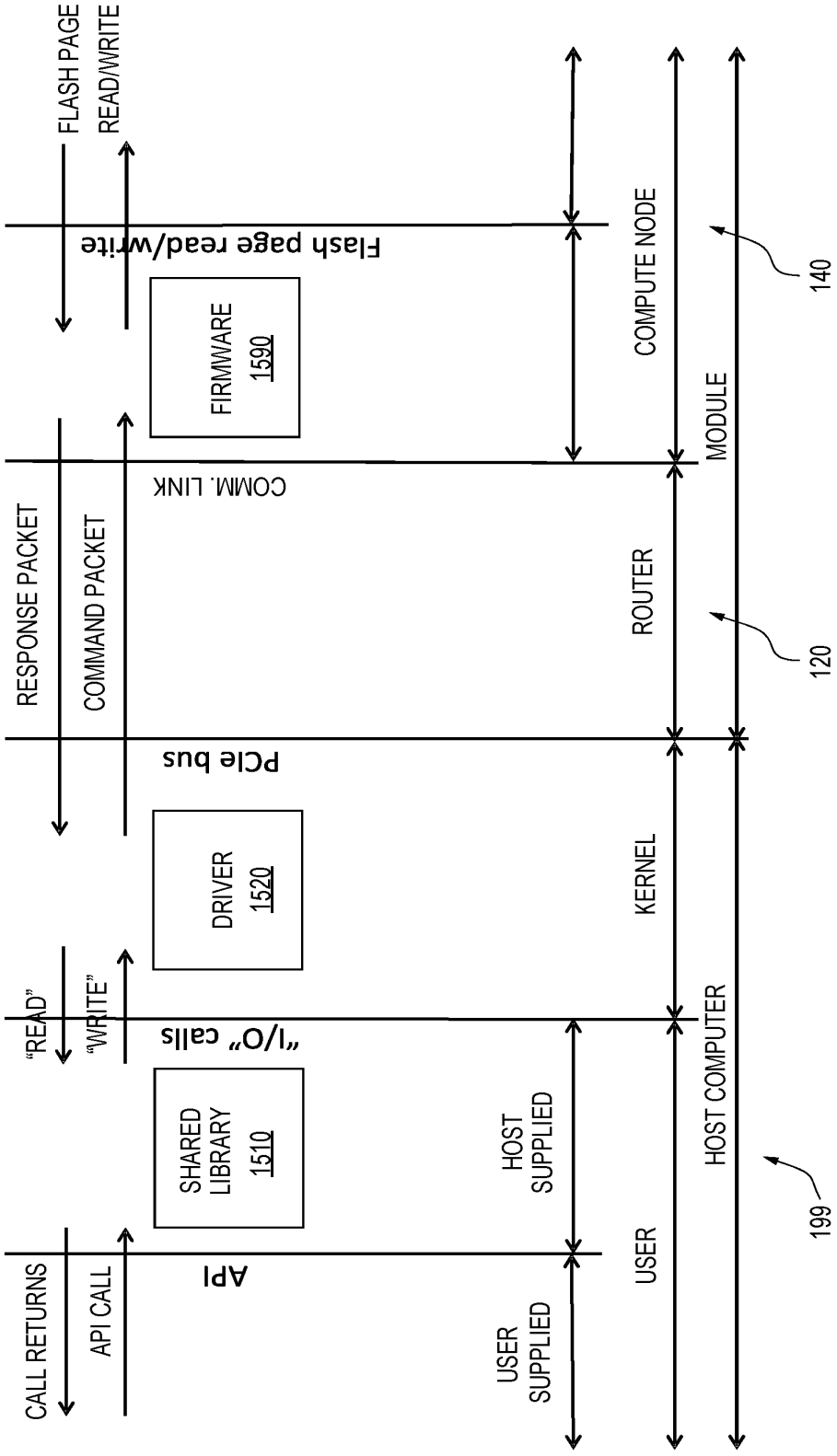

FLASH PAGE READ/WRITE
Flash page read/write
FIRMWARE 1590
COMPUTE NODE
140
COMM. LINK
MODULE
RESPONSE PACKET
COMMAND PACKET
ROUTER
120
PCIe bus
DRIVER 1520
KERNEL
"READ"
"WRITE"
"I/O" calls
SHARED LIBRARY 1510
HOST SUPPLIED
HOST COMPUTER
199
API
USER
CALL RETURNS
API CALL
USER SUPPLIED

FIG. 16

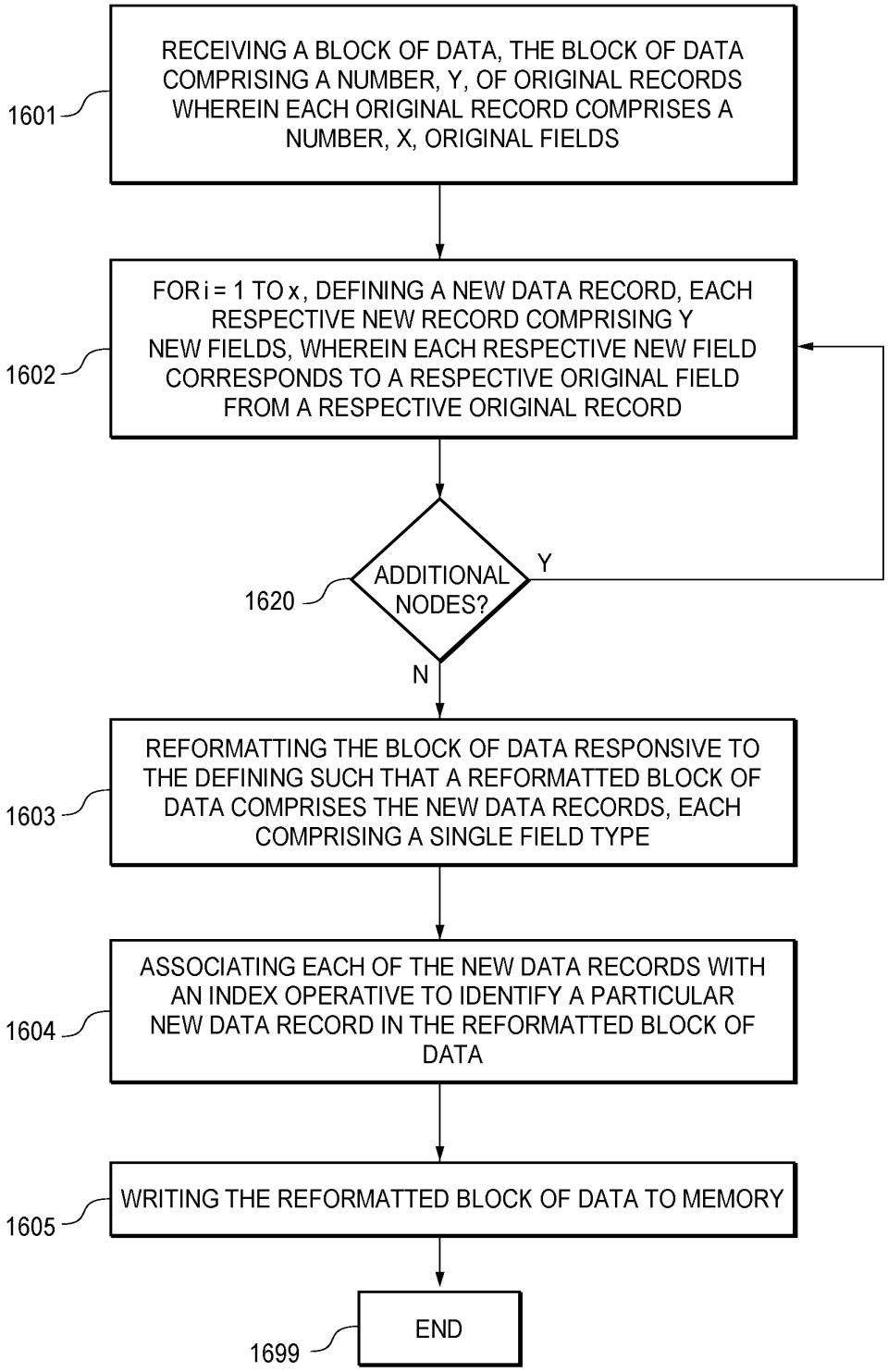

RECEIVING A BLOCK OF DATA, THE BLOCK OF DATA COMPRISING A NUMBER, Y, OF ORIGINAL RECORDS WHEREIN EACH ORIGINAL RECORD COMPRISES A NUMBER, X, ORIGINAL FIELDS
1601
FOR i = 1 TO x, DEFINING A NEW DATA RECORD, EACH RESPECTIVE NEW RECORD COMPRISING Y NEW FIELDS, WHEREIN EACH RESPECTIVE NEW FIELD CORRESPONDS TO A RESPECTIVE ORIGINAL FIELD FROM A RESPECTIVE ORIGINAL RECORD
1602
ADDITIONAL NODES?
1620
Y
N
REFORMATTING THE BLOCK OF DATA RESPONSIVE TO THE DEFINING SUCH THAT A REFORMATTED BLOCK OF DATA COMPRISES THE NEW DATA RECORDS, EACH COMPRISING A SINGLE FIELD TYPE
1603
ASSOCIATING EACH OF THE NEW DATA RECORDS WITH AN INDEX OPERATIVE TO IDENTIFY A PARTICULAR NEW DATA RECORD IN THE REFORMATTED BLOCK OF DATA
1604
WRITING THE REFORMATTED BLOCK OF DATA TO MEMORY
1605
END
1699

FIG. 17

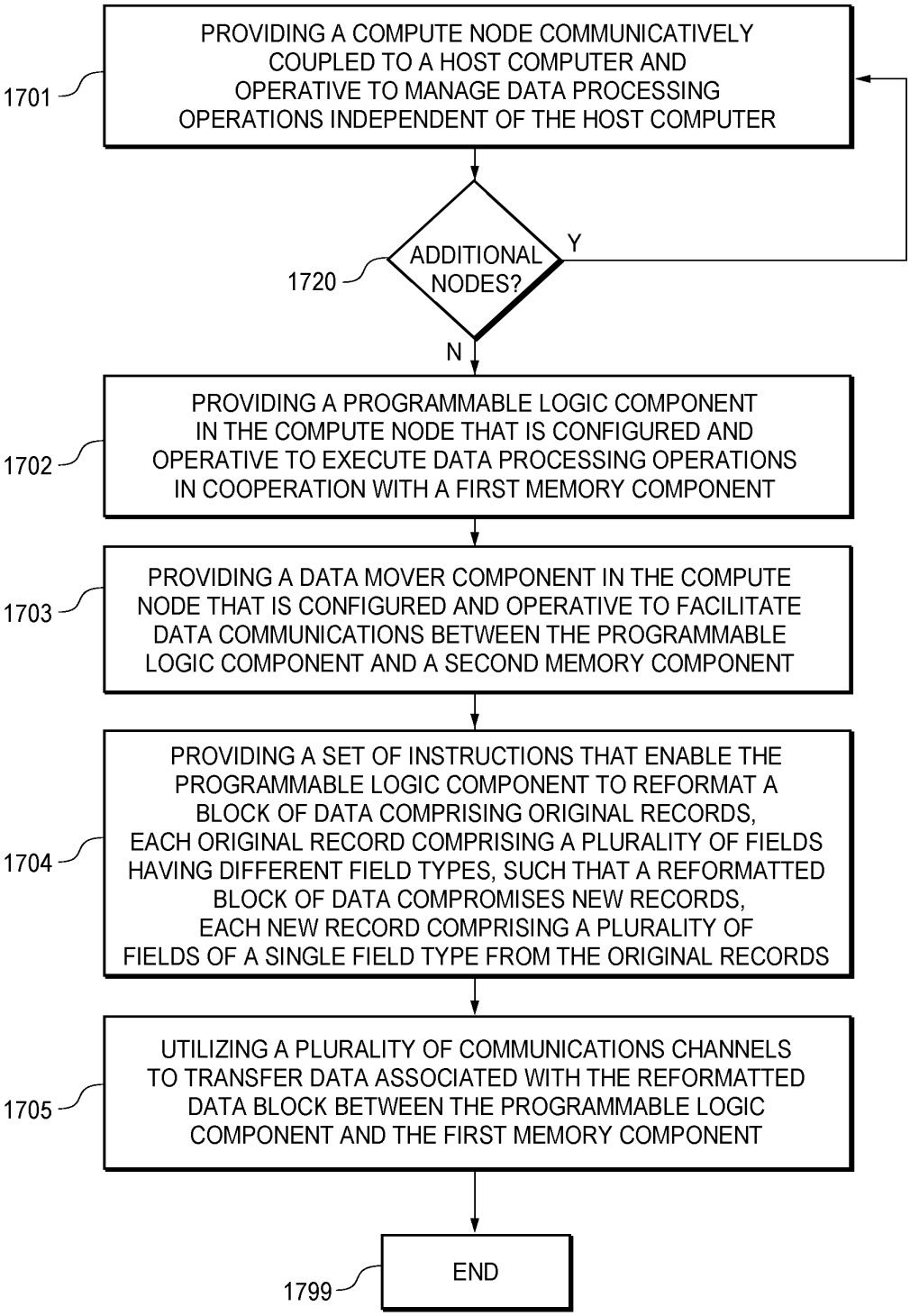
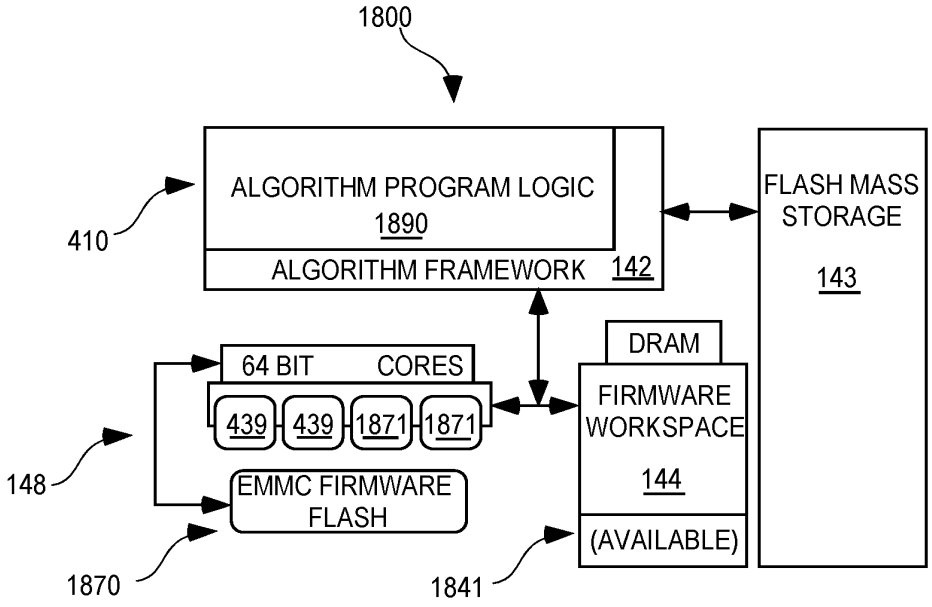
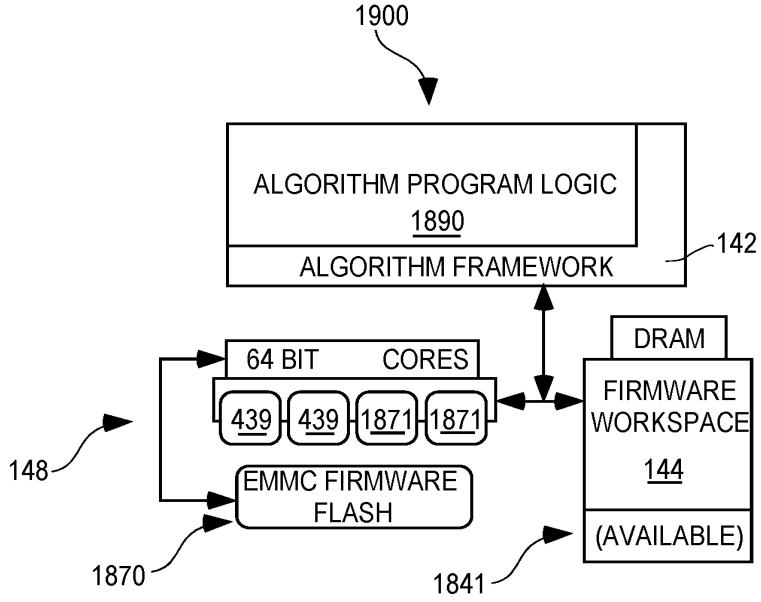

PROVIDING A COMPUTE NODE COMMUNICATIVELY COUPLED TO A HOST COMPUTER AND OPERATIVE TO MANAGE DATA PROCESSING OPERATIONS INDEPENDENT OF THE HOST COMPUTER
1701
ADDITIONAL NODES?
1720
Y
N
PROVIDING A PROGRAMMABLE LOGIC COMPONENT IN THE COMPUTE NODE THAT IS CONFIGURED AND OPERATIVE TO EXECUTE DATA PROCESSING OPERATIONS IN COOPERATION WITH A FIRST MEMORY COMPONENT
1702
PROVIDING A DATA MOVER COMPONENT IN THE COMPUTE NODE THAT IS CONFIGURED AND OPERATIVE TO FACILITATE DATA COMMUNICATIONS BETWEEN THE PROGRAMMABLE LOGIC COMPONENT AND A SECOND MEMORY COMPONENT
1703
PROVIDING A SET OF INSTRUCTIONS THAT ENABLE THE PROGRAMMABLE LOGIC COMPONENT TO REFORMAT A BLOCK OF DATA COMPRISING ORIGINAL RECORDS, EACH ORIGINAL RECORD COMPRISING A PLURALITY OF FIELDS HAVING DIFFERENT FIELD TYPES, SUCH THAT A REFORMATTED BLOCK OF DATA COMPROMISES NEW RECORDS, EACH NEW RECORD COMPRISING A PLURALITY OF FIELDS OF A SINGLE FIELD TYPE FROM THE ORIGINAL RECORDS
1704
UTILIZING A PLURALITY OF COMMUNICATIONS CHANNELS TO TRANSFER DATA ASSOCIATED WITH THE REFORMATTED DATA BLOCK BETWEEN THE PROGRAMMABLE LOGIC COMPONENT AND THE FIRST MEMORY COMPONENT
1705
END
1799

FIG. 18



FLASH/COMUTE NODE

FIG. 19



COMPUTE NODE

FIG. 20

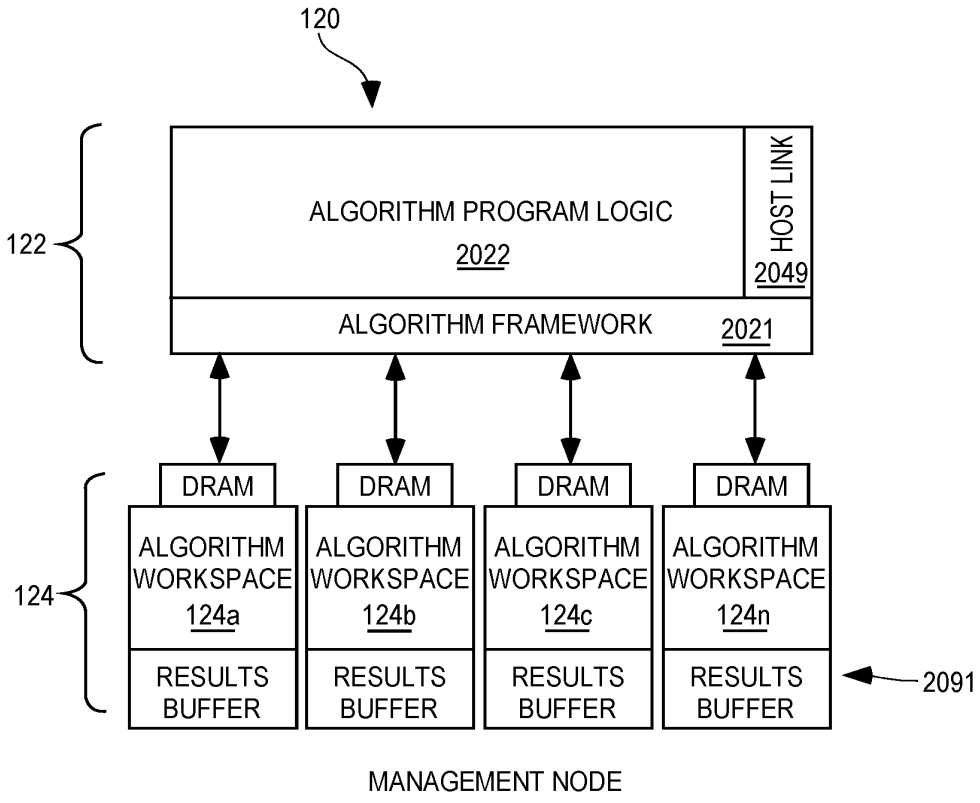

120
122
ALGORITHM PROGRAM LOGIC
2022
HOST LINK
2049
ALGORITHM FRAMEWORK
2021
124
DRAM
DRAM
DRAM
DRAM
ALGORITHM WORKSPACE
124a
ALGORITHM WORKSPACE
124b
ALGORITHM WORKSPACE
124c
ALGORITHM WORKSPACE
124n
RESULTS BUFFER
RESULTS BUFFER
RESULTS BUFFER
RESULTS BUFFER
2091
MANAGEMENT NODE

FIG. 21

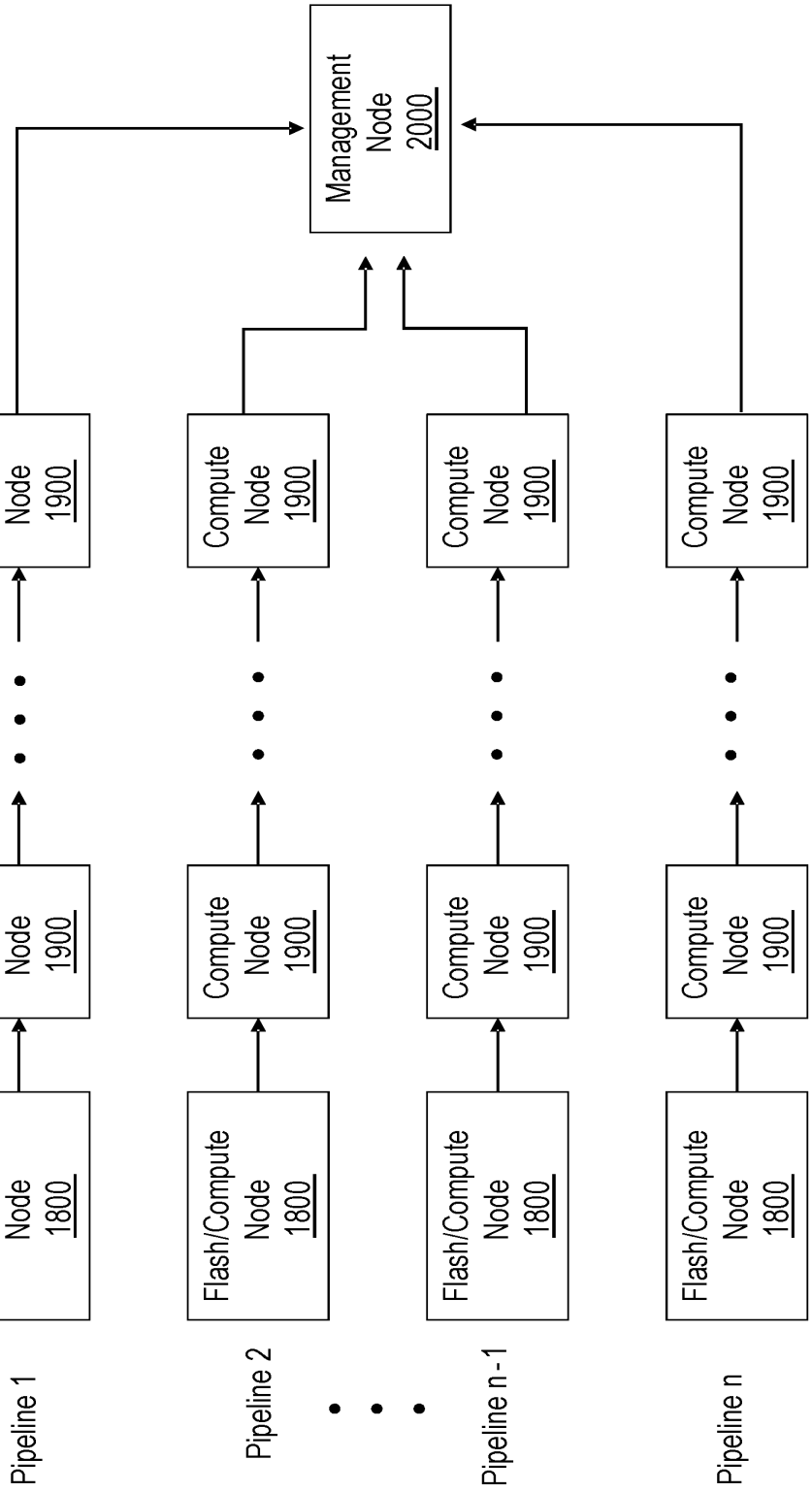
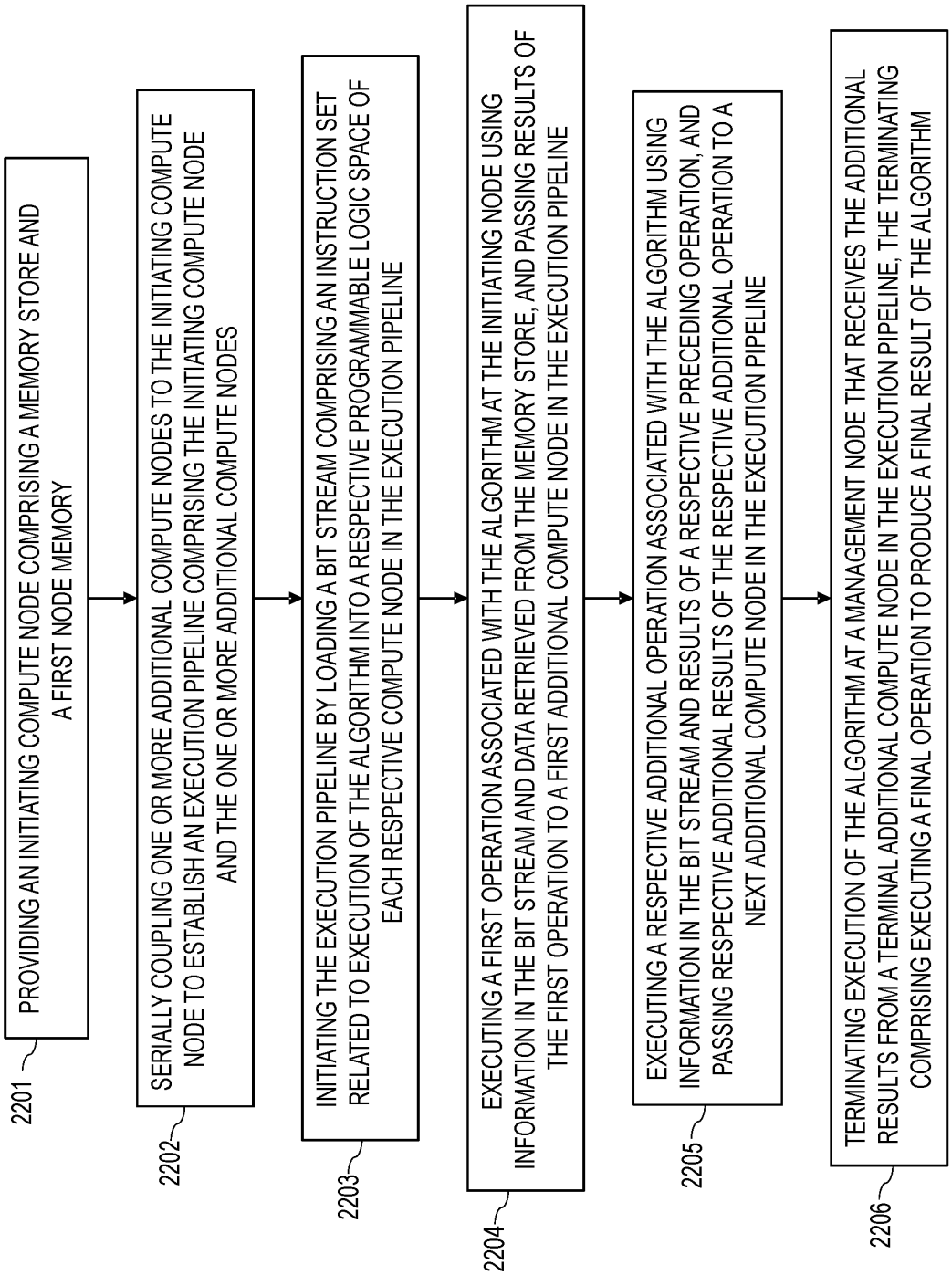

Management Node 2000
Tier t
Tier 2
Tier 1
Compute Node 1900
Flash/Compute Node 1800
Pipeline 1
Pipeline 2
Pipeline n-1
Pipeline n

FIG. 22


2201
PROVIDING AN INITIATING COMPUTE NODE COMPRISING A MEMORY STORE AND A FIRST NODE MEMORY
2202
SERIALLY COUPLING ONE OR MORE ADDITIONAL COMPUTE NODES TO THE INITIATING COMPUTE NODE TO ESTABLISH AN EXECUTION PIPELINE COMPRISING THE INITIATING COMPUTE NODE AND THE ONE OR MORE ADDITIONAL COMPUTE NODES
2203
INITIATING THE EXECUTION PIPELINE BY LOADING A BIT STREAM COMPRISING AN INSTRUCTION SET RELATED TO EXECUTION OF THE ALGORITHM INTO A RESPECTIVE PROGRAMMABLE LOGIC SPACE OF EACH RESPECTIVE COMPUTE NODE IN THE EXECUTION PIPELINE
2204
EXECUTING A FIRST OPERATION ASSOCIATED WITH THE ALGORITHM AT THE INITIATING NODE USING INFORMATION IN THE BIT STREAM AND DATA RETRIEVED FROM THE MEMORY STORE, AND PASSING RESULTS OF THE FIRST OPERATION TO A FIRST ADDITIONAL COMPUTE NODE IN THE EXECUTION PIPELINE
2205
EXECUTING A RESPECTIVE ADDITIONAL OPERATION ASSOCIATED WITH THE ALGORITHM USING INFORMATION IN THE BIT STREAM AND RESULTS OF A RESPECTIVE PRECEDING OPERATION, AND PASSING RESPECTIVE ADDITIONAL RESULTS OF THE RESPECTIVE ADDITIONAL OPERATION TO A NEXT ADDITIONAL COMPUTE NODE IN THE EXECUTION PIPELINE
2206
TERMINATING EXECUTION OF THE ALGORITHM AT A MANAGEMENT NODE THAT RECEIVES THE ADDITIONAL RESULTS FROM A TERMINAL ADDITIONAL COMPUTE NODE IN THE EXECUTION PIPELINE, THE TERMINATING COMPRISING EXECUTING A FINAL OPERATION TO PRODUCE A FINAL RESULT OF THE ALGORITHM

Electronic Patent Application Fee Transmittal	
Application Number:	
Filing Date:	
Title of Invention:	Analytics, Algorithm Architecture, and Data Processing System and Method
First Named Inventor/Applicant Name:	Robert Bismuth
Filer:	Victor Jay Castellucci/Maria Lima
Attorney Docket Number:	FERM0003CON2/00564319
Filed as Small Entity	
Filing Fees for Utility under 35 USC 111(a)	

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
UTILITY FILING FEE (ELECTRONIC FILING)	4011	1	80	80
UTILITY SEARCH FEE	2111	1	350	350
UTILITY EXAMINATION FEE	2311	1	400	400
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Post-Allowance-and-Post-Issuance:				
Extension-of-Time:				
Miscellaneous:				
		Total in USD ($)		**830**

Electronic Acknowledgement Receipt	
EFS ID:	47150063
Application Number:	18075576
International Application Number:	
Confirmation Number:	1074
Title of Invention:	Analytics, Algorithm Architecture, and Data Processing System and Method
First Named Inventor/Applicant Name:	Robert Bismuth
Customer Number:	27799
Filer:	Victor Jay Castellucci/Maria Lima
Filer Authorized By:	Victor Jay Castellucci
Attorney Docket Number:	FERM0003CON2/00564319
Receipt Date:	06-DEC-2022
Filing Date:	
Time Stamp:	11:47:35
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	$830
RAM confirmation Number	E2022B6B47531967
Deposit Account	
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1		1FERM0003CON2_564319_Specification_FINAL.pdf	667631 f99b6efb299fab26781a2b4698652b1a5fcc9677	yes	73

Multipart Description/PDF files in .zip description

Document Description	Start	End
Specification	1	70
Claims	71	72
Abstract	73	73

Warnings:

Information:

2	Transmittal of New Application	Transmittal_FERM0003CON2_FINAL.pdf	227527 00abf758a776a415a7a04d46508141f846475e87	no	2

Warnings:

Information:

3	Application Data Sheet	FERM0003CON_200564319_ADS_FINAL.pdf	107755 ba5d251e32650a606d0d223c0fb8c4d6628f932c	no	7

Warnings:

Information:

This is not an USPTO supplied ADS fillable form

4	Oath or Declaration filed	Oath_MIKE.pdf	132318 5614f37b3c1a49e5e9f6fb684edbe17a10a2d59f	no	1

Warnings:

Information:

5	Oath or Declaration filed	OATH_Robert.pdf	164113 120cee55db4ce99c9686c8063482d393c8d65107	no	1
Warnings:					
Information:					
6	Drawings-only black and white line drawings	FERM0003CON2_00564319DrawingsFINAL.pdf	609577 62c12f847f83b0419e3f32b2c8fb783bb1848e5d	no	29
Warnings:					
Information:					
7	Fee Worksheet (SB06)	fee-info.pdf	44442 2f3019f877d59f0e9bffa382fd12af2705366966	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	1953363		

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 a 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.

Analytics, Algorithm Architecture, and Data Processing System and Method

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation of, and claims priority to, co-pending United States patent application Serial No. 17/666,049, filed February 7, 2022, which is a continuation of United States patent application Serial No. 16/729,727, filed December 30, 2019, and also claims the benefit of the following United States provisional patent applications: Serial No. 62/801,839, filed February 6, 2019; Serial No. 62/853,764, filed May 29, 2019; and Serial No. 62/882,044, filed August 2, 2019. The disclosures of each of the foregoing applications are hereby incorporated herein by reference in their entireties.

FIELD OF THE DISCLOSURE

[0002] Aspects of the disclosed subject matter relate generally to high performance data processing, and more particularly to a system and method employing a distributed hardware architecture, either independently or in cooperation with an attendant data structure, in connection with various data processing strategies and data analytics implementations; additionally or alternatively, aspects of the disclosed subject matter relate to an unique algorithm architecture and processing system and method that may be implemented, either independently or in cooperation with an attendant data structure, in connection with various data processing strategies and data analytics implementations.

BACKGROUND

[0003] Recently, “Big Data,” high performance computing, and solid state device technologies have become increasingly important in many contexts, such as in connection with machine learning and artificial intelligence projects, for instance. In accordance with conventional computing strategies, solid state devices (such as Flash™ memory and other data storage hardware media), having no moving parts or motors and having relatively short access/read/write times as compared to traditional spinning media, have been gaining popularity in cycle-intensive applications that benefit from decreased latency in memory access. On the other hand, with the explosion of data available to such systems (as a result, for example, of the nascent Internet of Things (“IoT”), distributed memory systems, and other processing paradigms involving devices sharing data with other devices), the sheer volume of available data to process is increasing faster than traditional hardware and software systems are able to evolve in order to process those data in a meaningful and efficient manner.

[0004] Further, most conventional systems designed for high throughput data processing and analytics rely upon exhaustive (or "brute force") approaches that attempt to overpower the magnitude of the challenge with overwhelming computational resources, at the expense of cycle time and power consumption. In many instances, the dataset size creates a situation in which a practical or commercially reasonable solution is impossible, or at least prohibitively expense. As a practical matter, it will be appreciated that for as long as the rate at which new data become available for processing continues to outpace the rate at which processing methodologies advance to accommodate the increased size of a given dataset, it will continue to take longer to solve bigger and more complex data processing problems.

[0005] Therefore, there is a need for an improved system and method employing a distributed hardware architecture that facilitates high throughput data processing and analytics solutions for resource-intensive applications; as set forth below, some implementations of such a distributed hardware architecture may be configured and operative to employ a data structure that departs from traditional approaches. Further, there is a need for an improved algorithm architecture and processing system and method for high throughput data processing and analytics solutions for resource-intensive applications. As with the distributed hardware architecture noted above, such an algorithm architecture and processing system may have utility in connection with a non-traditional data structure.

SUMMARY OF THE DISCLOSURE

[0006] The following presents a simplified summary of the disclosure in order to provide a basic understanding of some aspects of various embodiments disclosed herein. This summary is not an extensive overview of the disclosure. It is intended neither to identify key or critical elements of the disclosed embodiments nor to delineate the scope of those embodiments. Its sole purpose is to present some concepts of the disclosed subject matter in a simplified form as a prelude to the more detailed description that is presented later.

[0007] The present disclosure describes a system and method employing a distributed hardware architecture to manage and to execute data processing operations, either independently or in cooperation with an attendant data structure; in the context of this description, a discrete component of such a distributed hardware architecture is generally referred to as a "compute node," though other terms may be used by those of skill in the art. In some implementations, one or more compute nodes may be utilized to manage and to execute data processing operations independent of, though communicatively coupled with, a host compute system. In addition, the present disclosure describes an algorithm architecture and processing

system and method that may be implemented, either independently or in cooperation with an attendant data structure, in connection with various data processing strategies and data analytics implementations that may be distributed across or amongst various compute nodes in a distributed processing environment or paradigm.

[0008] In accordance with one aspect of the disclosed subject matter, for example, a method of reformatting a data structure may generally comprise: receiving a block of data, the block of data comprising a number, y, of original records wherein each original record comprises a number, x, of original fields; for i = 1 to x, defining a new data record, each respective new record comprising y new fields, wherein each respective new field corresponds to a respective i^{th} original field from a respective original record; reformatting the block of data responsive to the defining such that a reformatted block of data comprises the new data records, each comprising a single field type; associating each of the new data records with an index operative to identify a particular new data record in the reformatted block of data; and writing the reformatted block of data to memory.

[0009] In accordance with some implementations, a reformatted data structure may be derived from an original data structure, and the reformatted data structure may generally comprise a plurality of new data records, each of the new data records comprising a plurality of new fields associated with a particular field type defined in the original data structure such that every new field in a particular new record is of the particular field type, wherein the reformatted data structure allows a processing system selectively to read the new data records as original records from the original data structure or as original fields from the original data structure.

[0010] In some disclosed implementations, a method of executing data processing operations may generally comprise: providing a compute node communicatively coupled to a host computer and operative to manage data processing operations independent of the host computer; providing a programmable logic component in the compute node that is configured and operative to execute data processing operations in cooperation with a first memory component; providing a data mover component in the compute node that is configured and operative to facilitate data communications between the programmable logic component and a second memory component; providing a set of instructions that enable the programmable logic component to reformat a block of data comprising original records, each original record comprising a plurality of fields having different field types, such that a reformatted block of data comprises new records, each new record comprising a plurality of fields of a single field type from the original records; and utilizing a plurality

of communications channels to transfer data associated with the reformatted data block between the programmable logic component and the first memory component.

[0011] In accordance with another aspect of the disclosed subject matter, a data processing system may be operative in cooperation with a host compute system, and the data processing system may generally comprise: a router module comprising a host interface and a node interface, wherein the router module is communicatively coupled to the host compute system via the host interface; and a compute node comprising: a communications link, wherein the compute node is communicatively coupled to the node interface at the router module via the communications link; a data store comprising records associated with a data processing operation; a programmable logic component to execute the data processing operation in cooperation with the data store; a node memory comprising data and instructions to support operation of the programmable logic component in connection with the data processing operation and to facilitate data communications via the communications link; a data mover component to facilitate data communications between the programmable logic component and the node memory; and a storage interface component to facilitate data communications between the programmable logic component and the data store; wherein the storage interface component utilizes a plurality of communications channels to transfer data between the programmable logic component and the data store.

[0012] Additionally or alternatively, a data processing system operative in cooperation with a host compute system may generally comprise: a router module comprising a host interface and a node interface, wherein the router module is communicatively coupled to the host compute system via the host interface; and a compute node comprising: a communications link, wherein the compute node is communicatively coupled to the node interface at the router module via the communications link; a data store comprising datasets associated with a data processing operation; a programmable logic component to execute the data processing operation in cooperation with the data store; a node memory comprising data and instructions to support operation of the programmable logic component in connection with the data processing operation and to facilitate data communications via the communications link; a data mover component to facilitate data communications between the programmable logic component and the node memory; and a storage interface component to facilitate data communications between the programmable logic component and the data store; wherein the programmable logic component executes a set of instructions to reformat a block of data comprising original records, each original record comprising a plurality of fields having different field types, such that a reformatted block of data comprises new records, each new record comprising a plurality of fields of a single field type

from the original records; and wherein the programmable logic component causes the new records to be stored as the datasets in the data store.

[0013] In accordance with another aspect of the disclosed subject matter, a data processing system operative in cooperation with a host compute system may be summarized as generally comprising: a management node comprising a host interface and a node interface, wherein the management node is communicatively coupled to the host compute system via the host interface; a memory-supported compute node comprising: (i) a communications link, wherein the memory-supported compute node is communicatively coupled to the node interface at the management node via the communications link; (ii) a data store comprising records associated with a data processing operation; (iii) a programmable logic component to execute the data processing operation in cooperation with the data store; and (iv) a node memory comprising data and instructions to support operation of the programmable logic component in connection with the data processing operation and to facilitate data communications via the communications link; and a pipeline of one or more additional compute nodes, each of the respective one or more additional compute nodes serially connected to the memory-supported compute node via a respective communications link and comprising a respective additional node memory and a respective programmable logic component to execute a respective data processing operation in cooperation with data received via the respective communications link.

[0014] In another aspect, a data processing system operative in cooperation with a host compute system may generally comprise: a management node comprising a host interface and a node interface, wherein the management node is communicatively coupled to the host compute system via the host interface; a memory-supported compute node comprising: (i) a communications link, wherein the memory-supported compute node is communicatively coupled to the node interface at the management node via the communications link; (ii) a data store comprising records associated with a data processing operation; (iii) a programmable logic component to execute the data processing operation in cooperation with the data store; and (iv) a node memory comprising data and instructions to support operation of the programmable logic component in connection with the data processing operation and to facilitate data communications via the communications link; and a pipeline of one or more additional compute nodes, each of the respective one or more additional compute nodes comprising: a respective communications link, wherein the respective additional compute node is communicatively coupled to an upstream compute node and a downstream compute node via the communications link; and a respective additional node memory and a respective programmable logic component, the respective

additional node memory comprising data and instructions to support operation of the respective programmable logic component in connection with a respective data processing operation and to facilitate data communications via the respective communications link; wherein each respective additional compute node executes the respective data processing operation using results from its respective upstream compute node, and passes respective results to its respective downstream compute node; in some implementations, it may be desirable that each respective additional compute node operates without storing the respective results in the respective additional node memory.

[0015] In accordance with yet another aspect of the disclosed subject matter, a method of executing an algorithm in a distributed processing compute environment may generally comprise: providing an initiating compute node comprising a memory store and a first node memory; serially coupling one or more additional compute nodes to the initiating compute node to establish an execution pipeline comprising the initiating compute node and the one or more additional compute nodes, each respective one of the one or more additional compute nodes comprising a respective second node memory; initiating the execution pipeline by loading a bit stream comprising an instruction set related to execution of the algorithm into a respective programmable logic space of each respective compute node in the execution pipeline; executing a first operation associated with the algorithm at the initiating node using information in the bit stream and data retrieved from the memory store, and passing results of the first operation to a first additional compute node in the execution pipeline; wherein the first additional compute node and each subsequent additional compute node in the execution pipeline executes a respective additional operation associated with the algorithm using information in the bit stream and results of a respective preceding operation, and passes respective additional results of the respective additional operation to a next additional compute node in the execution pipeline; and terminating execution of the algorithm at a management node that receives the additional results from a terminal additional compute node in the execution pipeline, the terminating comprising executing a final operation to produce a final result of the algorithm. As with the previous example, it may be desirable in some circumstances that each respective additional compute node operates without storing the respective results in the respective additional node memory.

[0016] The foregoing and other aspects of various disclosed embodiments will be apparent through examination of the following detailed description thereof in conjunction with the accompanying drawing figures, in which like reference numerals are used to represent like components throughout, unless otherwise noted.

DESCRIPTION OF THE DRAWING FIGURES

[0017] FIG. 1 is a functional block diagram illustrating one implementation of a data processing and analytics system for use in connection with a host compute system;

[0018] FIGS. 2A through 2C are functional block diagrams illustrating one example of a hardware environment accommodating components of a data processing and analytics system;

[0019] FIGS. 3A through 3C are functional block diagrams illustrating certain hardware components of an implementation of a data processing and analytics system such as that depicted in FIG. 1;

[0020] FIG. 4 is a functional block diagram illustrating components of an implementation of a compute node having utility in connection with a data processing and analytics system such as that depicted in FIG. 1;

[0021] FIG. 5 is a diagram illustrating one example of a data routing strategy to route data through portions of the compute node depicted in FIG. 4;

[0022] FIG. 6 is a diagram illustrating another example of a data routing strategy to route data through portions of the compute node depicted in FIG. 4;

[0023] FIG. 7 is a diagram illustrating another example of a data routing strategy to route data through portions of the compute node depicted in FIG. 4;

[0024] FIG. 8 is a diagram illustrating one implementation of a memory architecture having utility in connection with a compute node such as that depicted in FIG. 4;

[0025] FIG. 9 is a diagram illustrating one implementation of a portion of the memory architecture depicted in FIG. 8;

[0026] FIG. 10 is a diagram illustrating a conventional data record structure;

[0027] FIG. 11 is a diagram illustrating one implementation of a data record structure having utility in connection with a compute node operative in accordance with the present disclosure;

[0028] FIG. 12 is a diagram illustrating another implementation of a data record structure having utility in connection with a compute node operative in accordance with the present disclosure;

[0029] FIGS. 13 and 13A through 13H are diagrams illustrating another implementation of a data record structure having utility in connection with a compute node operative in accordance with the present disclosure;

[0030] FIGS. 14A through 14G are diagrams illustrating yet another implementation of a data record structure having utility in connection with a compute node operative in accordance with the present disclosure;

[0031] FIG. 15 is a diagram illustrating one strategy for data communications between a host compute system and a data processing and analytics system as depicted in FIG. 1;

[0032] FIG. 16 is a functional flow diagram illustrating aspects of one implementation of a method of reformatting a data structure; and

[0033] FIG. 17 is a functional flow diagram illustrating aspects of one implementation of a method of executing data processing operations.

[0034] FIGS. 18 and 19 are functional block diagrams illustrating components of various implementations of a compute node having utility in connection with a data processing and analytics system;

[0035] FIG. 20 is a functional block diagram illustrating components of an implementation of a management node having utility in connection with a data processing and analytics system;

[0036] FIG. 21 is a functional block diagram illustrating one implementation of an algorithm processing system using a management node; and

[0037] FIG. 22 is a functional flow diagram illustrating aspects of one implementation of a method of executing data processing operations.

DETAILED DESCRIPTION

[0038] Certain aspects and features of the disclosed subject matter may be further understood with reference to the following description and the appended drawing figures. In operation, a system and method employing a distributed hardware architecture, and optionally, an attendant data structure, may have utility in connection with various data processing strategies and data analytics implementations. Specifically, the present disclosure provides for an integrated data processing platform that may generally comprise hardware and software.

[0039] As set forth in more detail below, the present disclosure addresses design and implementation of an architectural framework that may employ high performance data processing elements (compute nodes) coupled with massive data storage (a data store) in a unique configuration that minimizes or eliminates typical processor/network wait states, optimizes instruction fetch memory cycles, and analyzes data that are streamed from an attached or associated data store at the maximum rate at which the data can be accessed or streamed by

the storage subsystem. The architectural framework generally contemplates one or multiple compute nodes operating in parallel (and in series, in some cases, as described below), each of which may be configured as a pipeline of computational elements that can be configured and operative to implement a variety of algorithms or other data processing operations. In the configurations provided as examples, a fully parallelized dataflow environment, which exhibits extremely high performance for analyzing massive amounts of data, is described.

[0040] The disclosed architectural arrangement may generally include a set of programmable logic blocks that may be linked in various ways to create (as well as to execute) arbitrary algorithms. This strategy greatly reduces the learning curve for using programmable processors to implement computational algorithms. For example, a typical application programmer (*e.g.*, with a limited understanding of programmable logic hardware, *per se*) may nevertheless easily use the disclosed architecture without needing to understand the intricacies of using one of more programmable logic block hardware components and how their structures or interconnections might affect or otherwise relate to algorithm requirements of the application program.

[0041] Additionally, embodiments are disclosed of a system and method of morphing (*i.e.*, reorganizing or reformatting) data located in data records; as is generally known in the art, these data are typically stored or maintained in files or datasets (or "records" or "pages"). The disclosed system and method may be implemented for the purpose of increasing the performance of analytic algorithms being applied to the data, and may optimize or maximize the rate at which data may be correctly presented to an analytic algorithm.

[0042] It will be appreciated that the present disclosure addresses design and implementation of memory controller (such as a Flash controller) hardware and strategies capable of supporting optimized, overlapping data read and write requests from or to a memory component, while also allowing streaming of data from the same memory component for analysis by an algorithm or other data processing operation. In one implementation, scheduling of read access requests issued to memory component targets may be sequenced and overlapped in a strategic manner such that data output from the memory component is maximized, optimized, or otherwise increased as compared to conventional memory controller technologies; specifically, the disclosed systems and methods may decrease or eliminate wait times typically caused by Flash (or other memory) read delays or latency.

[0043] In accordance with the disclosed subject matter, it will be appreciated that a power management system may readily be implemented using the architecture set forth below (see, *e.g.*, FIG. 4) to implement a system of microcontroller- or microprocessor-based regulators that

communicate with each other using an on-board local area network. This solution may allow power regulators to be sequenced as desired or required (for instance, by power consuming subsystems on a host hardware module), and also may allow the regulators to balance current load requirements. In some instances, this implementation may also provide a failover capability in the event that any single regulator or group of regulators fails. The result is a safe power environment that correctly starts up a module (such as a compute node or an entire interface card or other hardware component comprising or accommodating a plurality of compute nodes such as described below) without risk of damage due to uncoordinated power sequencing and also provides for a balanced, fault-tolerant running environment to facilitate or to improve reliability with respect to a particular hardware module operative in accordance with the disclosed subject matter. For instance, core voltage supplied to a processing unit in a router component as described below may be supplied via, or in cooperation with, a poly-phase, shared (*i.e.*, redundant) regulator to handle power sequencing, load balancing, and fault-tolerance; it will be appreciated that other voltage supply subsystems necessary or desirable to provide the following functionality may benefit from a similar or analogous voltage regulation strategy.

[0044] Turning now to the drawing figures, FIG. 1 is a functional block diagram illustrating one implementation of a data processing and analytics system for use in connection with a host compute system. As illustrated, one implementation of system 100 may generally comprise a host compute system 199, a router module 120, and a compute node 140.

[0045] Host compute system 199 may generally be embodied in or comprise a computer server, a desktop or workstation computer, a laptop or portable computer or tablet, or a combination of one or more of such components. In operation, host compute system 199 may be employed to initiate, instantiate, or otherwise to request data processing operations to be executed by system 100 substantially as set forth below. In that regard, host compute system 199 may include one or more microprocessors, field programmable gate arrays (FPGAs), microcontrollers, or other digital processing apparatus, along with attendant memory, controllers, firmware, network interface hardware, and the like. For example, host compute system 199 may generally comprise multiprocessor systems, microprocessor-based or programmable consumer electronics, personal computers ("PCs"), networked PCs, minicomputers, mainframe computers, and similar or comparable apparatus for general purpose or application-specific data processing. Various implementations of host compute system 199 may be deployed in distributed computing environments in accordance with which tasks or program modules may be performed or executed by remote processing devices, which may be linked through a communications network. Those

of skill in the art will appreciate that any of various computer servers, work stations, or other processing hardware components or systems of components may be suitable for implementation at host compute system 199, and that the disclosed subject matter is not limited to any particular hardware implementation or system architecture employed at host compute system 199.

[0046] With respect to requesting, initiating, enabling, facilitating, and receiving results of the data processing functionality described herein, host compute system 199 may be communicatively coupled to router module 120, enabling bi-directional data transmission in accordance with any of various communications interfaces or telecommunications protocols generally known in the art or developed and operative in accordance with known principles. Router module 120, in turn, may also be communicatively coupled to compute node 140 such that data may be transmitted or transferred, via router module 120, between host compute system 199 and compute node 140 substantially as set forth below.

[0047] Specifically, router module 120 may generally comprise a router 122, one or more router memory components 124a, 124b, 124c, … 124n ("router memory 124") and a host interface 126; additionally, router module 120 may also include one or more node interfaces (not illustrated in FIG. 1) such as network interface cards, wireless or inductive hardware components, or communications ports, jacks, or plugs, for instance, to enable router 122 to communicate with a respective compute node 140 as is generally known in the art. In use, router 122 may be operable to transmit data packets between host compute system 199 and a respective compute node 140 in cooperation with routing information (destination address, packet header data, flood domains, *etc*.) and other short term data (look up tables, resource availability, performance or load balancing data, *etc*.) stored in or accessible by router memory 124. In some instances, router 122 may distribute data packets and write commands to compute nodes 140 in accordance with a data-aware round robin strategy. Router memory 124 may be embodied in or comprise, by way of example, volatile memory such as random access memory (RAM) in any of its various forms, for instance, static RAM (SRAM), dynamic RAM (DRAM), double-data rate (DDR) RAM, and the like; in some applications, DDR4 RAM may be used as router memory 124. Attendant bus structures and memory controller elements are omitted from FIG. 1 for clarity, but are well-known in the art. In one implementation, router 122 may execute all routing functionality on or in cooperation with a Kintex™ KU095 FPGA platform marketed by Xilinx, Inc., but as many types of hardware implementations are generally available, the structure and operation of router 122 and router memory 124 are not described further.

[0048] In some implementations, host interface 126 may be operative in accordance with Peripheral Component Interconnect (PCI) or PCI Express (PCIe) hardware and associated communications protocols. Alternatively, other types of communications interface hardware and methodologies may be employed, such as Ethernet™ or Serial ATA (SATA). The specific hardware and communication strategy embodied by or incorporated into host interface 126 may be selected in accordance with bandwidth requirements, cooperating hardware at host compute system 199, functionality and operational characteristics of router 122, or a combination of these and a variety of other factors. The same is generally true of the hardware and communication protocol implemented at node interfaces, which may be configured and operative to communicate data packets bi-directionally between router 122 and compute nodes 140 as set forth below.

[0049] In that regard, router 122 may be communicatively coupled (via a respective node interface) to a respective compute node 140 via a communications link 149 integrated into each respective compute node 140. In the FIG. 1 illustration, only a single compute node 140 is illustrated, but an additional three in the first compute node tier are also depicted abstractly by the bi-directional arrows from router 122. It will be appreciated that each respective series of compute nodes 140 may employ a dedicated communications channel or "lane" for data transmission with router 122, such that the FIG. 1 architecture contemplates four parallel lanes, one for each of a series of compute nodes 140. In that regard, compute node 140 may be communicatively coupled, via communications link 149, to one or more additional compute nodes 140 connected in series (*i.e.*, at different successive tiers), as illustrated at the top right side of FIG. 1.

[0050] In addition to communications link 149, in the FIG. 1 arrangement, each compute node 140 may generally comprise a compute array 142, a data store 143, a data store interface 145, node memory 144, one or more datamover apparatus (DMA) 146a and 146b ("DMAs 146"), and a processing system 148. These elements are described in detail below with specific reference to FIG. 4.

[0051] In some implementations, data store 143 may be embodied in or comprise a mass data storage component, such as a non-volatile data storage device, one example of which is an Electronically Erasable Programmable Read Only Memory (EEPROM) store. For example, data store 143 may be, or include, Flash memory, though other memory types having suitable or appropriate characteristics to facilitate the functionality set forth herein may be in use currently or developed in the future. Specifically, though all Flash is EEPROM (as a technical matter), it is noted that not all EEPROM is Flash. Typically, those of skill in the art may tend to interpret the term EEPROM as a very specific type of non-volatile memory, but the term is not so limited as

employed in the context of this disclosure; examples using Flash are described below, but any non-volatile storage media and high-bandwidth interface protocol may have utility in the context of the disclosed subject matter. In situations in which data store 143 is implemented as Flash, data store interface 145 may employ an Open NAND Flash Interface (ONFI) protocol; in the case that data store 143 uses a different memory technology, data store interface 145 may be selected or modified accordingly. In operation, data store interface 145 manages access (*e.g.*, read and write access) to data store 143, as is generally known in the art.

[0052] As with router memory 124, node memory 144 may be implemented as or comprise DDR4 RAM, or other volatile or non-volatile memory suitable to support operations of compute node 140, in general, and processing system 148, in particular. In the illustrated architecture, processing system 148 may generally handle read/write commands and other memory access with respect to data store 143 (via data store interface 145), receive and act upon interrupt requests (*e.g.*, with respect to operations executing at compute array 142), and regulate, control, or otherwise influence data transfers managed by DMAs 146. For instance, in one embodiment, processing system 148 may comprise a set of one or more processing components to manage all interrupt handling as well as Advanced Extensible Interface (AXI) or other data transfers via DMAs 146, on the one hand, and a different set of one or more processing components to manage general system administration (or "house-keeping") and management tasks for compute node 140 as well as algorithm execution support for compute array 142, on the other hand. While this implementation is described in more detail below with reference to FIG. 4, it is worth noting here that it is only one example, and that many different component arrangements and task distributions are within the scope and contemplation of the disclosed subject matter.

[0053] It is noted that router module 120 (including host interface 126) and the various compute nodes 140 described above, as well as associated busses, communications pathways, and attendant or supporting hardware infrastructure, may be incorporated into or disposed on a carrier board (such as a printed circuit board (PCB)) or interface card ("card," see reference numeral 200 in FIGS. 2A and 2B) that is configured and operative to cooperate with structural components of host compute system 199 as is generally known. Specifically, one or more cards 200 may be inserted into or otherwise engage receptacles, slots, jacks, plugs, *etc.* that are electrically and communicatively coupled to a motherboard, bus, or other interface component at host compute system 199; accordingly, system 100 may be scalable as a function of slots that are available at host compute system 199 to receive a card 200.

[0054] At a high level, the disclosed architecture of system 100 generally includes a star arrangement in accordance with which a central host interface 126 (*e.g.*, a PCIe interface) to/from host compute system 199 may be routed to four (or more) "riser" positions via router 122 (*e.g.*, an FPGA or microprocessor platform). At each riser position, a stack (*i.e.*, various tiers in FIG. 1) of compute nodes 140 may be communicatively coupled in series. One configuration may generally comprise two rows of four compute nodes 140 (tiers), whereas the FIG. 1 configuration includes four rows, each having an indeterminate number of tiers. In accordance with a PCIe implementation, each riser position or stack may employ eight discrete lanes of 8 Gb/s data links, *i.e.*, a total of 64 Gb/s, full duplex, capacity per riser position.

[0055] As noted above, communications link 149 may be employed to couple compute nodes 140 within a stack (*i.e.*, at different successive tiers) in series. In some implementations, communications link 149 may be based upon and operative in accordance with standard serial interfaces or protocols, though some proprietary serial links may also be appropriate, depending upon the functionality and operational characteristics of compute array 142, for instance. For instance, one example of compute array 142 may employ or comprise a Xilinx Zynq™ Ultrascale+™ ZU9EG FPGA, in which case, it may be desirable that communications link 149 employ or comprise a Xilinx Aurora™ communication link, as it functions well with transceivers provided by the same manufacturer. In operation, communications link 149 may be selected, configured, and operative to provide a light-weight wrapper of the complex multi-gigabit transceivers deployed in the FPGA fabric of compute array 142, and so it may be desirable to select communications link 149 hardware as a function of, or otherwise taking into consideration, the hardware and operation of compute array 142. Those of skill in the art will appreciate that many options exist for both compute array 142 and communications link 149, and that specific commercial embodiments have been identified by way of example only, and not by way of limitation.

[0056] In connection with Big Data and other resource-intensive processing applications, it may generally be desirable to allow compute nodes 140 autonomously to handle much of the data ingest activities and distribution of processing responsibilities; accordingly, compute nodes 140 themselves may, in some cases, determine an appropriate method of round robin processing and tracking of write histories during complex data processing operations. Additionally or alternatively, it may be desirable that host compute system 199 provide directions or instructions regarding round robin processing and write operations.

[0057] For example, each compute node 140 may monitor and track, or log, its own local activity; as the level of detail or the quantity of activity to be tracked increases, however, the amount of node memory 144 required (to enable rapid access, for example) may generally increase, the latency for a given memory access operation (or for a given type or class of operations) will increase, or both. This is especially true if information or local activity sought to be tracked must be extracted from data store 143 prior to each new write transaction.

[0058] Further, communication with neighboring compute nodes 140 in the disclosed star architecture may generally become increasingly costly the further away a particular compute node 140 is from the neighboring compute node 140. Elapsed time, throughput degradation (if control or administrative paths are shared with payload data), programmable logic resource consumption, and energy consumption may all increase in proportion with or as a function of distance between respective compute nodes 140 that are to exchange information. One way to eliminate or to minimize such deleterious effects is to implement a mesh architecture (not shown), in accordance with which an "any to any" communications strategy may be integrated into the architecture of system 100. In such an embodiment, a bit of sideband control traffic may generally have minimal or negligible impact, and may be implemented effectively without having a materially adverse effect on payload data or overall performance of system 100.

[0059] One approach to round robin processing that may readily be implemented across multiple cards 200 is to allow host compute system 199 itself to direct write commands to specific data targets or memory addresses, down to the data store 143 level, for instance. In particular, host compute system 199 may generally be apprised of all the data (*i.e.*, system-wide) to be written, and is unique in that regard (as compared to other components of system 100); accordingly, host compute system 199 may be enabled to select, compute, or otherwise to determine a proper, appropriate, or optimal granularity of distribution (*e.g.*, after how many clock cycles may the round robin advance from one compute node 140 to another). Furthermore, host compute system 199 may keep track of last writes across the entirety of the topography of system 100, and then re-start the round robin at an appropriate point (*e.g.*, after a power cycle, crash, or other interruption).

[0060] Employing host compute system 199 in the foregoing manner may not scale well in terms of performance under certain circumstances, particularly in situations where host compute system 199 is limited to local resources and not distributed across multiple processing platforms. For applications or environments requiring high scalability, an alternative approach may include enabling host compute system 199 to inform each attached or coupled card 200 of an optimal

granularity of the round robin method that it is to utilize (for example, to attain maximum or some desired level of efficiency). In the case of a system employing multiple cards 200, host compute system 199 itself may implement the first tier of round robin distribution. In this example, each compute node 140 may keep track of the last channel as well as the memory address or data record (per channel) that was used for each record or column of data that is written as set forth in more detail below.

[0061] If the first stage of the round robin mechanism (*e.g.*, on the FPGA implemented at router 122) has no non-volatile storage available, it may not be possible to store the last target of distribution across power cycles. To avoid overburdening the first compute node 140 as a target in the case of many power cycles (each of which may require round robin re-initialization), the round robin logic may be initialized with a random first target in some applications. If the round robin granularity is too small and the data are distributed across multiple compute nodes 140 without filling a full page on any of them, the performance of system 100 incurs partial page write penalties on all compute nodes 140 involved.

[0062] FIGS. 2A through 2C are functional block diagrams illustrating one example of a hardware environment accommodating components of a data processing and analytics system.

[0063] FIG. 2A depicts a top view of a carrier board (such as a PCB) or interface card ("card" 200) that may support or carry router module 120 and compute node 140 substantially as set forth above. Card 200 may comprise a typical or conventional form factor host interface 126 suitably dimensioned and configured to create electrical and/or power connections when engaged with a cooperating structure integrated into, associated with, or otherwise communicatively coupled to host compute system 199. As noted above, host interface 126 may be a PCIe, Ethernet, SATA, or other interface, and may be coupled to electrical and power connections on or integrated with card 200 via conventional PCB fabrication techniques. FIG. 2A also illustrates communications links 149a, 149b, 149c, through 149n (links 149), a respective one dedicated to each of a respective number "n" of stacks of compute nodes 140 coupled in series as set forth above. It is noted that card 200 may employ more or fewer than the four communications links 149 that are depicted in FIG. 2A.

[0064] FIG. 2B depicts a top view of a card 200 on which is disposed one example of a managing interface 202 component, and FIG. 2C depicts a top view of the managing interface 202 component of FIG. 2B. In some implementations, managing interface 202 may be embodied in or comprise router module 120 described above; in that regard, managing interface 202 may be an example of one embodiment of router module 120, such that router 122 maps to an FPGA

299 and router memory 124 components map to interface memory components 204a and 204b, each of which is depicted as having a number of discrete memory cells or memory modules 214, such as dual in-line memory modules (DIMMs) or other independently addressable memory structures. It is noted that, in this example, managing interface 202 may further comprise additional interface memory components (not shown) to correspond to the number of router memory 124 components employed in router module 120. As an alternative, router module 120 may be deployed in the FPGA 299 block in FIG. 2, and interface memory components 204a and 204b may support the functionality of router module 120 substantially as described above. Those of skill in the art will appreciate that other alternatives exist, and that the disclosed subject matter is not intended to be limited to the architecture or structural arrangement illustrated in FIGS. 2B and 2C.

[0065] FIGS. 3A through 3C are functional block diagrams illustrating certain hardware components of an implementation of a data processing and analytics system such as that depicted in FIG. 1. The left side of FIG. 3A depicts communications links 149a, 149b, 149c, through 149n (links 149) and a power coupling 399 in isolation, while the right side of FIG. 3A depicts links 149 and power coupling 399 as they may be deployed on card 200. As noted above, each respective communications link 149 may be dedicated to each of a respective number “n” of stacks of compute nodes 140 coupled in series, and card 200 may employ more or fewer than the four communications links 149 that are depicted in FIG. 3A.

[0066] FIG. 3B is a first view of one side of a card or substrate upon which elements of compute nodes 340a, 340b, 340c, through 340n (compute nodes 340) may be disposed, and FIG. 3C is a second view of an opposite side of a card or substrate upon which other elements of compute nodes 340 may be disposed. The left side of each of FIGS. 3B and 3C illustrates a single compute node 340n in isolation, while the right side illustrates a number, “n,” of such compute nodes implemented on card 200.

[0067] It is noted that compute nodes 340 may be integrated with, or disposed upon, a PCB or other carrier board, interface card, or similar structural substrate such as described above with reference to card 200. In the example of FIGS. 3B and 3C, for instance, some elements or components of compute nodes 340 may be disposed upon, attached to, or integrated with one side of such a card, whereas other components of compute nodes 340 may be disposed upon, attached to, or integrated with an opposing side. This particular structural arrangement, and appropriate manufacturing methods for effectuating same, are generally known in the art of

electronics fabrication, and the disclosed subject matter is not intended to be limited to any particular fabrication techniques or process node methodologies.

[0068] Each compute node 340 may generally comprise communications link 149, a data store (depicted abstractly at reference numeral 343 in FIG. 3B), a compute array 342, and node memory 344.In some implementations, compute nodes 340 may map generally to compute nodes 140 illustrated in, and described with reference to, FIG. 1. In one example, data store 343, compute array 342, and node memory 344 may correspond, respectively, to data store 143, compute array 142, and node memory 144 described above; in such an example, compute nodes 340 may additionally comprise components that are equivalent or analogous to data store interface 145, DMAs 146, and processing system 148, though these are omitted from FIGS. 3B and 3C for clarity. In another embodiment using a different mapping, compute array 342 itself may incorporate some or all of the components illustrated and described with reference to compute node 140, in which case data store 343 and node memory 344 may be employed to support the functionality of compute array 342.

[0069] In the examples described above, system 100 may facilitate scalability, in terms of both memory or storage density as well as compute power, that is greater than conventional systems. Additionally, the foregoing modular approach may accommodate design longevity and flexibility with respect to module design and implementation (or "bring up"). In particular, data store 143/343 may be closely coupled to compute resources (such as compute array 142/342), while remaining physically separated, such as by deployment on opposing sides of an interface card 200 or PCB, which facilitates cooling and minimizes thermal effects that are typically associated with implementing mass storage in proximity to high power processing resources or system on chip (SoC) hardware.

[0070] FIG. 4 is a functional block diagram illustrating components of an implementation of a compute node having utility in connection with a data processing and analytics system such as that depicted in FIG. 1. The FIG. 4 example represents one implementation of a compute node 140n such as illustrated in FIG. 1, and generally comprises communications link 149n, a programmable logic block 410, a processing system 430, and node memory 450.

[0071] In the FIG. 4 example, compute node 140n (in tier "n" of a particular stack of compute nodes 140) may be communicatively coupled, via communications link 149n to one or more additional compute nodes 140 connected in series at different successive tiers, *i.e.*, to compute node 140n - 1 and compute node 140n + 1. As described above, communications link 149n may be based upon or operative in accordance with standard serial interfaces or protocols, or may

employ proprietary serial link technologies, for example, as a function of or influenced by the functionality and operational characteristics of compute array 142.

[0072] In the FIG. 4 implementation, programmable logic block 410 is illustrated as generally comprising, in addition to compute array 142 and DMA 146a, a storage interface component 411n. As noted above, compute array 142 may employ or comprise any of various commercial embodiments of an FPGA, and many alternatives may be equally suitable or adequate, depending upon design considerations such as power consumption, thermal factors, throughput and processing capabilities, overall requirements of system 100, price constraints, or a combination of these and other boundaries conditions that are typical in electronics design. DMA 146a may be employed to handle data traffic between processing system 430 or node memory 450, on the one hand, and communications link 149n, on the other hand, as illustrated in FIG. 4. In particular, DMA 146a may be embodied in or comprise a data transfer apparatus, device controller, or other hardware or firmware module that is configured and operative to facilitate bi-directional data communications between electronic devices, either independently or in cooperation with control signals or other instructions received from processing system 430 or specific components thereof. Any of various commercially available hardware- or firmware- supported datamover implementations may be suitable for DMA 146a, provided that a particular implementation enables data transfer operations and throughput bandwidths sufficient to support required or desired functionality of compute node 140n, in particular, and system 100, in general.

[0073] Storage interface component 411n may be one of a number, “n,” of such components employed in connection with a particular programmable logic block 410, as represented by the stack on the right side of FIG. 4. While a number of similar or identical such storage interface components may be integrated with or implemented in cooperation with programmable logic block 410, only one storage interface component 411n is described; it is noted that every such storage interface component in the illustrated stack need not be identical, and that various alternatives to the specific storage interface component 411n may be suitable for a particular implementation of programmable logic 410. The functionality and specific architecture of storage interface component 411n may vary as a function of data store 143 requirements, the operational characteristics and processing capabilities of compute array 142, or a combination of these and other factors. As noted above, a particular data store 143 having utility in connection with system 100 may be embodied in or comprise a mass data storage component, such as an EEPROM store. In the FIG. 4 embodiment, data store 143 may be a Flash memory, and accordingly, data store interface 145n may employ an ONFI protocol. In operation, data store interface 145n may

manage access (*e.g.*, read and write access) to data store 143, as is generally known in the art. Other types of controller technology may be substituted as a function of the particular technology employed at data store 143, for example.

[0074] Storage interface component 411n may also comprise DMA 146b. As with DMA 146a, those of skill in the art will appreciate that any of various commercially available hardware- or firmware- supported datamover implementations (such as a data transfer apparatus, device controller, or other hardware or firmware module) may be suitable for use at or in conjunction with DMA 146b. In operation, DMA 146b may facilitate bi-directional data communications between compute array 142 or processing system 430 (and node memory 450), on the one hand, and storage interface component 411n, on the other hand. DMA 146b may function independently or autonomously, for example, or in cooperation with control signals or other instructions received from processing system 430 or specific components thereof, to effectuate data transfer operations sufficient to support operation of compute array 142 with respect to read/write or other access to data store 143.

[0075] Storage interface component 411n may also comprise a routing block 413n. Routing block 413n may be implemented as a first in / first out (FIFO) buffer or similar memory structure temporarily to store one or both of data transferred (via data store interface 145n) between compute array 142 and data store 143, and data transferred (via DMA 146b) between compute array 142 and node memory 450. Various commercial implementations of a FIFO buffer structure are available (or may be developed and operative in accordance with known principles) to provide the functionality of routing block 413n. In some implementations, routing block 413n may comprise hardware logic, or may be operative in accordance with software instruction sets, sufficient to allow routing block 413n to access or to target specific memory locations in data store 143, either independently or in cooperation with compute array 142, processing system 430, data store interface 145n, or a combination of these and other components. The disclosed subject matter is not intended to be limited by the nature, structural configuration, or operational characteristics of routing block 413n, as many alternatives may be employed as a function of design considerations and operational requirements of system 100, in general, and compute node 140n, in particular.

[0076] Processing system 430 may generally comprise a first set of one or more processors 432a and 432b and a second set of one or more processors 439a, 439b, 439c, through 439n. Bus structures, power sources, controllers, internal memory, and other attendant and supporting hardware and communications paths have been omitted from FIG. 4 for clarity. It is noted that all

of the processor blocks (432a, 432b, 439a, 439b, 439c, and 439n) depicted in FIG. 4 may be implemented as or generally comprise a microprocessor, a multi-core microprocessor, a microcontroller, an FPGA, a programmable logic controller (PLC), a programmable logic unit (PLU), or some similar or analogous hardware component suitable for data processing as is generally known in the art. In one implementation, for example, one or both of processors 432a and 432b may be embodied in or comprise a Cortex™ A5™ microprocessor core marketed by ARM Holdings PLC, and some or all of processors 439a, 439b, 439c, through 439n may be embodied in or comprise a Cortex A53™ microprocessor core marketed by ARM. Other configurations (*e.g.*, the number of processors employed) and specific types of processing devices are contemplated, and may readily be implemented, as a function of, or influenced by, design goals, processing overhead, application-specific requirements for compute node 140n, and a variety of other factors. It is also noted that the example of processing system 430 illustrated in FIG. 4 represents only one implementation of processing system 148 depicted in FIG. 1, and that the FIG. 4 embodiment may comprise more or fewer processing resources than those shown.

[0077] In operation, processors 432a and 432b may manage or otherwise influence operation of DMAs 146a and 146b, respectively. Also as depicted in FIG. 4, each of processors 432a and 432b may employ on chip memory (OCM) to facilitate interoperation with DMAs 146. In that regard, DMAs 146a and 146b may be configured and operative to transfer data between and amongst components of compute node 140n as illustrated in FIG. 4 under control of or responsive to instructions from processors 432a and 432b; some or all of such instructions may be resident in or accessible by OCM components or hardware. Additionally or alternatively, OCM may also host or maintain status updates and attendant data acquired from or associated with operation of mailbox 452n described below. In some instances, OCM may have a lower latency than external memory, such as a DDR RAM component, and by employing OCM to manage mailbox 452n updates, memory 450 bandwidth may be reserved for payload traffic, rather than administrative traffic.

[0078] In some implementations, operation of processors 432a and 432b may be influenced by interrupts received, for example, from DMAs 146a and 146b. For example, one or both of DMAs 146a and 146b may be operative to issue an interrupt following expiration of a predetermined or dynamically selected duration (*i.e.*, a “timeout”), or upon the occurrence of a predetermined or pre-established event (such as completion of a certain threshold number of transactions). In this example, one or both of processors 432a and 432b may be operative to

initiate exchange of mailbox 452n status information, or to execute some other administrative or substantive function, upon receipt of such an interrupt. Additionally or alternatively, processors 432a and 432b may be configured and operative to schedule or otherwise to influence operation of processors 439a, 439b, 439c, through 439n, *e.g.*, via interprocessor interrupts or other communications protocols or instructions.

[0079] Specifically, processors 439a, 439b, 439c, through 439n may be responsible for a variety of tasks which, under some circumstances, may be requested or required by processors 432a and 432b, or which may be initiated autonomously or otherwise without direction from external components such as processors 432a and 432b. In various implementations, processors 439a, 439b, 439c, through 439n may perform some or all of the following functions, or a subset thereof, as requirements of system 100, in general, or node 140n, dictate: data transformation and storage (*e.g.*, with respect to node memory 450, memory store 143, or both); data dispatch for write/read operations (again, with respect to node memory 450, memory store 143, or both); command interpretation and execution (*e.g.*, with respect to data processing and algorithm compilation and execution); algorithm load and compute; and memory exception handling. Those of skill in the art will appreciate that various other functions may be assigned to or attributable to processing system 430 as a function of desired operability of compute node 140, and that any of numerous hardware arrangements, computing architectures, and distributed processing resources may be employed at processing system 430 to enable its functionality in the context of supporting desired capabilities of compute node 140.

[0080] Processing system 430 may cooperate with programmable logic block 410 for data transfer and read/write operations with respect to data store 143. In particular, flexible FIFO or other buffer structures and multiplexers may be employed to provide multiple data paths to and from each memory (such as Flash) channel. Some examples of specific data paths and supporting hardware are described below with reference to FIGS. 5 through 7.

[0081] In the FIG. 4 example, node memory 450 is one representation of the node memory 144 illustrated and described above with reference to FIG. 1. In that regard, node memory 450 may be implemented as or comprise DDR4 RAM, or other volatile or non-volatile memory suitable to support operations of compute node 140n and processing system 430. In some implementations, node memory 450 may generally comprise one or more mailboxes (*i.e.*, resource-specific data storage cells or memory locations) 452n, as well as one or more functional programming (FP) tables 454n, algorithm buffers 456n, and FP buffers 458n. These components may generally be operative to store data and instructions necessary or desirable for processing

system 430 to perform write/read/execute operations in support of programmable logic block 410 as is generally known. An example is provided below.

[0082] In order to minimize the need for processor 432a intervention during write transactions from host compute system 199 to compute node 140n, a largely autonomous mailbox 452n and DMA 146a strategy may handle the entire transfer (or a portion thereof) of incoming data packets to DDR4 (or similar or analogous) buffers at compute node 140n, in general, and at node memory 450, in particular. In some implementations, it may be desirable that the inbound buffers are "content agnostic" and are treated generically. In that regard, a plurality (such as one hundred twenty eight, for instance) of mailboxes 452n may be initialized for this purpose, and their status may be set to "empty" at boot time.

[0083] In programmable logic block 410, inbound DMA 146a may also be initialized and its internal control logic may be set to "full availability." In this instance, when data packets arrive, the first available mailbox 452n address may be retrieved (*e.g.*, from a local list maintained in node memory 450), and the incoming packed may be posted to that particular mailbox 452n. The local list may then be updated to reflect that a particular mailbox 452n is occupied, and that a next mailbox 452n in series, or otherwise selected or determined in accordance with processing system logic, should be used for a next inbound data packet or series of data packets. At that point, two processes may be initiated, as noted briefly above. First, a transaction counter or other meter may be incremented (or decremented, as a matter of design choice); once a message count threshold is reached, attaining the threshold may trigger an update of the occupied mailbox list by processing system 430. Second, a timer may be set; once a predetermined period of time has elapsed (*i.e.*, a timeout event), expiration of the time period may trigger an update, just as if the message count had been reached. This second process may ensure that single or last messages will be captured, even in the event that a message or transaction counter does not reach a predetermined or minimum threshold to trigger an update in accordance with the first process. In some implementations, during such an update, programmable logic block 410 may also receive an updated list (*e.g.*, a message or instruction representative of emptied mailboxes 452n since a last transaction, or some other relevant data) as prepared by processing system 430.

[0084] With the foregoing overview of system 100 and compute node 140n architecture in mind, the following addresses one example of algorithm load and execute initiation and functionality, by way of example only, and not by way of limitation. In one implementation, initialization of each compute node 140n may begin by booting from a local golden bit stream; this

initialization may generally provide a common (*i.e.*, across all compute nodes 140) infrastructure in each respective programmable logic block 410. For example, initialization may define data paths (such as those described below with reference to FIGS. 5 through 7), protocols for communications link 149n, identification and initialization of various I/O controllers, and the like – generally establishing and defining the operating environment for each compute node 140n ("compute node environment"). In one embodiment, the initial bit stream may include an algorithm block or other information concerning particular algorithm execution, though this may occur after the initial bit stream, if necessary or desired. In some circumstances, initialization may also provide boot code or other initial parameters for the various processors in processing system 430.

[0085] Upon initialization of compute node 140n, new or updated algorithm bit streams may then be transmitted, *e.g.*, by host compute system 199, via router module 120 and communications link 149n. In some implementations, algorithm bit streams may contain both algorithm logic and the same compute node environment used in the local golden bit stream employed during initialization. When an algorithm bit stream is received from (*e.g.*, from host compute system 199), some or all of the following processes may ensue: the algorithm bit stream may be provided to all compute nodes 140 simultaneously, or substantially simultaneously or concomitantly; each respective compute node 140n may complete any outstanding data transactions; normal compute node 140n operation may be briefly paused while the new bit stream is loaded into all compute arrays 142; normal compute node 140n operations may be resumed and all compute nodes 140 may begin executing the algorithm (and continue to do so until it completes or times out); and upon completion, computational results may be returned along with status, and the algorithm bit stream may be left in place for potential re-use at a later time.

[0086] Regarding some of the data paths supporting the foregoing functionality, it is noted that FIGS. 5 through 7 are diagrams illustrating examples of data routing strategies to route data through portions of the compute node depicted in FIG. 4.In the illustrated example, each storage interface component 411n may generally comprise a group of FIFO buffers or similar structures (such as represented by routing block 413n) connected with a versatile routing fabric. During operation, the routing fabric allows for run time configuration of data flow paths through routing block 413n and (1) to/from node memory 450 or OCM buffers associated with components of processing system 430, (2) to/from data store interface 145n, (3) to/from programmable logic block 410 compute pipelines, (4) to/from programmable logic block 410 path to communications link 149n.

[0087] In some embodiments, all such data paths may be implemented with 16-bit width and follow the AXI stream protocol. In that regard, an AXI stream is generally an address-less transport mechanism that requires a minimal amount of control signals (*e.g.*, only READY, VALID and LAST). Using this example approach, data flow may begin as soon as an intended target of a data transaction indicates "READY" and a source of the data transaction indicates "VALID." Conversely, data flow may be suspended when one of the foregoing two conditions is no longer true. The "LAST" indicator may be used by logic in the target to detect packet or data stream boundaries, for example.

[0088] In some implementations of interface routing fabric employed at routing block 413n, one or more 4:1 multiplexers may have particular utility. For instance, a 4:1 multiplexer may cooperate well with a 6:1 (four data signals, two control signals) logical unit (LUN) data structure (described below), and therefore may maximize use of the basic elements found in some viable options for FPGA structures of compute array 142. Other types of multiplexers may be employed, for instance, to match a fabric used in a commercial implementation of compute array 142, to combine certain functionality set forth below into fewer components, or as a result of a variety of other considerations. A 4:1 multiplexer also provides a reasonable multiplexing strategy, to the extent that it can devote one input section to each of the four sources noted above, which may generally cover all use cases in most practical applications and hardware configurations.

[0089] In addition to the 4:1 input section, each multiplexer output may be configured and operative to drive multiple targets simultaneously as each path may be implemented as 4:1 source select. So, in such an example, one implementation of a routing paradigm may focus on the source of the data, rather than the component to which the data are to be transmitted. This paradigm allows each source to provide data to multiple targets simultaneously (or substantially simultaneously), and it makes an illegal or conflicting configuration (*e.g.*, two or more sources trying to drive one target simultaneously) impossible.

[0090] In some implementations, some AXI stream flow control signals may be treated with some extra logic applicable to the architecture and methodologies described herein. For instance, some logic rules for AXI stream flow control may be expanded as follows: READY signals of data sinks that are connected to a common source may be ANDed (*i.e.*, combined); READY signals for destinations that not selected (*i.e.*, inactive) may be set to "1" by default; and VALID status may only be presented by a data source when it sees the ANDed READY from all possible destinations (data sinks) to prevent a single sink from initiating the dataflow. This mechanism is very flexible and resource efficient. It may generally enable a versatile interconnect with respect

to the compute pipeline, in accordance with which one source value may need to feed multiple compute engines in parallel (*e.g.*, count a value "x," square "x," and multiply "x" by a different value "y"). By instantiating multiple concurrent "listening ports" (*i.e.*, compute elements) to monitor one data source, it is possible to implement an arbitrary number of concurrent calculations in parallel, subject to the architectural and processing limitations of the hardware embodying or employed at compute array 142, for example.

[0091] FIGS. 5 through 7 show such an arrangement with two parallel FIFOs or other buffer structures, each of which is fed by an input multiplexer 511 coupled to all source interconnects, and selectively outputs to any of three paths to programmable logic block 410 for compute. The actual count of paths to programmable logic block 410 compute resources may depend, for example, upon the loaded algorithm, the corresponding compute pipeline, and a variety of other factors.

[0092] In particular, each of FIGS. 5 through 7 show four inputs to input multiplexers 511: from data store 143 (*e.g.*, "Flash Channel" via data store interface 145n); from node memory 450 (*e.g.*, via DMA 146a); a first input from the compute pipeline (*i.e.*, programmable logic block 410 compute array 142); and a second input from the compute pipeline (*i.e.*, programmable logic block 410 path to communications link 149n). These inputs are concatenated at input multiplexers 511; output from the respective FIFOs may be concatenated at output multiplexers 512.

[0093] FIG. 5 illustrates one data path from data store 143 (*e.g.*, "Flash Channel" via data store interface 145n) to node memory 450 (*e.g.*, via DMA 146a), and a separate data path from node memory 450 (*e.g.*, via DMA 146a) to data store 143 (*e.g.*, "Flash Channel" via data store interface 145n). FIG. 6 illustrates one data path from data store 143 (*e.g.*, "Flash Channel" via data store interface 145n) to one of the compute pipelines, and a separate data path from node memory 450 (*e.g.*, via DMA 146a) to a different one of the compute pipelines. FIG. 7 illustrates one data path from data store 143 (*e.g.*, "Flash Channel" via data store interface 145n) to one of the compute pipelines, and a separate data path from one of the compute pipelines to node memory 450 (*e.g.*, via DMA 146a). Those of skill in the art will appreciate that various modifications and alterations to the implementations shown in FIGS. 5 through 7 are possible, and that numerous data paths between the various components may be defined as necessary or useful for operation of a particular compute node 140n.

[0094] FIG. 8 is a diagram illustrating one implementation of a memory architecture having utility in connection with a compute node such as that depicted in FIG. 4, and FIG. 9 is a diagram illustrating one implementation of a portion of the memory architecture depicted in FIG. 8.

[0095] In some implementations, each compute node 140n may employ sixteen physical channels (or buses) for access to data store 143. In FIG. 8, these physical channels (reference numeral 899) are depicted in vertical columns labeled, from left to right, 0, 1, . . . , 15. On each respective channel 899, one arrangement may employ one or two memory chips, each of which may comprise one or more logical units (LUNs) (reference numeral 810). In FIG. 8, LUNs associated with channel 0 are labeled 810a, LUNs associated with channel 1 are labeled 810b, and LUNs associated with channel 15 are labeled 810n. It is noted that more or fewer channels 899 may be employed, and more or fewer LUNs 810 per channel may be implemented, depending upon overall arrangement and processing functionality of a particular compute node 140n, the type of memory employed at data store 143, and a variety of other factors.

[0096] In the context of the disclosed architecture and data path approaches set forth herein, it will be appreciated that a LUN 810 may execute more than one memory access (such as a Flash access) operation concurrently, even on the same Flash (or other memory) chip. In addition to LUNs 810, each chip in the FIG. 8 arrangement may generally feature multiple planes (as indicated by the stacks of LUNs 810 – each LUN 810 in a particular stack resides at a different respective plane in the architecture). By utilizing multiple (in this case, four) planes concurrently, the available access to a given memory page, in effect, quadruples in size, thereby improving the ratio of data transfer to command bandwidth.

[0097] In the discussion that follows, LUNs 810, in particular, and data store 143, in general, are referred to in terms of Flash memory for convenience; as noted above, however, the disclosed subject matter is not intended to be limited to the particular type of memory used for data store 143, except to the extent that changing a memory type or employing a different memory technology at data store 143 may influence selection and implementation of suitable or appropriate interface mechanisms, such as at data store interface 145. In any event, references to Flash memory are by way of example only, and not by way of limitation.

[0098] In some instances, data in each LUN 810n may be stored in the form of Flash pages, one example of which may generally comprise 16Kb of data payload, plus some overhead associated with error correction code (ECC) and header information. Pages may be organized into blocks, for instance, or sets of 1024 pages. This is illustrated best in FIG. 9, which depicts a LUN 810n comprising two blocks 811 of data; more than two blocks 811 per LUN 810 are contemplated. Similarly, while each block 811 is depicted as comprising only three discrete pages 812, it may be desirable that a block 811 comprise 1024 pages, 2048 pages, or more. Each page 812, in turn, may generally comprise an index 813, which may record metadata, information, or

metrics or statistics associated with the data payload of the page 812. For instance, index 813 may include ECC information and a data type indicator, a first index, a record count, an incarnation counter, various statistics (such as a minimum data value, a maximum data value, a local average value, *etc.*) and other information, or a combination of these and other types of information, with respect to data in a particular page 812. Additionally, LUN 810n may include a cache 819, which may serve as temporary storage for a page 812, or a portion thereof, for various purposes such as read operations.

[0099] For example, in order to read a given page 812, a read command is issued (for example, by compute array 142 or processing system 430, either individually or in cooperation). Such a read command may generally cause LUN 810n to copy page 812 to (on-chip) cache 819, from where it may be read rapidly across data store interface 145 and communicated to compute array 142.

[00100] As is generally known in the art, a Flash page 812 may only be written in its entirety in the context of any particular write operation; specifically, a particular Flash page 812 must be erased, in its entirety, before any new contents or data may be written to that particular page 812. This is true even if only relatively few data values (as compared to the overall contents of the particular page 812) are sought to be changed; in every case, manipulating data in a page 812 typically requires that the entire page 812 be erased and subsequently re-written with modified contents. To make matters worse, particularly in the context of Flash memory, not only must a page 812 be erased before any new value may be re-written (or altered) in that page 812, but the entire block 811 containing the affected page 812 must also be re-written.

[00101] In one implementation of LUN 810n and a method of interacting with Flash memory, erase granularity may be set as one block 811 (*e.g.*, a set of 1024 pages) as noted above; in accordance with a novel approach to an old problem, one method of updating a page 812 is to read its contents, selectively to modify whatever data are to be changed, and subsequently to write a new copy of that page 812 to a new location in memory – typically in the same block 811, or at least in the same LUN 810n, but this may not always be practical or even desired – and with an updated identifier to facilitate identification of the new copy as the definitive, updated version. As it is not economical to erase an entire block 811 (and to move all the unaltered pages 812 that may reside in the same block 811 to a new location) each time a page 812 is updated or erased, the old page 812 may remain unaltered and in its original location within the block 811.

[00102] To manage the resulting multiple copies of a page 812, each specific page 812 may be provided with an incarnation counter (or similar indicium or identifier) in its metadata; as noted

above, it may be useful that this incarnation counter is updated each time a page 812 is copied during an update or modification process. In some implementations, such an incarnation number count or other identifier may be read during boot and used to populate a page pointer index (not shown) or look up table that resides in, or is accessible by, node memory 450 or processing system 430, for example. In one approach, only the copy of a particular page 812 having the highest value in the incarnation counter may be identified in the pointer index or look up table, as it is this copy that represents the active and valid copy of that page 812 – all other copies contain data that have not been updated to reflect new values. In some embodiments, the page pointer index or look up table may also be updated when a page 812 is copied and its incarnation counter is incremented. The incarnation counter may be selected to have a range that is large enough (*e.g.*, 24 bit, double check) so that wrap-around conditions (*i.e.*, a reset from the highest value in the range back down to the lowest) occur infrequently.

[00103] Upon wrap around of a page incarnation counter, two methods of exception handling may be applied. In a brute force approach, all remaining stale copies of the page 812 may be eliminated; it will be appreciated that this approach may trigger a great number of page moves, even in the case where neighboring page data are still valid. In an alternative approach, an algorithm that closely interacts with garbage collection and block recycling (described below) may be employed to track the correct highest count even after wrap around has occurred. Many such algorithms may be devised to keep track of a most recently modified version of a page 812 after a counter wraps around, and the present disclosure is not intended to be limited by any particular approach or methodology.

[00104] In some embodiments, blocks 811 may be recycled efficiently by tracking a valid page (VP) count identifier and a continuity factor (CF), where VP count may be an indication of a number of valid pages 812 written to a block 811, and CF may be an indication of a number, or percentage, of pages 812 within a block 811 that no longer contain up to date data. In accordance with this paradigm, a block 811 having a low VP and a low CF is at the beginning of its life span – not all pages 812 have been written yet, and relatively few pages 812 are out of date – and hence this particular block 811 may be assigned a low priority for recycling operations. A block 811 with a high VP but a low CF may be nearly full or totally full – most, if not all, pages 812 have been written – but the block 811 as a whole contains relatively few pages 812 that have been subsequently invalidated by more updated versions; such a block 811 may be assigned a medium or middling priority for recycling. A block 811 with a high CF, on the other hand, may be

considered a candidate for recycling having a relatively higher priority, since many, if not most, of its pages 812 no longer contain valid data.

[00105] In accordance with one strategy, as long as there are sufficient blocks 811 available in a given LUN 810n to accommodate short term copying of pages 812 without rewriting or excessive wrap around activity, no old block 811 should be forcefully recycled as long as it has space to allocate to more than one unwritten page 812; this strategy may minimize, or at least reduce, unnecessary page writes. When free resources in a given block 811 become sparse, the block(s) 811 with the highest CF and the lowest VP may be freed by moving the remaining valid pages 812 to other blocks 811 having free space, and then erasing the block 811 from which all remaining valid pages 812 have been moved.

[00106] In most conventional Flash applications, retrieving (reading) a page 812 has a latency of about 100µs, as the page 812 is first copied to an on-chip cache 819 as set forth above; this typically occurs before the page 812 is ready to be transferred across a bus (*e.g.*, via data store interface 145n) to compute array 142. The data transfer of one page 812 from that on-chip cache 819 to the compute array 142, in contrast, requires less than a third of that overall time – *i.e.*, around 32 µs.

[00107] In order to mask the foregoing latency, each physical channel 899 in FIG. 8 may accommodate multiple LUNs 810, each of which may be retrieving pages 812 to cache 819 concurrently. Once an array of LUNs 810 (or targets) is loaded with read commands, further read commands may be interleaved while data from pending reads are retrieved from cache 819. With sufficient targets (for example, four or more) on a channel 899, the available bandwidth for a given channel 899 may be fully saturated with data moves (streaming). As the LUN 810 count increases, it becomes possible to read partial pages 812 (for example, a ½ page or a ¼ page) without any substantial loss in channel utilization (interleaving the additional read commands has minor impact on overall throughput). Reading partial pages 812 efficiently becomes a valuable feature in more advanced data mapping methods (see the discussion of data structures with reference to FIGS. 10 through 14 below).

[00108] Specifically, executing commands against a LUN 810 on a Flash chip will always result in latencies and minimum (but not deterministic) wait times before a previously issued command is executed and the LUN 810 is available for further exchanges. As noted above, the biggest latencies are related to commands that retrieve a page 812 (around 100µs) and write a page 812. During those wait times, many features of a LUN 810 are generally unavailable. The status of a

LUN 810 may be polled by issuing a status request command (for instance, from storage interface component 411n or elements thereof such as routing block 413n).

[00109] As also noted above, in accordance with the disclosed architecture, payload data and command sequences share the same bus or physical channel 899. Accordingly, to maximize throughput on a given channel 899, LUN 810 access may be interleaved across many targets (*i.e.*, pages 812), and an estimate of availability of each LUN 810 may be determined and maintained in order to minimize interrogation overhead. This functionality, and that set forth below, may be provided by storage interface component 411n, in general, and routing block 413n and data store interface 145n (either individually or in combination), in particular.

[00110] In that regard, storage interface component 411n or its components may take all of these requirements into consideration, and may be configured and operative to provide enough processing functionality to reduce or to eliminate the requirement for central processing unit (CPU) intervention (either by compute array 142, processing system 430, or both) by providing hardware resources necessary or sufficient to effectuate some of all of the following (*e.g.*, in routing block 413n): provide or control a dedicated command buffer per LUN 810 of a given physical channel 899; provide or control a dedicated command execution timer per LUN 810 (this timer may be set with a minimum latency of the Flash chip after issuing a command, such that status polling only begins after the minimum delay time has elapsed, and therefore keeps the channel 899 less congested with status requests); provide an autonomous status request command generator, so that status polling does not require CPU intervention; and provide a session identifier buffer for each command that is posted by processing system 430 into the various command FIFOs in routing block 413n (this may allow processing system 430 to "set and (almost) forget" issued commands, as storage interface component 411n may post completed (or failed) transaction updates in message mailboxes 452n along with a corresponding session identifier). In that regard, mailboxes 452n may have the utility set forth above with reference to FIG. 4.

[00111] It is also noted that routing block 413n, memory interface 145n, or a combination of these and (possibly or optionally) other components of storage interface component 411n may also provide hardware, firmware, or software resources necessary or sufficient to effectuate selective storage of data in a "co-located" manner. In this context, the terms "co-located" or "co-location" generally refer to the operational characteristic (or its practical effect) of maintaining data records that are related to each other in or at memory addresses such as are reflected by or implemented in blocks 811, pages 812, or both that are relatively proximate to each other (*e.g.*,

in physical space, logical space, or both) as compared to other blocks 811 or pages 812 in data store 143. Header information or metadata associated with particular data may be used to designate or otherwise to identify certain data as related to or relevant with respect to other data for this purpose. As set forth above, data may be supplied to system 100 with information (that may be recorded in index 813, for example) that may include a data type index or identifier, various statistics, and other information that may be used, among other things, to co-locate particular data with respect to other related data in a particular page 812; in the foregoing manner, related data may be stored in adjacent or contiguous pages 812 or blocks 811, facilitating retrieval of related data using reduced or minimal read operations.

[00112] One particular implementation that may benefit from co-located data may be configured and operative to organize large datasets acquired, by way of example, from a STARE (Surveillance Technology Advancement & Replacement for Ebsicons) data acquisition system. A typical STARE system employs a variety of sensors (*e.g.*, nine or more independent sensors, typically deployed in a satellite or high-altitude surveillance aircraft) that acquire optical or other electromagnetic data across a large geographic area and at successive points in time, such as for satellite reconnaissance, weather observations, time-lapse photographic applications, and the like. While the following discussion addresses such a STARE system, the disclosed subject matter is equally applicable to other types of large datasets, as will generally be appreciated by those of skill in the art. For example, on a micro-scale, the types of data storage and analytics techniques employed in a typical STARE application may be equally applicable to computer tomography (CT) scans, computer aided detection (CAD) mammography analyses, and other medical technologies involving optical or image data, as well as to design verification and quality assurance methodologies (such as machine vision inspection techniques, for instance) in the semiconductor, micro-electrical-mechanical (MEM) machines, and other small scale electrical and mechanical industries.

[00113] In accordance with one implementation, a respective data stream from each of a plurality of data sources (for instance, each respective sensor in a STARE array or each respective channel in a medical imaging device) may be ingested as a dedicated dataset for a given point in time (*e.g.*, a temporal STARE index). In this example, the respective output of each respective sensor or channel may be considered an independent data source, while the data from each respective independent data source may be considered (and identified, designated, or otherwise characterized as) “related” to all the other data acquired from others of the independent data sources. For instance, data acquired by a specific sensor operating in a specific spectral

band on a given satellite may be considered a specific, independent data source; if that satellite employs multiple sensors, each operating in a respective (though, perhaps, overlapping) spectral band such that the satellite, as a whole, acquires and provides data across multiple spectral bands, each such independent data stream may be ingested by system 100 (and, in particular, by compute node 140n) as different, independent, but related datasets. As noted above, the relationship between such independent datasets may be encoded in metadata or header information, for example, such that related datasets may be co-located when stored or written to data store 143 or other memory component.

[00114] In the foregoing manner, and in the context of a STARE system, each independent dataset may represent a collection of data points from a specific sensor observing or dedicated to a specific geographic region at one point in time. In the case of an arthroscopic, mammographic, CT, or other procedure involving medical imagery or other data collection application, for instance, each independent dataset may represent a time-specific or temporally varying collection of data points from a specific type of sensor covering a specific anatomical region or that is dedicated to collecting a particular type of data. Applications include automotive diagnostics, time-dependent stresses in aerospace structures, inventory and logistics tracking, machine vision and machine learning implementations, and any other application in which large datasets and analytics techniques are relevant.

[00115] Returning to the STARE example, each record in a particular independent dataset may include fields to designate, for instance, a STARE index and sensor data sufficient to identify a particular data source (*i.e.*, the sensor from which the data are acquired), a position of each sample in the data stream (*e.g.*, a time stamp or other unique indicium), and any related data streams to facilitate co-location of data during storage. It is noted that some or all data acquired by some or all of the plurality of data sources may be multi-dimensional (*e.g.*, intensity, frequency, directionality or orientation, altitude or distance from sensor to imaged object, and the like, depending upon the application and the type of data collected). Some or all of these factors may be encoded into, included in, or referenced by header information or other metadata associated with each data packet provided by the independent data source or otherwise derived by other components as a function of or dependent upon the independent data source and *a prior* or predetermined knowledge of its operational characteristics or deployment.

[00116] In some implementations, a user or administrator of system 100 may choose to divide data associated with independent data sources which cover large geographic areas (or other areas or regions of interest) with high resolution into multiple smaller datasets, each of which may

include or be specifically directed to a fraction of the overall geographic area or region of interest that is covered by the data source as a whole. If this is done with a suitable amount of spatial overlap, boundary conditions that may arise from distributing the smaller datasets to multiple compute nodes 140n may be mitigated.

[00117] Those of skill in the art will appreciate that the foregoing functionality may be readily implemented, for example, via suitable programming in an application programming interface (API) and/or in logic at compute array 142, routing block 143n, or both, such that related data may be recognized as such and ingested and reproduced as indexed and related datasets in connection with STARE or other applications. In some implementations, permanent, non-intrusive detailed instrumentation may be integrated into programmable logic block 410, for instance, and cooperate with an API to effectuate appropriate hardware configurations, analysis interfaces, and setup methods to accommodate this embodiment supporting co-location of related data.

[00118] Some of the same factors that enable co-location of data also facilitate implementations in which algorithms may operate on multiple datasets. In that regard, it will be appreciated that typical algorithms employ only a single dataset for input; this is both conventional and sensible, since a typical algorithm seeks to apply the same set of rules or instruction sets to data that are related in some way, and since the relationship between particular data points is conventionally defined by the fact that the data points reside in the same dataset. Where, as set forth above, different, independent data streams may be identified as related (*e.g.*, by header information or metadata) across independent datasets to support co-location, however, such relatedness may be leveraged to enable a particular algorithm to execute on multiple datasets in a single instantiation of the algorithm.

[00119] By way of example, a typical API call to execute an algorithm only allows the use of fields that belong to a single (open or active) dataset; multiple dataset computations are difficult to implement and require extensive programming and computational (and data record access) overhead. However, using the above techniques that contemplate importing and storing data streams as dedicated datasets from independent data sources, and since those multiple sources may be considered for thresholding calculations and other house-keeping and initiation procedures, an API or programmable logic block 410, or both, may readily be modified to accommodate use of multiple datasets by a single instantiation of an algorithm. In that regard, programmable logic block 410 may be implemented such that it allows for each field which provides data for computations to be a member of any dataset that is available (*e.g.*, stored in

data store 143n or other applicable memory structure). In a STARE implementation, such algorithms may be embodied in or comprise Connected Component Labelling (CCL) algorithms or other implementations, though these and other algorithms or heuristic approaches may have equal utility in image processing and other applications as noted above.

[00120] FIG. 10 is a diagram illustrating a conventional data record structure, and FIG. 11 is a diagram illustrating one implementation of a data record structure having utility in connection with a compute node operative in accordance with the present disclosure. Further, FIGS. 12 through 14G are diagrams illustrating additional implementations of data record structures having utility in connection with a compute node operative in accordance with the present disclosure.

[00121] In most Big Data and high performance computing applications, many computations may be performed on columns of data, or vectors, rather than on a single data point or value. This typical mode of operation may indicate or suggest that one useful way to store data in Flash is to map columns to dedicated Flash pages for efficient retrieval. On the contrary, however, some applications may benefit from organizing data in records organized in such a manner that a Flash page holds multiple fields (columns) of a common record. Prior data management and data processing systems address these competing requirements by storing the data twice: once, organized in columns; and then again, organized in records.

[00122] In accordance with the disclosed architecture of system 100, since each compute node 140 may have sixteen or more physical channels 899 that may operate concurrently, a sophisticated data mapping method, examples of which are set forth below, may allow for certain features that are typically unavailable in the context of single channel storage solutions.

[00123] The following examples, and the illustrations in FIGS. 10 through 14G, are based upon a hypothetical dataset of eight records, each record consisting of eight fields or columns. In the drawing figures, a respective color (represented by a unique hatching pattern) is used to indicate a respective column or field type in a record, and a unique number is used to indicate a row or index for each unique record. In the context of the present disclosure, the terms “record” and “page” may be used interchangeably, and the terms “field” and “column” may be used interchangeably.

[00124] In the prior art data record structure depicted in FIG. 10, for example, each distinct row (1 through 8) represents a distinct data record (numbered 1 through 8); as indicated by the legend “Records 1 – y,” it will be appreciated that such a data record structure may include more or fewer records than shown. Each column in FIG. 10 represents a particular field, each of which is distinct

from the others as indicated by the respective color hatching; as indicated by the legend "Fields 1 – x," it will be appreciated that each record (1 through 8) may include more or fewer fields than shown.

[00125] In accordance with the disclosed subject matter, however, when staged or reformatted for storage in Flash (such as data store 143), the data may be, in effect, rotated by 90 degrees, such that all original fields in the same original column are stored in a new dedicated page (or pages), *i.e.*, a new dedicated page comprises fields, each comprising a single field type. This is illustrated in FIG. 11. Specifically, each respective row in FIG. 11 represents a new respective record, 1 through x (eight, in this example); *i.e.*, one new record for each discrete field in the original data structure. Each new record comprises a number of fields (eight, in this example) 1 through y, corresponding to the number of records in the original data structure of FIG. 10, and each field is of the same field type mapped from the same column in the multiple records in the original data structure.

[00126] In the foregoing manner, original records may be re-mapped into columns, and original fields may be re-mapped into discrete records or pages 812, each comprising a column from the original data structure; each new record (*i.e.*, representing an original column) may be stored in a dedicated Flash page 812 of any channel 899. In the FIG. 11 example, page 812 retrieval may be executed efficiently, as set forth above, though fetching related data may be further optimized with some additional manipulation.

[00127] For example, if only one channel 899 were available, one additional mechanism to facilitate optimization may be to spread the newly formatted pages 812 across multiple LUNs 810 so that access to the various pages 812 may be executed in an interleaved manner. As noted above, such an interleaving strategy may have a positive effect on overall throughput with respect to the transfer time associated with writing a page 812 to cache 819 in the Flash array; in particular, this specific latency type of may be masked by use of interleaving, in accordance with which other data types or records from a second LUN 810 may be transferred across routing block 413n to compute array 142 while a first cache 819 associated with a first LUN 810 is still filling. This arrangement is illustrated in FIG. 12, which depicts the data structure of FIG. 11 distributed across multiple LUNs 810a, 810b, 810c, … 810n.

[00128] In some implementations in which each compute node 140 accommodates sixteen (or more) parallel channels 899, it may be useful to distribute data traffic across all available channels 899 as described above with reference to FIGS. 4 through 9. In such an embodiment, it may be useful to distribute the data structure of FIG. 11 across the available channels 899 in the manner

illustrated in FIG. 13 (only eight channels 899 are shown, for clarity). As illustrated in FIG. 13, a LUN 810a may comprise newly formatted pages 812 (each having an index 813) as set forth above; in this example, however, the newly formatted pages 812 are distributed across each channel 899 available to handle data transfers to and from the data store 143. It will be readily appreciated from examination of FIGS. 13A through 13H that the manner in which pages 812 are stored may be manipulated in accordance with a particular one of the channels 899 with which those pages 812 are associated. For each successive channel 899, for instance, a location of pages 812 associated with the same field type from the original data structure is incremented by an offset value, and each successive channel includes data from the next successive eight original records.

[00129] In the illustrated implementation, the order in which pages 812 are stored (*i.e.*, from record 1 through record 8) by original field type remains constant, but the first record 812 associated with a particular channel 899 increments by an offset value (for example, 1) for each successive channel 899. For example, a first channel 899 may be assigned or associated with a dataset in LUN 810a that begins with a first record or page 812 that represents a first field from the original data structure shown in FIG. 10, a second channel 899 may be assigned or associated with a dataset in LUN 810a that begins with a second record or page 812 that represents a second field from the original data structure, and so on. This is represented in FIGS. 13A through 13H, which depict a respective dataset for each respective one of eight channels 899 that are useable by LUN 810a (as indicated by the dashed lines in FIG. 13).

[00130] It is worth noting here that the actual (*i.e.*, physical) order in which pages 812 are stored in data store 143 (*e.g.*, a Flash array) need not change from channel 899 to channel 899 in many applications; for example, the same effect may be achieved logically or virtually by associating each series of pages 812 with an index or a look up table value that identifies a first page 812 for each channel 899. As long as a read operation (for example, requested by compute array 142 in cooperation with routing block 413n) is apprised of an index and an offset value for each channel 899, the specific order in which pages 812 are stored for any given channel 899 may be irrelevant. In one implementation, such indices and offset values may be maintained in node memory 450 (*e.g.*, in FP tables 454n) and communicated as necessary to routing block 413n via DMA 146b substantially as described above, though other alternatives may have similar or analogous utility. The depiction in FIG. 13 is provided by way of example only, and for purposes of visualization, but those of skill in the art will appreciate that the illustrated structure may be achieved virtually without undue experimentation. It is also noted that the data record structure

shown in FIGS. 13 and 13A through 13H may be altered or modified to accommodate the number of physical channels 899 available to a particular LUN 810n, the bandwidth or processing capabilities of compute array 142, the number of records, fields, or both in the original data structure, or a combination of these and a variety of other factors.

[00131] In particular, the diagonal striping depicted in FIG. 13 is indicative of a unique feature provided by this approach: data may be read massively parallel as records (*i.e.*, from the original data structure in FIG. 10), on the one hand, and as pure columns (*i.e.*, from the original data structure in FIG. 10), on the other hand. The type of read may be selectively adjusted or determined by identifying a page 812 selection for each channel 899. As noted above, such a read operation may be requested by compute array 142, either independently or in cooperation with routing block 413n, and may be supported by data retrieved from node memory 450, in some applications.

[00132] If a read request for all channels 899 points to the same page 812 location (*i.e.*, the number of spaces measured from the top down, or from the bottom up, in FIG. 13), then horizontally across all channels 899, this is represented by the row at reference numeral 1399. Reading all channels 899 across the entirety of row 1399 will result in retrieval of complete records for the entire dataset in LUN 810a in parallel. This method may be useful when an application or algorithm needs to check for many conditions concurrently. As noted above, the same result may be effectuated by using a page index and offset value for each respective channel 899, irrespective of location of a particular page 812 in a data structure maintained for that particular channel 899.

[00133] Additionally or alternatively, the patterned data record structure of FIG. 13 may also be effective in finding all individual records or pages 812 with a specific condition (or field type), for example, by setting an offset value in each channel 899 differently (*i.e.*, diagonally across) so that all channels 899 only deliver that specific condition or field type in parallel. For this search or read strategy, no reformatting of the data structure is required; both record-wide and column-centric selection may be equally effective in the context of effective storage bandwidth utilization.

[00134] To facilitate reassembly of related fields (from the newly formatted data structure) into complete original records after the different fields have been mapped to different Flash pages 812, each index 813 for a new page 812 may include, in addition to the information and metadata noted above, an indication of the field type (*e.g.*, a column number or other identifier from the original data structure in FIG. 10) and record (again, from the original data structure) with which a given page 812 should be associated, as well as an index number or other identifier of both the

first and last record (from the original data structure) from which the fields contained in page 812 originated. In some embodiments, compute nodes 140 may assign a local system index to each page 812, for example, as each page 812 is received or otherwise acknowledged (*e.g.*, during boot or initialization). This index may be globally unique if combined with a node identifier and a card identifier, for instance, which uniquely identify compute node 140n and its card 200 in the context of system 100.

[00135] In one simple implementation employing eight compute nodes 140 each having sixteen memory channels, it is possible to spread the various field types across all eight compute nodes 140 for a total of one hundred twenty eight parallel field types. In the absence of sophisticated mesh connectivity, however, it may be cumbersome to accommodate compound conditions across multiple compute nodes 140 in this context, and even one hundred twenty eight field types or columns may not be sufficient to reflect many real-world applications. Accordingly, it may be appropriate or desirable in some circumstances to utilize Flash pages 812 that hold, maintain, reflect, or represent multiple field types (*i.e.*, columns from the prior art data structure of FIG. 10). This is depicted in simplified form in FIGS. 14A through 14G.

[00136] Specifically, the illustrated approach may be employed in some situations to mitigate or to eliminate shortcomings generally associated with current known limitations of block RAM (BRAM) technologies implemented by many commercially available FPGA structures, for example, such as those having utility in connection with compute array 142. In applications requiring storage and/or retrieval of a vast number of datatypes (*i.e.*, fields) in parallel to facilitate examination or analysis of many conditions concurrently, conventional BRAM-based systems simply do not (and cannot, as a practical matter) comprise enough local storage to store or otherwise to accommodate many records with possibly thousands of fields at the same time.

[00137] Enabling multiple field types to be populated in a single page 812 of memory, and realizing benefits from reduced bandwidth overhead associated with partial page reads (made possible by deployment of multiple LUNs 810 in a high density constellation as illustrated and described herein), allows a compute node 140 to achieve higher throughput rates as compared to a strategy that requires reading specific entries from freshly retrieved entire pages 812 for each read request.

[00138] In some implementations, only minor adjustments to logic (*e.g.*, at routing block 413n, compute array 142, or both) or a few additional instructions (*e.g.*, in FP tables 454n at node memory 450) may be useful to enable partial page read functionality. For example, with mixed column types in a page 812, a normal page header or index 813 as described above may be

modified such that it refers to a secondary page header describing the contents of a page 812 and a boundary between field types; using this mechanism, and by apprising relevant logic or instructions sets that partial page reads are possible, fractional compound pages such as those illustrated in FIGS. 14A through 14G may be maintained without altering the above-mentioned methods for selecting pages, pre-sorting, sorting, and so forth.

[00139] Many different applications and algorithmic processing strategies may benefit from partial page reads, particularly in situations where only data from some of the subpages, or a portion thereof, are relevant for a particular computation to be executed; by using a partial page read strategy, other (irrelevant) field types in a particular page may be ignored (*i.e.*, may not be retrieved) to economize on bandwidth and computational or other processing resources. In some such embodiments, it may be desirable to implement a strategy to rotate between active LUNs 810, and their associated resources, to maximize or otherwise to increase throughput (as in the example above addressing a single field type per page).

[00140] FIG. 15 is a diagram illustrating one strategy for data communications between a host compute system and a data processing and analytics system as depicted in FIG. 1. FIG. 15 illustrates, in simplified form, one example of certain data communications that may support the foregoing functionality in the context of a data processing operation or compute task requested by a host compute system 199. Specifically, communications (calls/receipts) operations for host compute system 199 are illustrated on the left of FIG. 15, communications passing through router module 120 are illustrated in the middle of FIG. 15, and communications operations for compute node 140 are illustrated on the right of FIG. 15.

[00141] Among the other things noted above with specific reference to FIG. 1, host compute system 199 may generally comprise a shared library 1510. In some implementations, shared library 1510 may be configured and operative to implement an appropriate application programming interface (API) to enable bi-directional data communications between host compute system 199 and router module 120 as described above and as depicted in FIG. 15. In particular, an API at shared library 1510 may be embodied in or comprise a procedural interface implemented on one or more of common or popular operating systems typical for host compute system 199; examples of such an operating system include Linux™, Windows™, and Mac™ OS™, though others may be available currently, or may be developed in the future. As is generally known, an API binding typically provides basic functionality to authenticate API requests, to enable or to facilitate text encoding/decoding, and to provide generic support for read and write requests; in some implementations, the API at shared library 1510 may employ C™ and Python™

bindings, though others may also be employed. Further, the API may parse dataset descriptions and perform basic parameter validation, as is also generally known.

[00142] As set forth above, the API may also have utility in storing and retrieving datasets in a co-located manner. For example, shared library 1510 may be configured and operative (such as by the foregoing parsing of dataset descriptions, parameter validations, or other techniques) in cooperation with the API to identify specific datasets and to characterize same as related to other distinct datasets. Such a relatedness determination, which may be binary or effectuated in accordance with a weighted scale (*e.g.*, to provide a relatedness "score" or other value for a dataset relative to one or more selected other datasets) may be used to influence a physical or logical location in shared library 1510 at which data are maintained. For example, it may be beneficial in some circumstances to store related datasets in adjacent, continguous, or alternating memory blocks in shared libary 1510. The same may also be true for memory store 143n, as noted above, and so, additionally or alternatively, components at host compute system 199 may communicate datasets to compute nodes 140 (such as illustrated in FIG. 15) with such relatedness information encoded in metadata or header information, or even embedded within the datastream itself.

[00143] In some embodiments, shared library 1510 may be implemented as a binary libary (for example, on Windows, it may be in the form of a dynamic link library (DLL)) and may be provided as part of a simple software development kit (SDK) that includes appropriate header files, in a manner that is similar to support provided for any other generic C or Python library. In use, shared library 1510 may generally be apprised of, and may maintain, information related to open datasets for each individual user of host compute system 199 (*i.e.*, it may be desirable that information within shared library 1510 is not shared between individual users, but is rather shared between discrete resources of system 100).

[00144] As indicated in FIG. 15, host compute system 199 may further comprise a driver 1520. In operation, functionality of driver 1520 may be called by shared library 1510. In that regard, driver 1520 may employ standard input/output (I/O) calls, such as read and write, and may be operable in cooperation with Linux, Windows, Mac OS, or a combination of these and other operating systems (depending, for instance, upon the operating system governing functionality of host compute system 199 as a whole). Driver 1520 may translate dataset names to internal dataset identifiers, for example, as may be necessitated by host compute system 199 operating system or other resources, and may also provide full parameter validation, security and sharing management functionality, and administrative tasks as are generally required by operating

systems and other components in host compute system 199. Specifically in the context of the disclosed subject matter, driver 1520 may manage activity of compute nodes 140 (for instance, in cooperation with individual processing systems 430) by sending commands and receiving (and, optionally, aggregating) responses as necessary during run-time. In that regard, it is noted that some of the functionality of driver 1520 may be provided in cooperation with one or more of the components of processing systems 148/430, and may influence or benefit from data in node memory 144/450, described above with reference to FIGS. 1 and 4.

[00145] Driver 1520 may also maintain and organize open datasets for all registered or authorized users of host compute system 199, maintain dataset identification data or indicia for same, and enforce administration privileges or other access rights to components or modules that are fundamental to operation of host compute system 199.

[00146] Compute node 140 may generally comprise, in addition to the components described above, a firmware module 1590 or instruction set repository as indicated in FIG. 15. Firmware 1590 may be embodied in a dedicated memory component, for example, or may reside in or be maintained by an element of processing system 430 or node memory 450, in some instances. The disclosed subject matter is not intended to be limited by the manner in which firmware 1590 is stored or deployed in the context of the various hardware and software components of compute node 140. In operation, firmware 1590 may generally implement an appropriate data transfer command/response protocol sufficient to enable the communications and functionality set forth herein; in that regard, firmware 1590 may execute or accommodate an asynchronous command packet / response packet interface to allow the communications depicted in FIG. 15 and otherwise to facilitate data transfer between compute node 140 and router module 120 in accordance with the foregoing methodologies and principles. As set forth above, this feature may also comprise one or more mechanisms supporting col-location of related datasets, either independently or in cooperation with information received from host compute system 199 and associated with the data transfer.

[00147] Specifically, firmware 1590 may, in some embodiments, be configured and operative to manage data transfers between compute array 142 and data store 143 (*e.g.*, a Flash array). For example, firmware 1590 may be responsible for, manage, or otherwise facilitate some or all of the following functionality: handling boot-up initializations for compute node 140; building and managing indices or look up tables indicative of data locations for all dataset records managed by compute node 140; creating and managing datasets and any associated dataset descriptors (including co-location indications or relatedness scores across datasets); optimizing data storage

by field type per dataset; managing page 812 or block 811 write optimization protocols or strategies (including, but not limited to, co-location or dataset density approaches); managing page 812 replacement and page 812 aging strategies; loading and execution of algorithms in compute array 142 (*e.g.*, such as may be embodied in an FPGA fabric); loading relevant parameters needed or desired for algorithm execution (including whether a particular algorithm may execute on multiple, related datasets); collating and, optionally, passing data (*e.g.*, read data and/or algorithm results) between and amongst components in compute node 140; and generating dataset identifications or other indicia for dataset creation and co-location or relatedness scores, and coordinating same across all compute nodes 140 in a particular distributed processing system 100.

[00148] In the context of all datasets, firmware may maintain and organize all names, access strings, identifiers, descriptors, indices, relevant relatedness scores or co-location preferences, if applicable, and the like. For a particular dataset, firmware 1590 may also maintain a record of data pages 812 that contain dataset data and are located or that reside on or in association with a particular compute node 140. Additionally, firmware may also maintain data associated with a list of outstanding read or write requests (or both) for data that are currently being transferred. As noted above with respect to driver 1520, it will be appreciated that some of the functionality of firmware 1590 may be provided in cooperation with one or more of the components of processing system 148/430, and may influence or benefit from data stored in node memory 144/450.

[00149] Turning now specifically to the data flow illustrated in FIG. 15, a user's application program may make an API call at host compute system 199 to initiate a data processing operation. In some implementations, the API may include different routines for different functions, such as CREATE, OPEN, CLOSE, WRITE, READ, EXECUTE ALGORITHM, DELETE RECORDS, DELETE DATASET, and the like. Typically, the API expects that a dataset to be accessed will or must be specified by name and access string (password) or in accordance with some other unique identifier or identifying indicium.

[00150] In this example, shared library 1510 may field the API call and check, authenticate, verify, or otherwise confirm basic parameters. For example, if the request targets a particular dataset, then shared library 1510, either individually or in cooperation with other components of system 100, may check to ensure that host compute system 199 has that dataset open or is in the process of opening it (*e.g.*, such that the target dataset, or a relevant portion thereof, is accessible in, or via, RAM or via another short term storage medium). As another example, if the request were to create a dataset, shared library 1510, either individually or in cooperation with

other components of system 100, may parse and verify a relevant dataset descriptor or other information related to creation of such a dataset. If the request were to access dataset records, on the other hand, shared library 1510 may verify valid record numbers and ranges, again, either autonomously or in cooperation with other system 100 components.

[00151] In some implementations, shared library 1510 may be responsible for building the basis for the requested or required command, for example: the dataset may be specified by name; relevant command codes and other parameters may be set; a "write" command may be formatted to specify a number, say from 1 to n, of data records to be written; a "read," "delete record," or other command may be formatted to specify a starting record number (or an index and offset as set forth above, for instance) and a number of records to be accessed or affected; and so forth. The types of commands recognized and constructed by shared library 1510 are myriad, and may be application-specific or dynamically created, for example, as a function of the processing operation sought to be executed.

[00152] In that regard, shared library 1510 may build an "algorithm load" or "algorithm execute" command, or both, as necessary or as instructed by host compute system 199, and may retrieve, or request retrieval of, an algorithm bit stream or other pertinent instruction sets from an algorithm library or other memory storage facility maintained, for example, at host compute system 199. Upon specification of a suitable bit stream or other data and information associated with a particular algorithm to be executed, as well as selection of required or desired filters, boundary conditions, relevant variables, or a combination of these and other command parameters, shared library 1510, or a proxy, may then make a call to driver 1520, which call may include a dataset name and other relevant command parameters sufficient to initialize driver 1520 with appropriate data to begin data transfer and processing operations.

[00153] Such "I/O" calls and "Read" and "Write" data transfers are illustrated in FIG. 15 at the vertical line representing a boundary between user operations and kernel (or low level, operating system) operations in host compute system 199.

[00154] Driver 1520 may field calls from the user domain (*i.e.*, across the "I/O" call boundary illustrated in FIG. 15), such as may be embodied in or comprise a request or command string from shared library 1510. Responsive to such a command or request, driver 1520 may execute, or may ensure that some other component executes, a full validation or verification of each and every parameter in a particular command or request. For example, many commands use a dataset identifier or unique indicium or indicia for efficiency; in such instances, driver 1520 may translate a dataset name to an associated identifier for downstream use at compute node 140.

As another example involving security checks and administrative privileges, driver 1520 may maintain a record of authorized users, and thus may be apprised of which users are administrative users that have attendant administrative rights or privileges. In situations in which a particular command or request is one that requires administrative rights, driver 1520 may enforce administrative access rules as dictated or suggested by other elements of host compute system 199; alternatively, driver 1520 may ignore such rules involving regulated access to system resources in instances where a particular command or request does not implicate administrative access rules or policies.

[00155] Similarly, driver 1520 may be configured and operative to maintain data indicative of which datasets, system- or enterprise-wide, that are open and currently accessible; further, driver 1520 may also maintain data or other information related to open datasets with respect to which sharing is or may be allowed, and under what circumstances (*e.g.*, sharing may be permitted if so declared when the dataset was opened or if a dataset is related to an open dataset, for instance, or when the dataset was created), as well as which records are valid within a particular dataset (*e.g.*, including a total number of valid records and a percentage of valid records as compared to all records, valid and invalid or out of date).

[00156] If driver 1520 determines that all examined criteria and parameters are within acceptable ranges (*i.e.*, a command or request is properly constructed and formatted), then a cyclic redundancy check (CRC) may be calculated for the command packet (e.g., creating a validated or authenticated command packet), and the newly validated command packet may be transmitted to compute node 140 via host interface 126 (which may be, for example, a PCIe bus in some implementations, as noted above). This data transfer is illustrated in FIG. 15 at the vertical line (labeled “PCIe Bus”) representing a boundary between kernel operations in host compute system 199, on the one hand, and router module 120, on the other hand. Router module 120 may then forward the verified command packet to one or more compute nodes 140 (*e.g.*, via communications links 143) substantially as described above.

[00157] At a compute node 140, firmware 1590 may receive a verified command packet and may execute authentication procedures to verify the CRC for the packet; as described above with respect to driver 1520, firmware 1590 may also execute a full parameter check in the compute node 140 domain. Given valid parameters and successful CRC check, firmware 1590 may then execute the command specified by the verified command packet.

[00158] It is noted that a particular command may include zero, one, or a plurality of page read and/or write operations, and instructions for these may be parsed by firmware 1590 and transmitted to elements of compute array 142, processing system 430, or both for execution.

[00159] For algorithm load and algorithm execute commands, the algorithm bit stream provided by shared library 1510 may be loaded into compute array 142, processing system 430, or both, and executed substantially as set forth above with reference to FIGS. 1 and 4. For example, selected data records may be streamed to and from compute array 142, generally in cooperation with routing block 413n, and multi-channel algorithm results may be combined into a single result (for example, by compute array 142, routing block 413n, or both, either independently or in cooperation with processing system 430).

[00160] A response packet (illustrated at the top center of FIG. 15) generated at compute node 140 may generally describe what happened during data processing operations, and may provide, include, or reference a result (or an analysis of results) generated during computational execution of a relevant algorithm. Data payload in a response packet may include some or all of the following: a status of an ongoing processing operation (*e.g.*, percent complete, estimated time to completion, estimated compute cycles, estimated node resources that must or may be brought online, *etc.*); how many and which particular dataset record(s) are involved or may be implicated or affected by an ongoing processing operation; and what (interim) results have been obtained so far, along with an accounting or other report of such results. Most command packets generally expect a single response, but that is not true for all command packets; in some implementations in which command packets may expect or allow for multiple responses, those multiple responses may be aggregated, either at compute node 140 or at driver 1520, for example, as a function of system design choice or simply to decrease or otherwise to optimize data transfer efficiency or bandwidth utilization.

[00161] In any event, a CRC may be calculated for the whole response packet as generated by elements of compute node 140. Firmware 1590, or a surrogate or delegated component, may then transmit response packets back to host compute system 199 (*e.g.*, via router module 120, across host interface 126 bus, and to driver 1520). Upon receipt of a response packet (*e.g.*, from host interface 126), driver 1520 may then verify the CRC associated with or included in the response packet, ensure that the response matches or is coordinated with outstanding or pending command packets, and manage these and other operations in the context of a multi-access environment in which multiple compute nodes 140 are communicating with a single host compute system 199. In that regard, driver 1520 may aggregate responses, for instances, from a single

compute node 140 or from multiple compute nodes 140, and may return such responses, as well as any appropriate or attendant status information to shared library 1510. At this point in some implementations, shared library 1510 may then take control of program execution, update its data structures based upon information in received responses, and send new or updated data to a particular user's buffer or other temporary, user-specific data storage area. Program control may then be returned to a user's program or other software application that requested the data processing operations in the first place.

[00162] FIG. 16 is a functional flow diagram illustrating aspects of one implementation of a method of reformatting a data structure. As indicated at block 1601, a method of reformatting a data structure may begin by receiving a block of data, the block of data comprising a number, y, of original records wherein each original record comprises a number, x, of original fields. See, *e.g.*, FIG. 10 for an illustration of one example of such an original data record. The receiving operation depicted at block 1601 may be executed, for instance, by compute array 142 as illustrated in FIGS. 1 and 4; additionally or alternatively, this operation may be executed by, or supported by, one or more components of processing system 148/430 as described above with reference to FIGS. 1 and 4.

[00163] The method may proceed to an iterative loop as indicated at block 1602 and decision block 1620. In accordance with this iterative procedure, a number, x, of new data records may be defined, each respective new record comprising y new fields, wherein each respective new field corresponds to a respective original field from a respective original record. If a determination is made (*e.g.*, at block 1620) that additional iterations are required, the method may loop back to block 1602; otherwise, the method may proceed to block 1603. At block 1603, the block of data may be reformatted, responsive to the defining operation (at block 1602) such that a reformatted block of data comprises the new data records, each comprising a single field type. See FIG. 11 for an illustration of one example of a data structure that may result from the operations depicted at blocks 1602, 1620, and 1603. In particular, the operations depicted at these blocks are set forth in detail above with specific reference to FIGS. 10 through 14.

[00164] In this example, the method may continue by associating each of the new data records with an index operative to identify a particular new data record in the reformatted block of data. These indices are illustrated at reference numeral 813 in FIG. 11, for example. The method may then proceed by writing the reformatted block of data to memory (block 1605), whereupon the method may conclude (block 1699).

[00165] To summarize the operations depicted in FIG. 16, particularly in the context of FIGS. 1, 4, and 11, it will be appreciated that a method reformatting a data structure may result in a newly formatted data structure that enables flexible and efficient processing operations and data access in cooperation with conventional memory hardware architectures; for example, vector processing or traditional processing may be selectively implemented with minimal administrative overhead.

[00166] It is noted that the arrangement of the blocks and the order of operations depicted in FIG. 16 are not intended to exclude other alternatives or options. For example, the operations depicted at blocks 1603 and 1604 may be reversed in order, or they may be made to occur substantially simultaneously in some implementations. Further, one or more of these operations may occur substantially simultaneously with the operations depicted at blocks 1602 and 1605 in instances where it may be desirable to do so, *e.g.*, for efficiency, where processing resources are sufficient, when multithreaded or parallel operations are accommodated by the compute array 142 and attendant hardware, and the like. Those of skill in the art will appreciate that the foregoing subject matter is susceptible of various design choices that may influence the order or arrangement of the operations depicted in FIG. 16.

[00167] FIG. 17 is a functional flow diagram illustrating aspects of one implementation of a method of executing data processing operations. As indicated at block 1701, a method of executing data processing operations may begin by providing a compute node communicatively coupled to a host computer and operative to manage data processing operations independent of the host computer. Such a compute node may be embodied in or comprise elements described above with reference to compute node 140; similarly, the communicative coupling may be embodied in or comprise elements of router module 120 such as described in the examples above.

[00168] If a determination is made (*e.g.*, at decision block 1720) that additional nodes are required or desired for a particular data processing operation or application, then the method may loop back to block 1701; otherwise, the method may proceed to block 1702.

[00169] The method may continue by providing a programmable logic component in each compute node as indicated at block 1702; each programmable logic component may be configured and operative to execute data processing operations in cooperation with a first memory component. In some disclosed embodiments, the programmable logic component (e.g., compute array 142 in FIGS. 1 and 4) may be embodied in or comprise a microprocessor, a multicore microprocessor, an FPGA, or other digital data processing component suitable to execute the

functionality set forth above. Also as noted above, the first memory component may be, or have access to, a non-volatile data storage device such as EEPROM or Flash memory, though other types of memory are contemplated.

[00170] In some disclosed implementations, the method may proceed by providing a data mover component in the compute node that is configured and operative to facilitate data communications between the programmable logic component and a second memory component (block 1703). The second memory component may include any of various types of solid state storage hardware, such as RAM, SRAM, DRAM, DDR RAM, DDR4 RAM, and the like. The data mover component may utilize hardware and firmware components that are operable substantially as set forth above with reference to FIGS. 1 and 4.

[00171] The method may proceed, as indicated at block 1704, by providing a set of instructions that enable the programmable logic component to reformat a block of data comprising original records, each original record comprising a plurality of fields having different field types, such that a reformatted block of data comprises new records, each new record comprising a plurality of fields of a single field type from the original records. Aspects of an example of such reformatting operations are described above with reference to FIGS. 10 through 14.

[00172] The method may then proceed by utilizing a plurality of communications channels to transfer data associated with the reformatted data block between the programmable logic component and the first memory component (block 1705), whereupon the method may conclude (block 1799).

[00173] As with FIG. 16, the arrangement of the blocks and the order of operations depicted in FIG. 17 are not intended to exclude other alternatives or options. For example, it will be appreciated that in accordance with one embodiment, the operations depicted at blocks 1702, 1703, and 1704 may be executed in any order without materially impacting results of the method or the functionality of any particular hardware implementation utilized to execute the method. In addition to the alternatives set forth in detail above, various design choices that may influence the order or arrangement of the operations depicted in FIG. 17 will be readily apparent to those of skill in the art.

[00174] FIGS. 18 and 19 are functional block diagrams illustrating components of various implementations of a compute node having utility in connection with a data processing and analytics system. It will be appreciated that the architectures illustrated in FIGS. 18 and 19 may be implemented as variations of compute node 140, and that node 1800 may be similar in overall

architecture to compute node 140 depicted in, and described in detail with specific reference to, FIGS. 1 and 4. As illustrated in FIG. 19, node 1900 is similar to node 1800, though the former may omit a data store component (reference numeral 143 in FIGS. 1 and 1800) that is included in or accessible by the latter.

[00175] As set forth above with reference to FIG. 1, a communications link (not illustrated in FIGS. 18 and 19) may be employed to couple compute nodes such as 140, 1800, and 1900 at different successive tiers in series. As noted above, such a communications link (reference numeral 149 in FIGS. 1 and 4) may be based upon and operative in accordance with standard serial interfaces or protocols, though some proprietary serial links may also be used as a design choice. For instance, it may be desirable that a communications link employ or comprise a Xilinx Aurora™ communication link. In any event, such a communications link may be selected, configured, and operative as a function of, or otherwise taking into consideration, the hardware and operation of a compute array 142 (or "algorithm framework"). Those of skill in the art will appreciate that many options exist for both compute array 142 and a communications link, and that specific commercial embodiments have been identified by way of example only, and not by way of limitation.

[00176] The "Flash Mass Storage" functional block illustrated in FIG. 18 may be embodied in or comprise the data store 143 described above with reference to FIG. 1. In that regard, data store 143 may be embodied in or include Flash memory or other EEPROM or non-volatile storage medium. A suitable interface, such as data store interface 145 in FIG. 1 (not shown in FIG. 18), may employ a protocol that is necessary or desirable to allow compute array 142 and algorithm program logic 1890 to communicate with (*e.g.*, manage read and write access to) data store 143 as is generally known in the art.

[00177] The "Firmware Workspace" components illustrated in FIGS. 18 and 19 may be identical, similar, or analogous to node memory 144 described above with specific reference to FIGS. 1 and 4. In that regard, node memory 144 may be implemented as or comprise DDR4 RAM or other volatile or non-volatile memory suitable to support operations of node 1800 or 1900, in general, and processing system 148, in particular. As noted above, node memory 144 may be employed to support algorithm execution in some compute-intensive situations. It may be desirable or useful in certain circumstances that node memory 144 comprise, or at least have access to, dedicated or predetermined addresses or blocks of memory (such as algorithm memory 1841) that may be used, *e.g.*, in an "on demand" or "ad hoc" manner, as processing loads or bandwidth requirements dictate as set forth below.

[00178] In that regard, algorithm memory 1841 may be implemented as pre-allocated addresses, blocks, or sectors of node memory 144, for instance, or it may be dynamically allocated on an "as available" basis as processing requirements vary over time. The latter approach may not reserve or pre-allocate large chunks of node memory 144 when it is not needed, but may also ensure that sufficient algorithm memory 1841 may be available to support complex computational tasks. In the illustrated examples, algorithm memory 1841 may generally comprise resource-specific data storage cells or memory locations (such as the FP tables 454n, algorithm buffers 456n, and FP buffers 458n illustrated in FIG. 4), and may generally be operative to store data and instructions necessary or desirable for processing system 148 to perform write/read/execute operations in support of programmable logic block 410 as is generally known.

[00179] Those of skill in the art will appreciate that the present disclosure is not intended to be limited to any particular memory management techniques that may enable or facilitate allocation of algorithm memory 1841 in the context of using node memory 144 in connection with particular drivers, memory controllers, or address allocation methodologies.

[00180] In the illustrated architectures, processing system 148 (which may be similar or analogous to processing system 148 of FIGS. 1 and 4) may generally handle read/write commands and other memory access with respect to data store 143 (*e.g.*, via a suitable data store interface), receive and act upon interrupt requests (*e.g.*, with respect to operations executing at compute array 142), and regulate, control, or otherwise influence intra-node data transfers across the various components of nodes 1800 and 1900. For instance, in one embodiment described above, processing system 148 may comprise a set of one or more data processing elements or hardware components to manage all interrupt handling as well as AXI or other intra-node data transfers, on the one hand, and a different set of one or more processing components to manage general system administration (or "house-keeping") and management tasks for a node 1800, 1900 as well as algorithm execution support for compute array 142, on the other hand. While aspects of this implementation are described above with reference to FIG. 4, FIGS. 18 and 19 illustrate that different component arrangements and task distributions are within the scope and contemplation of the disclosed subject matter.

[00181] As depicted in FIGS. 18 and 19, processing system 148 may comprise processors 439 (or processing "cores") such as described above as well as additional processing cores 1871 that may be used for algorithm processing or execution support as set forth in more detail below. In that regard, processors 1871 may be analogous to (or implemented in addition to) processors 439c and 439d in FIG. 4, and may generally be embodied in or comprise digital processing

hardware and firmware that is similar to or identical to that employed in processors 439, though it may be desirable in some instances to use different hardware having different processing capabilities or operational characteristics for processors 1871 as compared to processors 439, particularly in the case where processors 1871 are dedicated to algorithm execution support (as opposed to a situation where a generic core may be used for algorithm execution only on a part-time or as needed basis). It is noted that bus structures, power sources, controllers, internal memory, and other attendant and supporting hardware and communications paths have been omitted from FIGS. 18 and 19 for clarity. Processors 1871 may be implemented as or generally comprise a microprocessor, a multi-core microprocessor, a microcontroller, an FPGA, a PLC, a PLU, or some similar or analogous hardware component suitable for data processing as is generally known in the art and described above with reference to processors 439.

[00182] In operation, processors 439 and 1871 may be responsible for a variety of tasks which, under some circumstances, may be requested or required by processors 432a and 432b (see FIG. 4), or which may be initiated autonomously or otherwise without direction from external components. In various implementations, processors 439 and 1871 may perform some or all of the following functions, or a subset thereof, as requirements of system 100, in general, or node 1800, 1900, dictate: data transformation and storage (*e.g.*, with respect to node memory 144, memory store 143, or both); data dispatch for write/read operations (again, with respect to node memory 144, memory store 143, or both); command interpretation and execution (*e.g.*, with respect to data processing and algorithm compilation and execution); algorithm load and compute; and memory exception handling. Those of skill in the art will appreciate that various other functions may be assigned to or attributable to processing system 148 as a function of desired operability of node 1800, 1900, and that any of numerous hardware arrangements, computing architectures, and distributed processing resources may be employed at processing system 148 to enable its functionality in the context of supporting desired capabilities of node 1800, 1900. Specifically, processors 1871 may be dedicated (either on a full-time or as needed basis) to support the algorithm execution functionality described in more detail below.

[00183] Further, and to support algorithm execution functionality as necessary or desired, processing system 148 may additionally comprise or have access to a memory component 1870. Memory component 1870 may be dedicated to facilitate processing system 148 functionality, and may be embodied in or comprise an Embedded Multimedia Card (EMMC) firmware Flash device or other EEPROM or non-volatile storage medium. By redirecting read/write operations from node memory 144 to a more local and dedicated memory component 1870 for computationally

intensive algorithm execution operations, processing system 148 may realize performance advantages in high-demand applications.

[00184] As with the FIG. 4 approach, programmable logic block 410 is illustrated in FIGS. 18 and 19 as generally comprising a compute array 142 or algorithm framework, which may in some instances be communicatively coupled to data store 143 by a suitable hardware interface (such as storage interface component 411n in FIG. 4) that is omitted from FIG. 18 for clarity. Aspects of the interoperability of programmable logic block 410 and data store 143 are described above with reference to FIG. 4, and it is noted that the FIG. 18 arrangement is not intended to be limited to any particular memory controller or communications protocol.

[00185] In the implementations of FIGS. 18 and 19, compute array 142 is depicted as further comprising or embodying an algorithm program logic functional block (or "APL," reference numeral 1890). As noted above, compute array 142 may employ or comprise any of various commercial embodiments of an FPGA that may be selected in accordance with any of a number of factors or boundaries conditions that are typical in electronics design. In this case, compute array 142 may be selected to accommodate operation of APL 1890 as described below, as well as to support other required or desired functionality of compute node 1800, 1900.

[00186] In operation, processing system 148 may cooperate with programmable logic block 410 for data transfer and read/write operations with respect to data store 143 (FIG. 18) or, more generally, to support algorithm execution at node 1800, 1900.

[00187] FIG. 20 is a functional block diagram illustrating components of an implementation of a management node having utility in connection with a data processing and analytics system. A management node 2000 such as depicted in FIG. 20 may be one implementation of router module 120 such as described above with reference to FIG. 1. In that regard, management node 2000 may generally comprise a router 122 portion, a router memory 124 portion (which, as noted above, may include one or more router memory components 124a, 124b, 124c, ... 124n), and one or more node interfaces or communications links (not illustrated, but described above with reference to FIG. 1). It is noted that attendant power supplies, bus structures, and memory controller elements are omitted from FIG. 20 for clarity, but are well-known in the art.

[00188] Management node 2000, in general, may also include a host interface 126 to communicate with a host compute system 199. As set forth above, host interface 126 may be operative in accordance with PCI or PCIe hardware and associated communications protocols such as Ethernet™, SATA, or other suitable communications technologies. The specific

hardware and communication strategy embodied by or incorporated into host interface 126 may be selected in accordance with bandwidth requirements, cooperating hardware at host compute system 199, functionality and operational characteristics of router 122 portion of management node 2000, or a combination of these and a variety of other factors.

[00189] As set forth above with reference to router module 120, one or more node interfaces or communications links may be implemented as a network interface card, a wireless or inductive hardware component, or a communications port, jack, or plug, for instance, and may enable router 122 portion of management node 2000 to communicate with a respective compute node 140, 1800, and 1900 as is generally known in the art. In use, router 122 portion may be operable to transmit data packets between host compute system 199 (via host interface 126) and a respective compute node 140, 1800, 1900 (via an appropriate node interface) in cooperation with routing information (destination address, packet header data, flood domains, *etc.*) and other short term data (look up tables, resource availability, performance or load balancing data, *etc.*) stored in or accessible by router memory 124 portion. Router memory 124 portion may be embodied in or comprise, by way of example, volatile memory such as RAM, SRAM, DRAM, DDR RAM, and the like. Router memory 124 portion, in general, or individual memory components such as 124a – 124n, in particular, may employ dedicated or dynamically allocated memory storage areas or buffers (such as result buffers 2091) to support algorithm processing in high-demand applications. As noted above, the disclosed subject matter is not to be limited to any particular memory management or resource allocation strategy, as many techniques and hardware architectures may be employed to facilitate the functionality described herein.

[00190] In one implementation, router 122 portion may execute all routing functionality (such as described above with reference to FIG. 1) using, or in cooperation with, algorithm program logic (APL) 2022 embedded in an algorithm hardware and/or firmware framework 2021. As with APL 1890 described above, APL 2022 may employ or comprise any of various commercial embodiments of an FPGA that may be selected in accordance with any of a number of factors or design choices. In this case, APL 2022 may be selected to accommodate operation of node 2000 in connection with algorithm processing as described below, in addition to support other required or desired functionality of router 122 portion set forth above. As many types of hardware implementations are generally available, the structure and operation of router 122 portion and router memory 124 portion of management node 2000 are not described further.

[00191] FIG. 21 is a functional block diagram illustrating one implementation of an algorithm processing system using a management node. FIG. 21 depicts one arrangement utilizing a matrix

(of size n X t) of distributed processing resources, wherein the matrix comprises an arbitrary number of processing pipelines (1, 2, ... n), each of which includes an arbitrary number of processing tiers (1, 2, ... t). Other arrangements, for instance, in which the pipelines do not have the same number of tiers, may also be employed in some circumstances as a design choice or as a function of computational requirements or other system parameters. In some implementations, the number of pipelines may be limited to four, eight, or sixteen, for example, depending upon the nature and operational characteristics of the nodes 140, 1800, 1900, and 2000, for instance, or in accordance with processing capabilities, power consumption, size requirements, economic, or other constraints.

[00192] As illustrated in FIG. 21, each processing pipeline (1 through n) may employ a Flash/Compute node 1800 at tier 1 (in some of the description that follows, such a Flash/Compute node 1800 may be referred to as a "memory-supported" compute node, owing to the fact that it has access to, or may incorporate, memory store 143). In some implementations, the tier 1 node 1800 in each pipeline is assigned a particular algorithm processing operation, and may use an associated memory store 143 to support such an assigned operation. To facilitate such processing, however, in addition to making use of available local resources and processing capabilities, node 1800 may employ (*e.g.*, via delegation of tasks, passing of interim results, or using alternative resources downstream in its particular pipeline) additional resources in downstream tiers (tiers 2 through t, for example). In the example illustrated in FIG. 21, each pipeline employs Compute nodes 1900 in tiers 2 through t; as noted above, each node 1900 may omit on-board or accessible memory stores 143, while still having computational capabilities sufficient to support activities delegated or requested by a respective upstream node 1800 (or node 1900, as the case may be) in its particular pipeline. It will be appreciated that a node 1900 at a particular tier in a particular pipeline may be replaced with or supplemented by a node 1800 having access to memory store 143 where desirable or appropriate for a given application or system configuration. Further, it is noted that FIG. 21 has been simplified to show only a single set of slaved, serialized Compute nodes 1900 per Flash/Compute node 1800 in a particular pipeline; this may not always be the case, as a single Flash/Compute node 1800 may support multiple, parallel pipelines of Compute nodes 1900 in some circumstances, depending upon, for example, the nature, operational characteristics, and bandwidth of the communications links 149 employed, the sophistication of compute arrays or algorithm frameworks 142, 2021 and their respective encodings, or a combination of these and a variety of other system parameters.

[00193] At the terminus of each processing pipeline, a terminal node (*i.e.*, at tier t in FIG. 21) may provide results to a management node 2000 as illustrated in FIG. 21. Though the illustrated arrangement depicts each respective pipeline feeding results to a single management node 2000, other possibilities are contemplated. Similarly, the management node 2000 at the terminus of each pipeline may be the same management node 2000 that originally assigned the processing tasks to the Flash/Compute nodes 1800 at tier 1 of the pipelines, or it may also be a different management node 2000 in other applications or hardware architectures.

[00194] FIG. 22 is a functional flow diagram illustrating aspects of one implementation of a method of executing data processing operations. As indicated in FIG. 22, one method of executing an algorithm in a distributed processing compute environment may begin by providing an initiating compute node comprising a memory store and a first node memory (see block 2201). In one such implementation, an initiating node may be embodied in or comprise a Flash/Compute node 1800 at tier 1 of a pipeline as described above with reference to FIGS. 18 and 21. This initiating compute node may engage in bi-directional data communication with a host compute system 199 substantially as set forth above.

[00195] A method may continue, as indicated a block 2202, by serially coupling one or more additional compute nodes to the initiating compute node to establish an execution pipeline comprising the initiating compute node and the one or more additional compute nodes, each respective one of the one or more additional compute nodes comprising a respective second node memory. In some implementations, the one or more additional compute nodes may not include or have access to a memory store; in such circumstances, these additional compute nodes may be embodied in or comprise Compute nodes 1900 such as those described above with reference to FIGS. 19 and 21 or similar hardware constructs.

[00196] The execution pipeline may be initiated by loading a bit stream comprising an instruction set related to execution of the algorithm into a respective programmable logic space of each respective compute node in the execution pipeline (block 2203). As noted above, loading a bit stream into each compute node responsible for execution of the algorithm may be desirable in many instances, providing efficiencies and increasing data flow through a particular pipelined architecture. In that regard, it is noted that information in the bit stream may be used in several operations depicted in FIG. 22, as well as in other methods operative in accordance with the disclosed subject matter.

[00197] For algorithm processing, a method may continue with executing a first operation associated with the algorithm at the initiating node using information in the bit stream and data

retrieved from the memory store, and passing results of the first operation to a first additional compute node in the execution pipeline (block 2204). As noted above, some implementations may execute this functionality without storing the results of the first operation in the first node memory.

[00198] As described above with specific reference to FIG. 21, the first additional compute node and each subsequent additional compute node in the execution pipeline may execute a respective additional operation associated with the algorithm using information in the bit stream and results of a respective preceding operation, and may then pass respective additional results of the respective additional operation to a next additional compute node in the execution pipeline (block 2205). As with block 2204, in some circumstances, it may be desirable that the operations depicted at block 2205 occur without storing the additional results in the respective second node memory of each respective additional compute node.

[00199] In accordance with the disclosed subject matter, a method may conclude by terminating execution of the algorithm at a management node that receives the additional results from a terminal additional compute node in the execution pipeline, the terminating comprising executing a final operation to produce a final result of the algorithm (block 2206). The management node may be embodied in or comprise a Management node 2000 substantially as described above with reference to FIGS. 20 and 21, and may execute the functionality described above.

[00200] As with FIGS. 16 and 17, the arrangement of the blocks and the order of operations depicted in FIG. 22 are not intended to exclude other alternatives or options. For example, it will be appreciated that in accordance with one embodiment, the operations depicted at blocks 2201 and 2202 may be executed substantially simultaneously, or may be integrated into a single operation. As another example, the operation depicted at block 2203 may be executed concomitantly with the operations at block 2201, 2202, or both, where compute nodes are selected in real-time or near real-time as a function of execution requirements, for example, which may be determined by the sophistication of the bit stream. Further, those of skill in the art will appreciate that the operations depicted at blocks 2204 and 2205 may occur substantially simultaneously, as a subsequent or downstream compute node is executing a next step while one or more upstream nodes are executing preceding steps in an algorithm processing operation. These and other such alternatives may readily be effectuated without materially impacting results of the method or the functionality of any particular hardware implementation utilized to execute the method. In addition to the alternatives set forth in detail above, various design choices that

may influence the order or arrangement of the operations depicted in FIG. 22 will be readily apparent to those of skill in the art.

[00201] Those of skill in the art will appreciate that the foregoing distributed processing system may be particularly suited to complex algorithm processing operations, and that the platform architecture may provide performance and efficiency advantages in many applications. In particular, the disclosed subject matter may have utility in connection with a pipeline architecture machine in accordance with which algorithm execution parameters may be programmed as processing pipelines using various facilities implemented in hardware and firmware resident at or accessible by the nodes 140, 1800, 1900, and 2000 described herein. In some implementations, for example, data may be streamed from a memory component (such as memory store 143) into an execution pipeline (such as depicted in FIG. 21) generally at the maximum clock rate of the algorithm framework 142, 2021 processing components that form or support the nodes in the pipeline.

[00202] As set forth above, an aspect of the disclosed subject matter that facilitates implementing an algorithm to execute in a particular processing node is the compute array or algorithm framework integrated into the programmable logic 410 space (see FIGS. 4 and 18-20) of the Flash/Compute nodes 140, 1800, or, when a module or card (see FIGS. 3B and 3C) is so configured, its Compute nodes 1900 or management node 2000. As noted above, nodes 1900 and 2000 may simply be implemented as Flash/Compute nodes 140, 1800 with no memory store 143 component, and the former may serve in such an arrangement as additional compute accelerators.

[00203] In operation of an algorithm processing application, most single pass algorithms may be distributed as binary bit streams to be loaded and executed in the programmable logic 410 space of one or more nodes 1800 (see FIGS. 4 and 18).More complex algorithms may also utilize other facilities or resources of a particular node or a combination of nodes (140, 1800, 1900, or 2000) such as is illustrated in the pipelines of FIG. 21. These other facilities or resources may include: programmable logic 2022 space of a management node 2000; node memory 144 at nodes 1800, 1900, router memory 124 at node 2000, or both; one of more processors 439, 1871 at a node 1800, 1900; or a combination of the foregoing or other resources.

[00204] Thus, to execute a given algorithm, a bit stream loaded into each node 1800, 1900 may also be accompanied by a bit stream for a management node 2000 that may be tasked with overseeing or managing the relevant or related pipeline (see FIG. 21) and/or executable code to be processed by a local processing system 148, 430. Whether implemented by a single node

based bit stream or by a more complex combination of node 1800, 1900 and management node 2000 facilities, all algorithms may be managed by firmware and a proprietary algorithm framework (PAF) code that is part of every bit stream and complements an algorithm's specific bit stream in the programmable logic 410, 2022 space in a given node. For example, FIG. 20 illustrates some functional blocks of a management node 2000 and its facilities, including APL 2022 integrated with or embedded in algorithm framework 2021. These facilities may be operative in accordance with the present disclosure to support the functionality described herein, as will be appreciated by those of skill in the art.

[00205] From an application programmer point of view, the complexities of this approach to algorithm processing may be completely ignored. In that regard, application programmers may simply specify the algorithm to be executed, provide parameters for the algorithm (including the name and selection criteria for the dataset upon which the algorithm is to be executed), and request execution. It is noted that an application programmer may also identify datasets that are related to the named dataset, or that the disclosed algorithm execution methodologies may identify such related datasets without input, depending upon the manner in which the datasets are identified as related in the memory store paradigm. In this manner, a single instantiation of an algorithm execution process may operate on multiple datasets efficiently and without undue programming or experimentation on the part of the application programmer. The facilities or system resources used or required by that algorithm, how nodes 140, 1800, 1900, and 2000 may interact to access data and to execute the algorithm, how much memory is required or where the appropriate memory hardware resides, and other functional and operational details may be entirely transparent to application programmers.

[00206] Some use cases are provided below.

[00207] In the context of the present disclosure, the term "Level 1" algorithm is intended to mean an algorithm that may be implemented in accordance with the present system and method using just the programmable logic 410 space of one or more Flash/Compute nodes 140, 1800 (*i.e.*, those nodes incorporating or having access to a memory store 143). These Level 1 algorithms may make use of a single processing pipeline (say, pipeline n-1 in FIG. 21), which may be replicated and executed in parallel across a plurality of nodes 140, 1800 in the pipeline. A Level 1 algorithm may be complex, large (in terms of number of execution steps), or both; a Level 1 algorithm may even be recursive, but it may generally be characterized by the fact that it has little or no need for intermediate data storage. With the foregoing as context, it will be appreciated that a Level 1 algorithm may interface with a memory store 143 and return results via the PAF

(*e.g.*, compute array 142 in FIGS. 4 and 18) under the control of firmware resident in the programmable logic 410 space.

[00208] Examples of such Level 1 algorithms are: basic mathematics and statistics functions such as min, max, average, *etc.*; complex data searches and selections; and more complex mathematical computations such as Fast Fourier Transforms and the like.

[00209] In the context of the present disclosure, the term "Level 2" algorithm is intended to mean an algorithm that generally requires one or more blocks of significant amounts of data storage for intermediate results between the various steps in an associated processing pipeline. It is noted that "significant amounts" of data storage may vary as a function of overall processing requirements, hardware capabilities, throughput or bandwidth thresholds, and other system parameters. In that regard, a typical Level 2 algorithm may be characterized as such because, in order to satisfy or to exceed some performance threshold or processing bandwidth requirement, which may be pre-determined or dynamically calculated or adjusted, the algorithm must or beneficially should: pass intermediate results to slaved nodes 1900; store those intermediate results in a temporary data store for use in connection with subsequent steps within the overall algorithm execution; or both. Specifically, a Level 2 algorithm in this context may be characterized as a series of execution layers in which each execution layer essentially operates on a Level 1 algorithm, the output (*e.g.*, a significant quantity of data) of which is necessary for operations to occur at a subsequent execution layer. In accordance with the disclosed subject matter, such Level 2 algorithms may be executed in a variety of ways.

[00210] In one implementation, for example, a Level 2 algorithm may be approached as a series of simple (*i.e.*, Level 1) algorithms stored within the programmable logic 410 space of a Flash/Compute node 140, 1800, each of which may be configured and operative to feed its output into node memory 144 (of the associated node 140, 1800); the input for each such simple algorithm at each tier in a particular pipeline may be stored results read from either the associated memory store 143 or the node memory 144 at a previous layer (*i.e.*, from the node 140, 1800 in a preceding tier in the pipeline).

[00211] As another example of implementation, a Level 2 algorithm may be approached as a series of simple algorithms stored with the programmable logic 410 space of a Flash/Compute node 140, 1800, and also stored in the programmable logic 410 space of one or more Compute nodes 1900; in this embodiment (illustrated in FIG. 21), each node 1800, 1900 may feed intermediate results or data to the next to form a pipeline. Each node 1900 in a particular pipeline may be slaved to the node 140, 1800 at tier 1, and each particular pipeline may ultimately

terminate at the APL 2022 of a management node 2000.The final result of the algorithm execution may be computed (*e.g.*, at management node 2000) and returned to a host compute system 199. In this case, instead of writing intermediate results to short-term storage (*e.g.*, at node memory 144), such results may be transmitted or handed off by the i-1th node (1800, 1900, as the case may be) to the ith node in the slaved pipeline (see FIG. 21). As the cascade of intermediate results propagates through the pipeline, the tier 1 node 1800, for example, may then accept the next data element(s) (and any additional arguments) from data store 143 (or its equivalent) while the tier 2 node 1900 continues to execute concomitantly or substantially simultaneously, ultimately passing its output to the tier 3 node 1900 in the pipeline, whereupon the tier 2 node 1900 may then be ready to accept the next input from the initiating tier 1 node 1800, and so on. The terminal node in a given pipeline (*e.g.*, the tier t node 1900 in FIG. 21) may finally pass or transmit its results to the management node 2000 for any final computation steps (*e.g.*, using APL 2022), after which a final result may be buffered for returned to host compute system 199, and each relevant tier 1 node 1800 may be notified that a result is ready to be returned.

[00212] One example of a Level 2 algorithm as contemplated herein is a common inference algorithm for a neural network executing under TensorFlow™ (an open-source platform for machine learning, artificial intelligence training, and the like). As is generally known in the art, neural networks are typically organized into layers (or tiers): all the neurons (or nodes) in a given tier are tasked with executing processing operations using both input data for that tier (and per node, as directed by the network) and a coefficient (as supplied by the network) for each node in the tier. The output data or results of the inference processing operations at each node may be quite large; typically, these data are transmitted or fed to the next subsequent tier downstream in the processing pipeline, after which they may be discarded by the node transmitting the results. The final or terminal tier's output (or a computation based upon output from all the nodes in the terminal tier) is the result of the inference, and may be returned to the application that requested the inference that instantiated the algorithm processing in the first place. In essence, then, the basic TensorFlow computational element may be considered a Level 1 algorithm that accepts primarily two inputs or arguments: a coefficient which is computed from results of original training of the particular relevant neural network; and an element of data, which is either taken from (in the case of the first layer or tier of the neural network) the dataset that is the target of the inference process, or from (in the case of a subsequent layer or tier of the neural network) output data from a previous tier in the neural network.

[00213] In the context of the present computing architecture such as illustrated in FIGS. 1 and 4, for instance, the foregoing pipelined processing methodologies may be implemented to run exclusively within a series of Flash/Compute nodes 140, 1800, for example, by storing intermediate results from processing within a given tier in node memory 144 of an applicable node 140, 1800. While this approach may achieve satisfactory performance for most applications, it may not achieve the best possible performance for computationally expensive pipelined processing since it may not, under certain circumstances, fully leverage a particular node's (or tier's) available assets. For example, since the executable code (in compute array 142 and processing system 148, for instance) must store and retrieve intermediate results from node memory 144, the streaming capabilities of the architecture may not be optimized to achieve maximum throughput, as reads from data store 143 must wait during node memory 144 access cycles. As noted above, in many processing scenarios, even in a Big Data context involving high performance computing and massive datasets, the architecture illustrated in FIGS. 1 and 4 may be adequate to provide sufficient performance and data throughput rates. For applications in which performance and data throughput are critical, however, it may be desirable to combine a Flash/Compute node 140, 1800 with a Compute node 1900, for example, such as in the architecture illustrated in FIG. 21.

[00214] As noted above with reference to FIG. 19, some embodiments of a Compute node 1900 may omit data store 143. In addition to economizing on silicon real estate, power consumption, manufacturing costs, and the like, utilization of node 1900 may enable or facilitate implementation of an overall system architecture that resembles that depicted in FIG. 21. Such a system architecture or hardware arrangement may link a single, particular Flash/Compute node 1800 (which does incorporate or have access to memory store 143) to a pipeline of serially coupled Compute nodes 1900, each of which may be uniquely associated with that particular Flash/Compute node 1800. In the FIG. 21 example, tier 1 comprises node 1800 (*i.e.*, the "initiating node"), and every successive tier (2, 3, … t) comprises a node 1900. The illustrated architecture, where appropriately coded and leveraged to increase or to maximize data throughput, may have particular utility in connection with neural network processing operations.

[00215] In the context of processing the TensorFlow inference algorithm in connection with this hardware architecture, for example, for all nodes in a particular pipeline, the same bit stream implementing the algorithm may be loaded into the programmable logic 410 space of each node 1900, in addition to the programmable logic 410 space of Flash/Compute node 140, 1800. In some implementations, there may be minor differences in the data supplied to node 1800 as

compared to those supplied to nodes 1900 in a particular pipeline. For instance, Flash/Compute node 140, 1800 may be programmed not to use its node memory 144 in connection with executing the algorithm, whereas each Compute node 1900 may be instructed or coded to do so before passing or transmitting its output to the next Compute node 1900 in the pipeline. These and any other desired operational parameters may be communicated to nodes 1800, 1900 via the bit stream.

[00216] In operation, the tier 1 initiating Flash/Compute node 1800 may retrieve input for the algorithm processing operation (*e.g.*, from memory store 143), execute required or desired computations for the first layer in the neural network, and transmit computational results or output data to the first Compute node 1900 in the pipeline. Once such computational results or output data are received from the initiating tier 1 node, the first Compute node 1900 in the pipeline (in this case, at tier 2), may begin processing operations at the next layer in the neural network; meanwhile, back at the tier 1 Flash/Compute node 1800, the next element of data to be used as input for the next algorithm processing operation may be retrieved (*e.g.*, from memory store 143) such that Flash/Compute node 1800 may already be executing the next subsequent required or desired computations or processing operations. Data processing responsibilities may propagate downstream through the pipeline, along with output from the previous tier, substantially as illustrated and described above with reference to FIG. 21.

[00217] Each Compute node 1900 in a particular pipeline may handle one or more layers of processing in the given algorithm. Eventually, upon completion of processing at a particular Compute node 1900 in a particular tier of the pipeline, that particular Compute node 1900 may then pass output to the next Compute node 1900 in the next subsequent tier. Those of skill in the art will appreciate that the number of layers for a particular algorithm that may be computed by a given Compute node 1900 in a particular pipeline may depend upon the coding provided in the bit stream, the complexity and structure of the neural network (including the overall architecture, pipeline length, specific implementations of nodes 1800, 1900, bandwidth of the communications links 149, read/write performance of node memory 144 or memory store 143, and the like), sophistication of the algorithm being processed, or a combination of these and other factors. In that regard, the number of Compute nodes 1900 employed in a given pipeline may also depend upon the complexity and structure of the neural network in the context of some or all of the factors listed above. In some instances, it may be useful to balance the ratio of layers (of algorithm processing) per Compute node 1900 with the number of Compute nodes 1900 in a given architecture in such a manner that the associated Flash/Compute node 1800 in a given pipeline

may continue to accept data streamed from memory store 143 without needing to stall the read process.

[00218] As noted above, the terminal Compute node 1900 (*e.g.*, at tier t in FIG. 21) in a particular pipeline may transmit its data to a management node 2000; in some implementations, such a management node 2000 may also be supplied with a copy of the algorithm bit stream that instantiated the processing operations in the first place. In operation, management node 2000 generally may execute the final layer (or final several layers) of computation required by the algorithm to produce a result. In some implementations, it may be desirable that a management node 2000 comprises four independent channels to router memory 124; this architecture may enable a four-way parallel computation, for example, such that a given management node 2000 may support up to four pipelines. This is illustrated in FIG. 21, which has been simplified to show only a single set of slaved, serialized Compute nodes 1900 per Flash/Compute node 1800. Other embodiments comprising multiple pipelines per Flash/Compute node 1800 are contemplated as set forth above, and may be implemented as a design choice or as a function of overall system throughput requirements.

[00219] In operation, management node 2000 may buffer computational results or algorithm output for return to host compute system 199 when appropriate or required. When resources at management node 2000 are taxed or reach a capacity threshold that may be pre-determined or dynamically computed or adjusted (*e.g.*, when result buffers 2091 are full or exceed a threshold or other pre-determined or dynamically computed limit), management node 2000 may signal the algorithm initiating nodes 140, 1800 under its control that computed results are ready for transmission. In some embodiments, one or more applicable Flash/Compute nodes 140, 1800 may initiate the return, *e.g.*, by commanding or instructing management node 2000 to transmit buffered results to host compute system 199; alternatively, management node 2000 may initiate such a data transfer in accordance with coded instructions, *e.g.*, unbidden or otherwise without instructions or requests supplied by any other system component.

[00220] The following description addresses certain hardware and functional details of some of the system components and implementations described above. In that regard, recall that FIGS. 2A through 2C are functional block diagrams illustrating one example of a hardware environment accommodating components of a data processing and analytics system. These hardware environments may be referred to as “cards” in some instances, such as in the description of FIGS. 2A through 2C; additionally, those of skill in the art will appreciate that such cards may also be referred to as “modules” in some industries or for some applications. In the description that

follows, the term "module" may be used as a synonym for "card" as that term is used in reference to FIGS. 2A through 2C.

Program Logic 410 Space

[00221] In some implementations, a single implementation of a card or module may generally comprise two types of FPGAs or processing components.

[00222] The first may be embodied in or comprise a straightforward FPGA or other processing component that handles all the functionality of a management node 2000 (FIG. 20) or router module 120 (FIG. 1) set forth above. In the illustrated implementations, there is only one such processing component per module. To accommodate the described functionality, this processing component (for example, algorithm framework 2021 in router 122 portion) may comprise or incorporate a large (in terms of overall silicon real estate or area) programmable logic space (such as APL 2022, for instance) and a large number of I/O lanes. In operation, this FPGA or processing component may generally implement appropriate protocols for messaging to/from the host compute system 199, vector messaging to/from pipelined nodes 140, 1800, and 1900, and other routing and housekeeping functionality substantially as described above.

[00223] The second is an FPGA or other processing component that may generally be implemented as a Multi-Processor System on a Chip (MPSoC). In the illustrated embodiments, every module comprises at least one of these; in some embodiments such as described above, a module or card may comprise eight of such processing components, which may operate in parallel (though other configurations are also possible, and may reflect design choices, cost constraints, desired processing characteristics, and the like). In embodiments, each such processing component includes a reasonably large programmable logic 410 space, four or more (*e.g.*, 64 bit) processors or cores 439, two or more (*e.g.*, 32 bit) processors or cores 432, and a sufficient number of I/O lanes to enable both communications with a management node 2000 via communications links 149 as well as high speed connections to node memory 144 and memory store 143 elements, if applicable. Note that some nodes 1900 may also employ the same or similar processing components, but do not need to implement a connection to a memory store 143 component.

[00224] Within each type of FPGA or processing component, an algorithm's use of the available resources while residing in the programmable logic 410 space may be managed by the algorithm framework (142 in FIGS. 18 and 19 or 2021 in FIG. 20). In accordance with some methodologies, other than restrictions imposed by the algorithm framework 142, 2021, an

algorithm may be configured and operative to use any or all available logic within the programmable logic space.

[00225] In operation, local firmware (*e.g.*, executing on one of more of the embedded processors 439, 1871) may initialize the programmable logic 410 space upon system boot or as otherwise requested, and may then load a "golden image" or template containing the appropriate or requested algorithm framework 142, 2021. Once loaded and initialized as is generally known, the local firmware may then load and execute algorithms or attendant or associated instruction sets as dictated by messages received from the host compute system 199.

Algorithm Framework 142, 2021

[00226] The algorithm framework 142, 2021 may generally provide two basic categories of services: system utilities; and a data selection engine.

[00227] In some implementations, system utilities may generally include the following: communications link 149 messaging system protocols and support; control of access (*i.e.*, driver support, controller functionality, *etc.*) to node memory 144 and memory store 143 elements, if applicable; data movers (described above) that may queue and initiate read/write activities for node memory 144 and memory store 143 elements, if applicable; start/stop/resume control for algorithm execution processes; and management of any unhandled condition (such as interrupts and the like) arising from algorithm processing operations.

[00228] In some implementations, data selection engine services may be requested or utilized in cooperation with the data movers, for instance, to implement a flexible selection capability when an algorithm operates on only a subset of data from a target dataset. For example, selections of targeted portions or subsets of data may be made using any of a variety of fields in a dataset, irrespective of whether those particular data are from the fields on which a particular algorithm will ultimately be operating. In such cases, selection of specific subsets of data may be based upon specific fields in the target dataset, fields in a different dataset that includes indices to the target dataset, both, or a combination of these and other factors or system parameters related to or arising from the applicable algorithm and processing boundary conditions. It is noted that co-location of data or datasets, as well as providing an option for a particular algorithm to execute on multiple datasets, complement these features and may result in increased computational and data access efficiencies in may applications.

[00229] In some embodiments, it may be desirable that the only permitted interaction between algorithms and the algorithm framework facilities of a node 1800, 1900 may be via the algorithm

framework 142, 2021 hardware and firmware implementation. In accordance with the disclosed subject matter, this restricted permission paradigm may be a deliberate design choice, as the algorithm framework 142, 2021 may be responsible for preventing algorithms and the constituent hardware and firmware components from corrupting system data and user data maintained in node memory 144 and memory store 143, if applicable. Other alternative embodiments may not be so restricted, however, as will be appreciated by those of skill in the art.

Node Memory 144

[00230] The majority of node memory 144 located on or accessible by a Flash/Compute node 142, 1800 may be dedicated to or otherwise used for managing access to memory store 143, as well as for storing system information required by local firmware that manages operation and overall functionality of the node 142, 1800. Even in light of the foregoing, however, several gigabytes (or more, in some implementations) of node memory 144 may be available for use by Level 2 algorithms, if needed (this is represented by reference numeral 1841 in FIG. 18). Access to such algorithm memory 1841 may be achieved via the PAF or via interaction with one or more cores 439 or 1871.

[00231] In some embodiments of a Compute node 1900, a minority or relatively small portion of local node memory 144 (typically, only a few gigabytes) may be used for system information and intra-node messaging operations required by local firmware. The remainder of the available capacity (typically, 16 gigabytes or more) of a node memory 144 storage component (*i.e.*, algorithm memory 1841) may thus be made available for Level 2 algorithm execution operations, for example: buffering intermediate results for use within the node 1900 itself; buffering a final intermediate result for transmission to the next node 1900 in the pipeline or to the management node 2000; storing data relevant to a previous or subsequent neural network processing layer; and the like.

[00232] In typical Flash/Compute node 1800 as set forth herein, an algorithm using local node memory 144 generally may slow down the streaming of data from memory store 143 into the programmable logic 410 space as required during algorithm processing operations. Such an intentional governing of data flow may be implemented as a design choice, since the data movers (described above) represent shared resources for both functions (*i.e.*, retrieving data from memory store 143 and managing intra-node data transfers with respect to node memory 144 and other components of node 1800). In some situations, this may affect throughput in a manner that is measureable or significant, which is one of the motivations to implement the architecture

depicted in FIG. 21 for high-throughput, computationally expensive applications in which it may be useful to regulate or to limit access to memory store 143.

Processing Cores

[00233] Local firmware running on either a Flash/Compute node 140, 1800, on the one hand, or on Compute node 1900, on the other hand, may generally utilize one or more processors 439, 1871 or cores (see processing system 430 in FIG. 4). As is generally known in the art, such firmware may employ multiple cores, for instance, to increase performance under normal loads, or it may throttle back and dedicate fewer cores to intra-node work in the event that a Level 2 algorithm requires the use of one or more processors 439, 1871 for high load situations.

[00234] In some implementations, it may be desirable to run a host Linux operating system in connection with any processor 432, 439, 1871 that is being used by, or that is managing, an algorithm processing operation or any application-specific code. Access to the various facilities of an applicable node 142, 1800, 1900, and 2000 may be strictly controlled by such a host operating system, and Linux is one operating system that provides certain protections for user and system data that are located in node memory 144 and memory store 143 that are not available in other commercially available operating systems. Specifically, using Linux, direct access to node memory 144 and memory store 143 elements may be strictly regulated in a manner that is not readily implemented using other operating systems. In some circumstance, the EMMC memory component 1870 integrated into or accessible by a Flash/Compute node 1800 and a Compute node 1900 may be utilized for maintaining data and instruction sets associated with the host operating system, for instance, and to act as a "hard drive" for the local node 1800, 1900 when needed or desired. Such activity may be limited to a specific partition on the memory component 1870 if desired, as is generally known in the art.

Management Node 2000 Facilities

[00235] A management node 2000, as previous described, may have a distinct architecture and specific facilities that are available to accommodate processing of Level 2 algorithms in some situations. In some implementations, for instance, a management node 2000 may include or have access to up to four times (or more) the local memory capacity (*e.g.*, router memory 124) of other nodes (such as nodes 140, 1800, and 1900) and no memory store 143 component; additionally or alternatively, it may also not have any local processors or processing cores.

[00236] In the illustrated embodiments (provided by way of example only) a management node 2000 may generally be controlled by its own local version of the algorithm framework 2021, which

may include extensions or appropriate modifications suitable to enable bi-directional data communication with the host compute system 199 (as set forth above with reference to FIG. 1) and any necessary functional blocks to implement necessary PCIe or other communications protocols (or other applicable protocols for interfacing with any other communication fabric that is generally known in the art or developed in accordance with known principles).

[00237] In the context of algorithm processing operations utilizing the programmable logic 2022 space of a management node 2000, for example, a single FPGA or other processing component resident at or integrated into the management node 2000 may be made available via algorithm framework 2021, and may be loaded and made available in the event that a Level 2 algorithm, which may require use of its resources, is selected for execution.

[00238] In the illustrated arrangement, router memory 124 integrated into a management node 2000 may be implemented as four or more separate DRAM modules (reference numerals 124a through 124n in FIGS. 1 and 20), though other configurations may have utility in some situations. In one implementation, each router memory 124 module may be sixteen gigabytes in capacity, and may be coupled to the local algorithm framework 2021 via a dedicated or individual direct memory access (DMA) or other channel. In that regard, a particular management node 2000 may alternatively function effectively as four (or more) virtual Compute nodes 1900 that each have programmable logic 2022 space only. In addition, the memory usage requirements for a given management node 2000 (in terms of overall system operation) may be generally low, so the vast majority of the total memory capacity resident at a management node 2000 may be made available to an algorithm executing in its programmable logic 2022 space.

[00239] A management node 2000 may also be capable of communicating directly with any and all other nodes 140, 1800, and 1900 connected to it via a pipeline or other distributed architecture hardware construct. These communications facilities may generally be used at any point during algorithm execution, though it is most likely that inter-node communications involving a management node 2000 will be related to or associated with a final or terminal operation at the end of execution or when data buffers or other temporary data storage components (such as result buffers 2091) reach some pre-determined capacity, at which point it may be necessary or desirable to initiate a return of data to the host compute system 199.

[00240] Several features and aspects of a system and method have been illustrated and described in detail with reference to particular embodiments by way of example only, and not by way of limitation. Those of skill in the art will appreciate that alternative implementations and various modifications to the disclosed subject matter are within the scope and contemplation of

the present disclosure. Therefore, it is intended that the present disclosure be considered as limited only by the scope of the appended claims.

What is claimed is:

CLAIMS

1. A data processing system operative in cooperation with a host compute system, the data processing system comprising:

a router module comprising a host interface and a node interface, wherein the router module is communicatively coupled to the host compute system via the host interface; and

multiple parallel first tier compute nodes coupled to one or more additional compute nodes in series, each compute node comprising:

a communications link, wherein each first tier compute node is communicatively coupled to the node interface at the router module via the communications link, at different successive tiers, each compute node in a series being communicatively coupled, via its communications link to one or more additional compute nodes connected at different successive tiers;

a data store comprising records associated with a data processing operation;

a programmable logic component to execute the data processing operation in cooperation with the data store;

a node memory comprising data and instructions to support operation of the programmable logic component in connection with the data processing operation and to facilitate data communications via the communications link;

a data mover component to facilitate data communications between the programmable logic component and the node memory; and

a storage interface component to facilitate data communications between the programmable logic component and the data store;

wherein the storage interface component utilizes a plurality of communications channels to transfer data between the programmable logic component and the data store.

2. The data processing system of claim 1 wherein the communications link employs a serial communications protocol.

3. The data processing system of claim 1 wherein the data store comprises an Electronically Erasable Programmable Read Only Memory device.

4. The data processing system of claim 3 wherein the storage interface component employs and Open NAND Flash Interface protocol.

5. The data processing system of claim 1 wherein the programmable logic component comprises a field programmable gate array.

6. The data processing system of claim 1 further comprising a processor to manage operation of the data mover component.

7. The data processing system of claim 6 wherein the data mover component comprises a device controller.

8. The data processing system of claim 1 wherein the host compute system executes instructions to influence operation of the compute node.

9. The data processing system of claim 1 wherein the router module and the compute node are integrated into an interface card communicatively coupled to the host compute system.

ABSTRACT

A system and method employing a distributed hardware architecture, either independently or in cooperation with an attendant data structure, in connection with various data processing strategies and data analytics implementations are disclosed. A compute node may be implemented independent of a host compute system to manage and to execute data processing operations. Additionally, an unique algorithm architecture and processing system and method are also disclosed. Different types of nodes may be implemented, either independently or in cooperation with an attendant data structure, in connection with various data processing strategies and data analytics implementations.

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
18/075,576	12/06/2022		830	FERM0003CON2/00564319	9	1

CONFIRMATION NO. 1074

27799
COZEN O"CONNOR
3WTC, 175 Greenwich Street
55th Floor
NEW YORK, NY 10007

FILING RECEIPT

OC000000051993342

Date Mailed: 12/20/2022

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)

Robert Bismuth, Seattle, WA;
Mike Stengle, Seattle, WA;

Applicant(s)

Fermat International, Inc., Paris, FRANCE;

Power of Attorney: None

Domestic Priority data as claimed by applicant

This application is a CON of 17/666,049 02/07/2022
which is a CON of 16/729,727 12/30/2019 PAT 11,249,690
which claims benefit of 62/882,044 08/02/2019
and claims benefit of 62/853,764 05/29/2019
and claims benefit of 62/801,839 02/06/2019

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: No

Permission to Access Search Results: No

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: 12/19/2022

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 18/075,576**

Projected Publication Date: 03/30/2023

Non-Publication Request: No

Early Publication Request: No
**** SMALL ENTITY ****
Title

Analytics, Algorithm Architecture, and Data Processing System and Method

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: 18/075,576

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	80		N/A	
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A	350		N/A	
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	400		N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	9 minus 20 =	* 0	x =	0	OR		
INDEPENDENT CLAIMS (37 CFR 1.16(h))	1 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	830		TOTAL	

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	RATE($)	ADDITIONAL FEE($)	OR	RATE($)	ADDITIONAL FEE($)
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

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
18/075,576		Robert Bismuth	FERM0003CON200564319

CONFIRMATION NO. 1074

27799
COZEN O"CONNOR
3WTC, 175 Greenwich Street
55th Floor
NEW YORK, NY 10007

WITHDRAWAL NOTICE

OC000000052011365

Date Mailed: 12/21/2022

Letter Regarding a New Notice and/or the Status of the Application

If a new notice or Filing Receipt is enclosed, applicant may disregard the previous notice mailed on 12/20/2022. The time period for reply runs from the mail date of the new notice. Within the time period for reply, applicant is required to file a reply in compliance with the requirements set forth in the new notice to avoid abandonment of the application.

Registered users of EFS-Web may alternatively submit their reply to this notice via EFS-Web. https://sportal.uspto.gov/authenticate/AuthenticateUserLocalEPF.html

For more information about EFS-Web please call the USPTO Electronic Business Center at **1-866-217-9197** or visit our website at http://www.uspto.gov/ebc.

If the reply is not filed electronically via EFS-Web, the reply must be accompanied by a copy of the new notice.

If the Office previously granted a petition to withdraw the holding of abandonment or a petition to revive under 37 CFR 1.137, the status of the application has been returned to pending status.

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at **(571) 272-4000** or **(571) 272-4200** or **1-888-786-0101**.

/ldvan/

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
18/075,576	12/06/2022	Robert Bismuth	FERM0003CON200564319

CONFIRMATION NO. 1074

27799
COZEN O"CONNOR
3WTC, 175 Greenwich Street
55th Floor
NEW YORK, NY 10007

IMPROPER CFR REQUEST

OC000000052012003

Date Mailed: 12/21/2022

RESPONSE TO REQUEST FOR CORRECTED FILING RECEIPT

Continuity, Priority Claims, Petitions, and Non-Publication Requests

In response to your request for a corrected Filing Receipt, the Office is unable to comply with your request because:

Improper Benefit Claim(s) to Prior-Filed Provisional Application(s)

The benefit claim(s) to prior-filed provisional application(s) is improper because the instant application was not filed within twelve (12) months from the filing date of the provisional application, and there is no indication of an intermediate nonprovisional or international application designationg the United States that is directly claiming the benefit of the provisional application and filed within 12 months of the filing date of the provisional application. Applicant should review each benefit claim submitted and, if appropriate, provide the proper reference to the prior-filed applications (including an indication of any intermediate application that is directly claiming the benefit of the provisional application and filed with 12 months of the filing date of the provisional application) as required by 37 CFR 1.78.

Alternatively, if the instant application (or intermediate application) was filed within 14 months of the prior provisional application, applicant may file a petition (in the instant application or the intermediate application, as appropriate) to restore the benefit of the provisional application if the delay in filing the instant application (or intermediate application) within the 12 month time period was unintentional. See 37 CFR 1.78 for the requirements of a petition to restore the benefit of the provisional application.

For applications filed on or after September 16, 2012, applicant must submit a corrected application data sheet (ADS) to include a specific reference.

Timeliness: The required reference for each benefit claim must be filed during the pendency of the instant application and within the later of: (1) four months from the actual filing date of the instant application, or the national stage commencement date if the instant application is a national stage application under 35 U.S.C. 371; or (2) sixteen months from the filing date of the prior-filed application. Failure to timely file the required reference is considered a waiver of any benefit claim, unless a grantable petition to accept an unintentionally delayed claim under 37 CFR 1.78, the petition fee set forth in 37 CFR 1.17(m), and the required reference are filed.

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at **(571) 272-4000** or **(571) 272-4200** or **1-888-786-0101**.

/ldvan/

PATENT APPLICATION FEE DETERMINATION RECORD

Substitute for Form PTO-875

Application or Docket Number: 18/075,576

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	80		N/A	
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A	350		N/A	
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	400		N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	9 minus 20 =	* 0	x =	0	OR		
INDEPENDENT CLAIMS (37 CFR 1.16(h))	1 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	830		TOTAL	

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	RATE($)	ADDITIONAL FEE($)	OR	RATE($)	ADDITIONAL FEE($)
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

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
18/075,576	12/06/2022	Robert Bismuth	FERM0003CON200564319

CONFIRMATION NO. 1074

27799
COZEN O"CONNOR
3WTC, 175 Greenwich Street
55th Floor
NEW YORK, NY 10007

FORMALITIES LETTER

OC000000052011456

Date Mailed: 12/21/2022

NOTICE TO FILE CORRECTED APPLICATION PAPERS

Filing Date Granted

An application number and filing date have been accorded to the above-identified nonprovisional application. The application is informal since it does not comply with the regulations for the reason(s) indicated below. Applicant is given **TWO MONTHS** from the date of this Notice within which to file all required items below to avoid abandonment. Extensions of time may be obtained by filing a petition accompanied by the extension fee under the provisions of 37 CFR 1.136(a).

- **Specification Items**

  The specification, claims, and/or abstract page(s) submitted is not acceptable because:
  - The line spacing on page(s) ALL of the claims is not 1½ or double spaced (see 37 CFR 1.52(b)). A proper claim listing, having the proper line spacing throughout, is required to satisfy this requirement.
- **Drawing Items**

  The drawings submitted on 12/06/2022 are not acceptable because:
  - The application contains a section in the specification containing the brief description of the figures, but is missing the description of Figure(s) **14** as required by 37 CFR 1.74 and 37 CFR 1.77(b)(9). A proper substitute specification in compliance with 37 CFR 1.52, 1.121(b)(3), and 1.125, providing a brief description of each drawing figure presented, is required to satisfy this requirement.
  - Note: 37 CFR 1.84(u)(1) requires that each figure must be labeled with "Fig." with a consecutive Arabic numeral (1, 2, etc.) or an Arabic numeral and capital letter in the English alphabet (1A, 1B, etc.). Therefore, the substitute specification must refer to each of the properly labeled figure(s) in the brief description. Also, if applicant wishes to delete the undescribed figure, replacement drawing sheets with proper figure labeling in compliance with 37 CFR 1.84 and 1.121(d) are required.

Replies must be received in the USPTO within the set time period or must include a proper Certificate of Mailing or Transmission under 37 CFR 1.8 with a mailing or transmission date within the set time period. For more information and a suggested format, see Form PTO/SB/92 and MPEP 512.

Replies should be addressed to:

Mail Stop Missing Parts
Commissioner for Patents
P.O. Box 1450
Alexandria VA 22313-1450

REPLY VIA ELECTRONIC FILING

A reply may be submitted via electronic filing only by registered users of the appropriate USPTO patent electronic filing system and must include:

- A separate document having the document description, "**Applicant response to Pre-Exam Formalities Notice**"; and
- the item(s) for which a reply period is set forth above.

For more information about electronic filing, contact the Patent Electronic Business Center (EBC) 1-866-217-9197 or visit https://www.uspto.gov/ebc

NOTE: If the reply is not being submitted electronically, a copy of this notice must be included.

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at **(571) 272-4000** or **(571) 272-4200** or **1-888-786-0101**.

/ldvan/

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
18/075,576	12/06/2022	2181	830	FERM0003CON200564319	9	1

CONFIRMATION NO. 1074

27799
COZEN O"CONNOR
3WTC, 175 Greenwich Street
55th Floor
NEW YORK, NY 10007

FILING RECEIPT

OC000000052011454

Date Mailed: 12/21/2022

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)

Robert Bismuth, Seattle, WA;
Mike Stengle, Seattle, WA;

Applicant(s)

Fermat International, Inc., Paris, FRANCE;

Power of Attorney: None

Domestic Priority data as claimed by applicant

This application is a CON of 17/666,049 02/07/2022
and is a CON of 16/729,727 12/30/2019 PAT 11,249,690

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: No

Permission to Access Search Results: No

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: 12/19/2022

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 18/075,576**

Projected Publication Date: WDCA

Non-Publication Request: No

Early Publication Request: No
**** SMALL ENTITY ****
Title

Analytics, Algorithm Architecture, and Data Processing System and Method

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.

PTO/AIA/14 (07-14)
Approved for use through 04/30/2017. 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	FERM0003CON2/00564319
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

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
	Robert		Bismuth	

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

City	Seattle	State/Province	WA	Country of Residence	US

Mailing Address of Inventor:

Address 1	2113 West Wheeler St.
Address 2	

City	Seattle	State/Province	WA
Postal Code	98199	Country	US

Inventor 2 Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Mike		Stengle	

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

City	Seattle	State/Province		Country of Residence	WA

Mailing Address of Inventor:

Address 1	Ackerstrasse 30
Address 2	4057

City	Basel	State/Province	
Postal Code		Country	CH

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).

PTO/AIA/14 (07-14)
Approved for use through 04/30/2017. 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	FERM0003CON2/00564319
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

☐ An Address is being provided for the correspondence Information of this application.			
Customer Number	27799		
Email Address	NYIPDocketing@cozen.com	Add Email	Remove Email

Application Information:

Title of the Invention	Analytics, Algorithm Architecture, and Data Processing System and Method		
Attorney Docket Number	FERM0003CON2/00564319	**Small Entity Status Claimed**	☒
Application Type	Nonprovisional		
Subject Matter	Utility		
Total Number of Drawing Sheets (if any)	29	**Suggested Figure for Publication (if any)**	

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	27799		

PTO/AIA/14 (07-14)
Approved for use through 04/30/2017. 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	FERM0003CON2/00564319
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

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 this 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 blank.

Prior Application Status	Pending		Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
18075576	Continuation of	17666049	2022-02-07
Prior Application Status	Patented Allowed		Remove

Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
17666049	Continuation of	16729727	2019-12-30	11249690	2022-02-15

Prior Application Status	Expired		Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
16729727	Claims benefit of provisional	62882044	2019-08-02
Prior Application Status	Expired		Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
16729727	Claims benefit of provisional	62853764	2019-05-29
Prior Application Status	Expired		Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)
16729727	Claims benefit of provisional	62801839	2019-02-06

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

Foreign Priority Information:

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.

PTO/AIA/14 (07-14)
Approved for use through 04/30/2017. 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	FERM0003CON2/00564319
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

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.

Authorization to Permit Access:

☐ Authorization to Permit Access to the Instant Application by the Participating Offices

If 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 World Intellectual Property Office (WIPO), and any other intellectual property offices in which a foreign application claiming priority to the instant patent application is filed access to the instant patent application. See 37 CFR 1.14(c) and (h). This box should not be checked if the applicant does not wish the EPO, JPO, KIPO, WIPO, or other intellectual property office in which a foreign application claiming priority to the instant patent application is filed to have access to the instant patent application.

In accordance with 37 CFR 1.14(h)(3), access will be provided to a copy of the instant patent application with respect to: 1) the instant patent application-as-filed; 2) any foreign application to which the instant patent application claims priority under 35 U.S.C. 119(a)-(d) if a copy of the foreign application that satisfies the certified copy requirement of 37 CFR 1.55 has been filed in the instant patent application; and 3) any U.S. application-as-filed from which benefit is sought in the instant patent application.

In accordance with 37 CFR 1.14(c), access may be provided to information concerning the date o f filing this Authorization.

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.

PTO/AIA/14 (07-14)
Approved for use through 04/30/2017. 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	FERM0003CON2/00564319
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

Applicant 1

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	Fermat International, Inc.

Mailing Address Information For Applicant:

Address 1	2625 MIDDLEFIELD ROAD		
Address 2	#800		
City	PALO ALTO	**State/Province**	CA
Country	US	Postal Code	94306
Phone Number		Fax Number	
Email Address			

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

Assignee Information including Non-Applicant Assignee Information:

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

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.

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

PTO/AIA/14 (07-14)
Approved for use through 04/30/2017. 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	FERM0003CON2/00564319
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

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.

Signature:

NOTE: This form must be signed in accordance with 37 CFR 1.33. See 37 CFR 1.4 for signature requirements and certifications.

Signature	/Victor Castellucci/			Date (YYYY-MM-DD)	2022-02-07
First Name	Victor	Last Name	Castellucci	Registration Number	43535

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

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 C o o p eration 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.

SUBSTITUTE SPECIFICATION
Version with Markings to Show Changes Made

Analytics, Algorithm Architecture, and Data Processing System and Method

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation of, and claims priority to, co-pending United States patent application Serial No. 17/666,049, filed February 7, 2022, which is a continuation of United States patent application Serial No. 16/729,727, filed December 30, 2019, ~~and also~~which claims the benefit of the following United States provisional patent applications: Serial No. 62/801,839, filed February 6, 2019; Serial No. 62/853,764, filed May 29, 2019; and Serial No. 62/882,044, filed August 2, 2019. The disclosures of each of the foregoing applications are hereby incorporated herein by reference in their entireties.

FIELD OF THE DISCLOSURE

[0002] Aspects of the disclosed subject matter relate generally to high performance data processing, and more particularly to a system and method employing a distributed hardware architecture, either independently or in cooperation with an attendant data structure, in connection with various data processing strategies and data analytics implementations; additionally or alternatively, aspects of the disclosed subject matter relate to an unique algorithm architecture and processing system and method that may be implemented, either independently or in cooperation with an attendant data structure, in connection with various data processing strategies and data analytics implementations.

BACKGROUND

[0003] Recently, "Big Data," high performance computing, and solid state device technologies have become increasingly important in many contexts, such as in connection with machine learning and artificial intelligence projects, for instance. In accordance with conventional computing strategies, solid state devices (such as Flash™ memory and other data storage hardware media), having no moving parts or motors and having relatively short access/read/write times as compared to traditional spinning media, have been gaining popularity in cycle-intensive applications that benefit from decreased latency in memory access. On the other hand, with the explosion of data available to such systems (as a result, for example, of the nascent Internet of Things ("IoT"), distributed memory systems, and other processing paradigms involving devices sharing data with other devices), the sheer volume of available data to process is increasing faster than traditional hardware and software systems are able to evolve in order to process those data in a meaningful and efficient manner.

[0004] Further, most conventional systems designed for high throughput data processing and analytics rely upon exhaustive (or "brute force") approaches that attempt to overpower the magnitude of the challenge with overwhelming computational resources, at the expense of cycle time and power consumption. In many instances, the dataset size creates a situation in which a practical or commercially reasonable solution is impossible, or at least prohibitively expense. As a practical matter, it will be appreciated that for as long as the rate at which new data become available for processing continues to outpace the rate at which processing methodologies advance to accommodate the increased size of a given dataset, it will continue to take longer to solve bigger and more complex data processing problems.

[0005] Therefore, there is a need for an improved system and method employing a distributed hardware architecture that facilitates high throughput data processing and analytics solutions for resource-intensive applications; as set forth below, some implementations of such a distributed hardware architecture may be configured and operative to employ a data structure that departs from traditional approaches. Further, there is a need for an improved algorithm architecture and processing system and method for high throughput data processing and analytics solutions for resource-intensive applications. As with the distributed hardware architecture noted above, such an algorithm architecture and processing system may have utility in connection with a non-traditional data structure.

SUMMARY OF THE DISCLOSURE

[0006] The following presents a simplified summary of the disclosure in order to provide a basic understanding of some aspects of various embodiments disclosed herein. This summary is not an extensive overview of the disclosure. It is intended neither to identify key or critical elements of the disclosed embodiments nor to delineate the scope of those embodiments. Its sole purpose is to present some concepts of the disclosed subject matter in a simplified form as a prelude to the more detailed description that is presented later.

[0007] The present disclosure describes a system and method employing a distributed hardware architecture to manage and to execute data processing operations, either independently or in cooperation with an attendant data structure; in the context of this description, a discrete component of such a distributed hardware architecture is generally referred to as a "compute node," though other terms may be used by those of skill in the art. In some implementations, one or more compute nodes may be utilized to manage and to execute data

processing operations independent of, though communicatively coupled with, a host compute system. In addition, the present disclosure describes an algorithm architecture and processing system and method that may be implemented, either independently or in cooperation with an attendant data structure, in connection with various data processing strategies and data analytics implementations that may be distributed across or amongst various compute nodes in a distributed processing environment or paradigm.

[0008] In accordance with one aspect of the disclosed subject matter, for example, a method of reformatting a data structure may generally comprise: receiving a block of data, the block of data comprising a number, y, of original records wherein each original record comprises a number, x, of original fields; for i = 1 to x, defining a new data record, each respective new record comprising y new fields, wherein each respective new field corresponds to a respective i^{th} original field from a respective original record; reformatting the block of data responsive to the defining such that a reformatted block of data comprises the new data records, each comprising a single field type; associating each of the new data records with an index operative to identify a particular new data record in the reformatted block of data; and writing the reformatted block of data to memory.

[0009] In accordance with some implementations, a reformatted data structure may be derived from an original data structure, and the reformatted data structure may generally comprise a plurality of new data records, each of the new data records comprising a plurality of new fields associated with a particular field type defined in the original data structure such that every new field in a particular new record is of the particular field type, wherein the reformatted data structure allows a processing system selectively to read the new data records as original records from the original data structure or as original fields from the original data structure.

[0010] In some disclosed implementations, a method of executing data processing operations may generally comprise: providing a compute node communicatively coupled to a host computer and operative to manage data processing operations independent of the host computer; providing a programmable logic component in the compute node that is configured and operative to execute data processing operations in cooperation with a first memory component; providing a data mover component in the compute node that is configured and operative to facilitate data communications between the programmable logic component and a second memory component; providing a set of instructions that enable the programmable logic component to reformat a block of data comprising original records, each original record comprising a plurality of fields having different

field types, such that a reformatted block of data comprises new records, each new record comprising a plurality of fields of a single field type from the original records; and utilizing a plurality of communications channels to transfer data associated with the reformatted data block between the programmable logic component and the first memory component.

[0011] In accordance with another aspect of the disclosed subject matter, a data processing system may be operative in cooperation with a host compute system, and the data processing system may generally comprise: a router module comprising a host interface and a node interface, wherein the router module is communicatively coupled to the host compute system via the host interface; and a compute node comprising: a communications link, wherein the compute node is communicatively coupled to the node interface at the router module via the communications link; a data store comprising records associated with a data processing operation; a programmable logic component to execute the data processing operation in cooperation with the data store; a node memory comprising data and instructions to support operation of the programmable logic component in connection with the data processing operation and to facilitate data communications via the communications link; a data mover component to facilitate data communications between the programmable logic component and the node memory; and a storage interface component to facilitate data communications between the programmable logic component and the data store; wherein the storage interface component utilizes a plurality of communications channels to transfer data between the programmable logic component and the data store.

[0012] Additionally or alternatively, a data processing system operative in cooperation with a host compute system may generally comprise: a router module comprising a host interface and a node interface, wherein the router module is communicatively coupled to the host compute system via the host interface; and a compute node comprising: a communications link, wherein the compute node is communicatively coupled to the node interface at the router module via the communications link; a data store comprising datasets associated with a data processing operation; a programmable logic component to execute the data processing operation in cooperation with the data store; a node memory comprising data and instructions to support operation of the programmable logic component in connection with the data processing operation and to facilitate data communications via the communications link; a data mover component to facilitate data communications between the programmable logic component and the node memory; and a storage interface component to facilitate data communications between the programmable logic component and the data store; wherein the programmable logic component

executes a set of instructions to reformat a block of data comprising original records, each original record comprising a plurality of fields having different field types, such that a reformatted block of data comprises new records, each new record comprising a plurality of fields of a single field type from the original records; and wherein the programmable logic component causes the new records to be stored as the datasets in the data store.

[0013] In accordance with another aspect of the disclosed subject matter, a data processing system operative in cooperation with a host compute system may be summarized as generally comprising: a management node comprising a host interface and a node interface, wherein the management node is communicatively coupled to the host compute system via the host interface; a memory-supported compute node comprising: (i) a communications link, wherein the memory-supported compute node is communicatively coupled to the node interface at the management node via the communications link; (ii) a data store comprising records associated with a data processing operation; (iii) a programmable logic component to execute the data processing operation in cooperation with the data store; and (iv) a node memory comprising data and instructions to support operation of the programmable logic component in connection with the data processing operation and to facilitate data communications via the communications link; and a pipeline of one or more additional compute nodes, each of the respective one or more additional compute nodes serially connected to the memory-supported compute node via a respective communications link and comprising a respective additional node memory and a respective programmable logic component to execute a respective data processing operation in cooperation with data received via the respective communications link.

[0014] In another aspect, a data processing system operative in cooperation with a host compute system may generally comprise: a management node comprising a host interface and a node interface, wherein the management node is communicatively coupled to the host compute system via the host interface; a memory-supported compute node comprising: (i) a communications link, wherein the memory-supported compute node is communicatively coupled to the node interface at the management node via the communications link; (ii) a data store comprising records associated with a data processing operation; (iii) a programmable logic component to execute the data processing operation in cooperation with the data store; and (iv) a node memory comprising data and instructions to support operation of the programmable logic component in connection with the data processing operation and to facilitate data communications via the communications link; and a pipeline of one or more additional compute nodes, each of the

respective one or more additional compute nodes comprising: a respective communications link, wherein the respective additional compute node is communicatively coupled to an upstream compute node and a downstream compute node via the communications link; and a respective additional node memory and a respective programmable logic component, the respective additional node memory comprising data and instructions to support operation of the respective programmable logic component in connection with a respective data processing operation and to facilitate data communications via the respective communications link; wherein each respective additional compute node executes the respective data processing operation using results from its respective upstream compute node, and passes respective results to its respective downstream compute node; in some implementations, it may be desirable that each respective additional compute node operates without storing the respective results in the respective additional node memory.

[0015] In accordance with yet another aspect of the disclosed subject matter, a method of executing an algorithm in a distributed processing compute environment may generally comprise: providing an initiating compute node comprising a memory store and a first node memory; serially coupling one or more additional compute nodes to the initiating compute node to establish an execution pipeline comprising the initiating compute node and the one or more additional compute nodes, each respective one of the one or more additional compute nodes comprising a respective second node memory; initiating the execution pipeline by loading a bit stream comprising an instruction set related to execution of the algorithm into a respective programmable logic space of each respective compute node in the execution pipeline; executing a first operation associated with the algorithm at the initiating node using information in the bit stream and data retrieved from the memory store, and passing results of the first operation to a first additional compute node in the execution pipeline; wherein the first additional compute node and each subsequent additional compute node in the execution pipeline executes a respective additional operation associated with the algorithm using information in the bit stream and results of a respective preceding operation, and passes respective additional results of the respective additional operation to a next additional compute node in the execution pipeline; and terminating execution of the algorithm at a management node that receives the additional results from a terminal additional compute node in the execution pipeline, the terminating comprising executing a final operation to produce a final result of the algorithm. As with the previous example, it may be desirable in some circumstances

that each respective additional compute node operates without storing the respective results in the respective additional node memory.

[0016] The foregoing and other aspects of various disclosed embodiments will be apparent through examination of the following detailed description thereof in conjunction with the accompanying drawing figures, in which like reference numerals are used to represent like components throughout, unless otherwise noted.

DESCRIPTION OF THE DRAWING FIGURES

[0017] FIG. 1 is a functional block diagram illustrating one implementation of a data processing and analytics system for use in connection with a host compute system;

[0018] FIGS. 2A through 2C are functional block diagrams illustrating one example of a hardware environment accommodating components of a data processing and analytics system;

[0019] FIGS. 3A through 3C are functional block diagrams illustrating certain hardware components of an implementation of a data processing and analytics system such as that depicted in FIG. 1;

[0020] FIG. 4 is a functional block diagram illustrating components of an implementation of a compute node having utility in connection with a data processing and analytics system such as that depicted in FIG. 1;

[0021] FIG. 5 is a diagram illustrating one example of a data routing strategy to route data through portions of the compute node depicted in FIG. 4;

[0022] FIG. 6 is a diagram illustrating another example of a data routing strategy to route data through portions of the compute node depicted in FIG. 4;

[0023] FIG. 7 is a diagram illustrating another example of a data routing strategy to route data through portions of the compute node depicted in FIG. 4;

[0024] FIG. 8 is a diagram illustrating one implementation of a memory architecture having utility in connection with a compute node such as that depicted in FIG. 4;

[0025] FIG. 9 is a diagram illustrating one implementation of a portion of the memory architecture depicted in FIG. 8;

[0026] FIG. 10 is a diagram illustrating a conventional data record structure;

[0027] FIG. 11 is a diagram illustrating one implementation of a data record structure having utility in connection with a compute node operative in accordance with the present disclosure;

[0028] FIG. 12 is a diagram illustrating another implementation of a data record structure having utility in connection with a compute node operative in accordance with the present disclosure;

[0029] FIGS. 13 and 13A through 13H are diagrams illustrating another implementation of a data record structure having utility in connection with a compute node operative in accordance with the present disclosure;

[0030] FIGS. 14 and 14A through 14G are diagrams illustrating yet another implementation of a data record structure having utility in connection with a compute node operative in accordance with the present disclosure;

[0031] FIG. 15 is a diagram illustrating one strategy for data communications between a host compute system and a data processing and analytics system as depicted in FIG. 1;

[0032] FIG. 16 is a functional flow diagram illustrating aspects of one implementation of a method of reformatting a data structure; and

[0033] FIG. 17 is a functional flow diagram illustrating aspects of one implementation of a method of executing data processing operations.

[0034] FIGS. 18 and 19 are functional block diagrams illustrating components of various implementations of a compute node having utility in connection with a data processing and analytics system;

[0035] FIG. 20 is a functional block diagram illustrating components of an implementation of a management node having utility in connection with a data processing and analytics system;

[0036] FIG. 21 is a functional block diagram illustrating one implementation of an algorithm processing system using a management node; and

[0037] FIG. 22 is a functional flow diagram illustrating aspects of one implementation of a method of executing data processing operations.

DETAILED DESCRIPTION

[0038] Certain aspects and features of the disclosed subject matter may be further understood with reference to the following description and the appended drawing figures. In

operation, a system and method employing a distributed hardware architecture, and optionally, an attendant data structure, may have utility in connection with various data processing strategies and data analytics implementations. Specifically, the present disclosure provides for an integrated data processing platform that may generally comprise hardware and software.

[0039] As set forth in more detail below, the present disclosure addresses design and implementation of an architectural framework that may employ high performance data processing elements (compute nodes) coupled with massive data storage (a data store) in a unique configuration that minimizes or eliminates typical processor/network wait states, optimizes instruction fetch memory cycles, and analyzes data that are streamed from an attached or associated data store at the maximum rate at which the data can be accessed or streamed by the storage subsystem. The architectural framework generally contemplates one or multiple compute nodes operating in parallel (and in series, in some cases, as described below), each of which may be configured as a pipeline of computational elements that can be configured and operative to implement a variety of algorithms or other data processing operations. In the configurations provided as examples, a fully parallelized dataflow environment, which exhibits extremely high performance for analyzing massive amounts of data, is described.

[0040] The disclosed architectural arrangement may generally include a set of programmable logic blocks that may be linked in various ways to create (as well as to execute) arbitrary algorithms. This strategy greatly reduces the learning curve for using programmable processors to implement computational algorithms. For example, a typical application programmer (*e.g.*, with a limited understanding of programmable logic hardware, *per se*) may nevertheless easily use the disclosed architecture without needing to understand the intricacies of using one of more programmable logic block hardware components and how their structures or interconnections might affect or otherwise relate to algorithm requirements of the application program.

[0041] Additionally, embodiments are disclosed of a system and method of morphing (*i.e.*, reorganizing or reformatting) data located in data records; as is generally known in the art, these data are typically stored or maintained in files or datasets (or "records" or "pages"). The disclosed system and method may be implemented for the purpose of increasing the performance of analytic algorithms being applied to the data, and may optimize or maximize the rate at which data may be correctly presented to an analytic algorithm.

SUBSTITUTE SPECIFICATION
Version with Markings to Show Changes Made

[0042] It will be appreciated that the present disclosure addresses design and implementation of memory controller (such as a Flash controller) hardware and strategies capable of supporting optimized, overlapping data read and write requests from or to a memory component, while also allowing streaming of data from the same memory component for analysis by an algorithm or other data processing operation. In one implementation, scheduling of read access requests issued to memory component targets may be sequenced and overlapped in a strategic manner such that data output from the memory component is maximized, optimized, or otherwise increased as compared to conventional memory controller technologies; specifically, the disclosed systems and methods may decrease or eliminate wait times typically caused by Flash (or other memory) read delays or latency.

[0043] In accordance with the disclosed subject matter, it will be appreciated that a power management system may readily be implemented using the architecture set forth below (see, *e.g.*, FIG. 4) to implement a system of microcontroller- or microprocessor-based regulators that communicate with each other using an on-board local area network. This solution may allow power regulators to be sequenced as desired or required (for instance, by power consuming subsystems on a host hardware module), and also may allow the regulators to balance current load requirements. In some instances, this implementation may also provide a failover capability in the event that any single regulator or group of regulators fails. The result is a safe power environment that correctly starts up a module (such as a compute node or an entire interface card or other hardware component comprising or accommodating a plurality of compute nodes such as described below) without risk of damage due to uncoordinated power sequencing and also provides for a balanced, fault-tolerant running environment to facilitate or to improve reliability with respect to a particular hardware module operative in accordance with the disclosed subject matter. For instance, core voltage supplied to a processing unit in a router component as described below may be supplied via, or in cooperation with, a poly-phase, shared (*i.e.*, redundant) regulator to handle power sequencing, load balancing, and fault-tolerance; it will be appreciated that other voltage supply subsystems necessary or desirable to provide the following functionality may benefit from a similar or analogous voltage regulation strategy.

[0044] Turning now to the drawing figures, FIG. 1 is a functional block diagram illustrating one implementation of a data processing and analytics system for use in connection with a host compute system. As illustrated, one implementation of system 100 may generally comprise a host compute system 199, a router module 120, and a compute node 140.

SUBSTITUTE SPECIFICATION
Version with Markings to Show Changes Made

[0045] Host compute system 199 may generally be embodied in or comprise a computer server, a desktop or workstation computer, a laptop or portable computer or tablet, or a combination of one or more of such components. In operation, host compute system 199 may be employed to initiate, instantiate, or otherwise to request data processing operations to be executed by system 100 substantially as set forth below. In that regard, host compute system 199 may include one or more microprocessors, field programmable gate arrays (FPGAs), microcontrollers, or other digital processing apparatus, along with attendant memory, controllers, firmware, network interface hardware, and the like. For example, host compute system 199 may generally comprise multiprocessor systems, microprocessor-based or programmable consumer electronics, personal computers ("PCs"), networked PCs, minicomputers, mainframe computers, and similar or comparable apparatus for general purpose or application-specific data processing. Various implementations of host compute system 199 may be deployed in distributed computing environments in accordance with which tasks or program modules may be performed or executed by remote processing devices, which may be linked through a communications network. Those of skill in the art will appreciate that any of various computer servers, work stations, or other processing hardware components or systems of components may be suitable for implementation at host compute system 199, and that the disclosed subject matter is not limited to any particular hardware implementation or system architecture employed at host compute system 199.

[0046] With respect to requesting, initiating, enabling, facilitating, and receiving results of the data processing functionality described herein, host compute system 199 may be communicatively coupled to router module 120, enabling bi-directional data transmission in accordance with any of various communications interfaces or telecommunications protocols generally known in the art or developed and operative in accordance with known principles. Router module 120, in turn, may also be communicatively coupled to compute node 140 such that data may be transmitted or transferred, via router module 120, between host compute system 199 and compute node 140 substantially as set forth below.

[0047] Specifically, router module 120 may generally comprise a router 122, one or more router memory components 124a, 124b, 124c, ... 124n ("router memory 124") and a host interface 126; additionally, router module 120 may also include one or more node interfaces (not illustrated in FIG. 1) such as network interface cards, wireless or inductive hardware components, or communications ports, jacks, or plugs, for instance, to enable router 122 to communicate with a respective compute node 140 as is generally known in the art. In use, router 122 may be operable

to transmit data packets between host compute system 199 and a respective compute node 140 in cooperation with routing information (destination address, packet header data, flood domains, *etc.*) and other short term data (look up tables, resource availability, performance or load balancing data, *etc.*) stored in or accessible by router memory 124. In some instances, router 122 may distribute data packets and write commands to compute nodes 140 in accordance with a data-aware round robin strategy. Router memory 124 may be embodied in or comprise, by way of example, volatile memory such as random access memory (RAM) in any of its various forms, for instance, static RAM (SRAM), dynamic RAM (DRAM), double-data rate (DDR) RAM, and the like; in some applications, DDR4 RAM may be used as router memory 124. Attendant bus structures and memory controller elements are omitted from FIG. 1 for clarity, but are well-known in the art. In one implementation, router 122 may execute all routing functionality on or in cooperation with a Kintex™ KU095 FPGA platform marketed by Xilinx, Inc., but as many types of hardware implementations are generally available, the structure and operation of router 122 and router memory 124 are not described further.

[0048] In some implementations, host interface 126 may be operative in accordance with Peripheral Component Interconnect (PCI) or PCI Express (PCIe) hardware and associated communications protocols. Alternatively, other types of communications interface hardware and methodologies may be employed, such as Ethernet™ or Serial ATA (SATA). The specific hardware and communication strategy embodied by or incorporated into host interface 126 may be selected in accordance with bandwidth requirements, cooperating hardware at host compute system 199, functionality and operational characteristics of router 122, or a combination of these and a variety of other factors. The same is generally true of the hardware and communication protocol implemented at node interfaces, which may be configured and operative to communicate data packets bi-directionally between router 122 and compute nodes 140 as set forth below.

[0049] In that regard, router 122 may be communicatively coupled (via a respective node interface) to a respective compute node 140 via a communications link 149 integrated into each respective compute node 140. In the FIG. 1 illustration, only a single compute node 140 is illustrated, but an additional three in the first compute node tier are also depicted abstractly by the bi-directional arrows from router 122. It will be appreciated that each respective series of compute nodes 140 may employ a dedicated communications channel or "lane" for data transmission with router 122, such that the FIG. 1 architecture contemplates four parallel lanes, one for each of a series of compute nodes 140. In that regard, compute node 140 may be communicatively

coupled, via communications link 149, to one or more additional compute nodes 140 connected in series (*i.e.*, at different successive tiers), as illustrated at the top right side of FIG. 1.

[0050] In addition to communications link 149, in the FIG. 1 arrangement, each compute node 140 may generally comprise a compute array 142, a data store 143, a data store interface 145, node memory 144, one or more datamover apparatus (DMA) 146a and 146b ("DMAs 146"), and a processing system 148. These elements are described in detail below with specific reference to FIG. 4.

[0051] In some implementations, data store 143 may be embodied in or comprise a mass data storage component, such as a non-volatile data storage device, one example of which is an Electronically Erasable Programmable Read Only Memory (EEPROM) store. For example, data store 143 may be, or include, Flash memory, though other memory types having suitable or appropriate characteristics to facilitate the functionality set forth herein may be in use currently or developed in the future. Specifically, though all Flash is EEPROM (as a technical matter), it is noted that not all EEPROM is Flash. Typically, those of skill in the art may tend to interpret the term EEPROM as a very specific type of non-volatile memory, but the term is not so limited as employed in the context of this disclosure; examples using Flash are described below, but any non-volatile storage media and high-bandwidth interface protocol may have utility in the context of the disclosed subject matter. In situations in which data store 143 is implemented as Flash, data store interface 145 may employ an Open NAND Flash Interface (ONFI) protocol; in the case that data store 143 uses a different memory technology, data store interface 145 may be selected or modified accordingly. In operation, data store interface 145 manages access (*e.g.*, read and write access) to data store 143, as is generally known in the art.

[0052] As with router memory 124, node memory 144 may be implemented as or comprise DDR4 RAM, or other volatile or non-volatile memory suitable to support operations of compute node 140, in general, and processing system 148, in particular. In the illustrated architecture, processing system 148 may generally handle read/write commands and other memory access with respect to data store 143 (via data store interface 145), receive and act upon interrupt requests (*e.g.*, with respect to operations executing at compute array 142), and regulate, control, or otherwise influence data transfers managed by DMAs 146. For instance, in one embodiment, processing system 148 may comprise a set of one or more processing components to manage all interrupt handling as well as Advanced Extensible Interface (AXI) or other data transfers via DMAs 146, on the one hand, and a different set of one or more processing components to manage

general system administration (or "house-keeping") and management tasks for compute node 140 as well as algorithm execution support for compute array 142, on the other hand. While this implementation is described in more detail below with reference to FIG. 4, it is worth noting here that it is only one example, and that many different component arrangements and task distributions are within the scope and contemplation of the disclosed subject matter.

[0053] It is noted that router module 120 (including host interface 126) and the various compute nodes 140 described above, as well as associated busses, communications pathways, and attendant or supporting hardware infrastructure, may be incorporated into or disposed on a carrier board (such as a printed circuit board (PCB)) or interface card ("card," see reference numeral 200 in FIGS. 2A and 2B) that is configured and operative to cooperate with structural components of host compute system 199 as is generally known. Specifically, one or more cards 200 may be inserted into or otherwise engage receptacles, slots, jacks, plugs, *etc.* that are electrically and communicatively coupled to a motherboard, bus, or other interface component at host compute system 199; accordingly, system 100 may be scalable as a function of slots that are available at host compute system 199 to receive a card 200.

[0054] At a high level, the disclosed architecture of system 100 generally includes a star arrangement in accordance with which a central host interface 126 (*e.g.*, a PCIe interface) to/from host compute system 199 may be routed to four (or more) "riser" positions via router 122 (*e.g.*, an FPGA or microprocessor platform). At each riser position, a stack (*i.e.*, various tiers in FIG. 1) of compute nodes 140 may be communicatively coupled in series. One configuration may generally comprise two rows of four compute nodes 140 (tiers), whereas the FIG. 1 configuration includes four rows, each having an indeterminate number of tiers. In accordance with a PCIe implementation, each riser position or stack may employ eight discrete lanes of 8 Gb/s data links, *i.e.*, a total of 64 Gb/s, full duplex, capacity per riser position.

[0055] As noted above, communications link 149 may be employed to couple compute nodes 140 within a stack (*i.e.*, at different successive tiers) in series. In some implementations, communications link 149 may be based upon and operative in accordance with standard serial interfaces or protocols, though some proprietary serial links may also be appropriate, depending upon the functionality and operational characteristics of compute array 142, for instance. For instance, one example of compute array 142 may employ or comprise a Xilinx Zynq™ Ultrascale+™ ZU9EG FPGA, in which case, it may be desirable that communications link 149 employ or comprise a Xilinx Aurora™ communication link, as it functions well with transceivers

provided by the same manufacturer. In operation, communications link 149 may be selected, configured, and operative to provide a light-weight wrapper of the complex multi-gigabit transceivers deployed in the FPGA fabric of compute array 142, and so it may be desirable to select communications link 149 hardware as a function of, or otherwise taking into consideration, the hardware and operation of compute array 142. Those of skill in the art will appreciate that many options exist for both compute array 142 and communications link 149, and that specific commercial embodiments have been identified by way of example only, and not by way of limitation.

[0056] In connection with Big Data and other resource-intensive processing applications, it may generally be desirable to allow compute nodes 140 autonomously to handle much of the data ingest activities and distribution of processing responsibilities; accordingly, compute nodes 140 themselves may, in some cases, determine an appropriate method of round robin processing and tracking of write histories during complex data processing operations. Additionally or alternatively, it may be desirable that host compute system 199 provide directions or instructions regarding round robin processing and write operations.

[0057] For example, each compute node 140 may monitor and track, or log, its own local activity; as the level of detail or the quantity of activity to be tracked increases, however, the amount of node memory 144 required (to enable rapid access, for example) may generally increase, the latency for a given memory access operation (or for a given type or class of operations) will increase, or both. This is especially true if information or local activity sought to be tracked must be extracted from data store 143 prior to each new write transaction.

[0058] Further, communication with neighboring compute nodes 140 in the disclosed star architecture may generally become increasingly costly the further away a particular compute node 140 is from the neighboring compute node 140. Elapsed time, throughput degradation (if control or administrative paths are shared with payload data), programmable logic resource consumption, and energy consumption may all increase in proportion with or as a function of distance between respective compute nodes 140 that are to exchange information. One way to eliminate or to minimize such deleterious effects is to implement a mesh architecture (not shown), in accordance with which an “any to any” communications strategy may be integrated into the architecture of system 100. In such an embodiment, a bit of sideband control traffic may generally have minimal or negligible impact, and may be implemented effectively without having a materially adverse effect on payload data or overall performance of system 100.

SUBSTITUTE SPECIFICATION
Version with Markings to Show Changes Made

[0059] One approach to round robin processing that may readily be implemented across multiple cards 200 is to allow host compute system 199 itself to direct write commands to specific data targets or memory addresses, down to the data store 143 level, for instance. In particular, host compute system 199 may generally be apprised of all the data (*i.e.*, system-wide) to be written, and is unique in that regard (as compared to other components of system 100); accordingly, host compute system 199 may be enabled to select, compute, or otherwise to determine a proper, appropriate, or optimal granularity of distribution (*e.g.*, after how many clock cycles may the round robin advance from one compute node 140 to another). Furthermore, host compute system 199 may keep track of last writes across the entirety of the topography of system 100, and then re-start the round robin at an appropriate point (*e.g.*, after a power cycle, crash, or other interruption).

[0060] Employing host compute system 199 in the foregoing manner may not scale well in terms of performance under certain circumstances, particularly in situations where host compute system 199 is limited to local resources and not distributed across multiple processing platforms. For applications or environments requiring high scalability, an alternative approach may include enabling host compute system 199 to inform each attached or coupled card 200 of an optimal granularity of the round robin method that it is to utilize (for example, to attain maximum or some desired level of efficiency). In the case of a system employing multiple cards 200, host compute system 199 itself may implement the first tier of round robin distribution. In this example, each compute node 140 may keep track of the last channel as well as the memory address or data record (per channel) that was used for each record or column of data that is written as set forth in more detail below.

[0061] If the first stage of the round robin mechanism (*e.g.*, on the FPGA implemented at router 122) has no non-volatile storage available, it may not be possible to store the last target of distribution across power cycles. To avoid overburdening the first compute node 140 as a target in the case of many power cycles (each of which may require round robin re-initialization), the round robin logic may be initialized with a random first target in some applications. If the round robin granularity is too small and the data are distributed across multiple compute nodes 140 without filling a full page on any of them, the performance of system 100 incurs partial page write penalties on all compute nodes 140 involved.

[0062] FIGS. 2A through 2C are functional block diagrams illustrating one example of a hardware environment accommodating components of a data processing and analytics system.

SUBSTITUTE SPECIFICATION
Version with Markings to Show Changes Made

[0063] FIG. 2A depicts a top view of a carrier board (such as a PCB) or interface card ("card" 200) that may support or carry router module 120 and compute node 140 substantially as set forth above. Card 200 may comprise a typical or conventional form factor host interface 126 suitably dimensioned and configured to create electrical and/or power connections when engaged with a cooperating structure integrated into, associated with, or otherwise communicatively coupled to host compute system 199. As noted above, host interface 126 may be a PCIe, Ethernet, SATA, or other interface, and may be coupled to electrical and power connections on or integrated with card 200 via conventional PCB fabrication techniques. FIG. 2A also illustrates communications links 149a, 149b, 149c, through 149n (links 149), a respective one dedicated to each of a respective number "n" of stacks of compute nodes 140 coupled in series as set forth above. It is noted that card 200 may employ more or fewer than the four communications links 149 that are depicted in FIG. 2A.

[0064] FIG. 2B depicts a top view of a card 200 on which is disposed one example of a managing interface 202 component, and FIG. 2C depicts a top view of the managing interface 202 component of FIG. 2B. In some implementations, managing interface 202 may be embodied in or comprise router module 120 described above; in that regard, managing interface 202 may be an example of one embodiment of router module 120, such that router 122 maps to an FPGA 299 and router memory 124 components map to interface memory components 204a and 204b, each of which is depicted as having a number of discrete memory cells or memory modules 214, such as dual in-line memory modules (DIMMs) or other independently addressable memory structures. It is noted that, in this example, managing interface 202 may further comprise additional interface memory components (not shown) to correspond to the number of router memory 124 components employed in router module 120. As an alternative, router module 120 may be deployed in the FPGA 299 block in FIG. 2, and interface memory components 204a and 204b may support the functionality of router module 120 substantially as described above. Those of skill in the art will appreciate that other alternatives exist, and that the disclosed subject matter is not intended to be limited to the architecture or structural arrangement illustrated in FIGS. 2B and 2C.

[0065] FIGS. 3A through 3C are functional block diagrams illustrating certain hardware components of an implementation of a data processing and analytics system such as that depicted in FIG. 1. The left side of FIG. 3A depicts communications links 149a, 149b, 149c, through 149n (links 149) and a power coupling 399 in isolation, while the right side of FIG. 3A

depicts links 149 and power coupling 399 as they may be deployed on card 200. As noted above, each respective communications link 149 may be dedicated to each of a respective number "n" of stacks of compute nodes 140 coupled in series, and card 200 may employ more or fewer than the four communications links 149 that are depicted in FIG. 3A.

[0066] FIG. 3B is a first view of one side of a card or substrate upon which elements of compute nodes 340a, 340b, 340c, through 340n (compute nodes 340) may be disposed, and FIG. 3C is a second view of an opposite side of a card or substrate upon which other elements of compute nodes 340 may be disposed. The left side of each of FIGS. 3B and 3C illustrates a single compute node 340n in isolation, while the right side illustrates a number, "n," of such compute nodes implemented on card 200.

[0067] It is noted that compute nodes 340 may be integrated with, or disposed upon, a PCB or other carrier board, interface card, or similar structural substrate such as described above with reference to card 200. In the example of FIGS. 3B and 3C, for instance, some elements or components of compute nodes 340 may be disposed upon, attached to, or integrated with one side of such a card, whereas other components of compute nodes 340 may be disposed upon, attached to, or integrated with an opposing side. This particular structural arrangement, and appropriate manufacturing methods for effectuating same, are generally known in the art of electronics fabrication, and the disclosed subject matter is not intended to be limited to any particular fabrication techniques or process node methodologies.

[0068] Each compute node 340 may generally comprise communications link 149, a data store (depicted abstractly at reference numeral 343 in FIG. 3B), a compute array 342, and node memory 344.In some implementations, compute nodes 340 may map generally to compute nodes 140 illustrated in, and described with reference to, FIG. 1. In one example, data store 343, compute array 342, and node memory 344 may correspond, respectively, to data store 143, compute array 142, and node memory 144 described above; in such an example, compute nodes 340 may additionally comprise components that are equivalent or analogous to data store interface 145, DMAs 146, and processing system 148, though these are omitted from FIGS. 3B and 3C for clarity. In another embodiment using a different mapping, compute array 342 itself may incorporate some or all of the components illustrated and described with reference to compute node 140, in which case data store 343 and node memory 344 may be employed to support the functionality of compute array 342.

[0069] In the examples described above, system 100 may facilitate scalability, in terms of both memory or storage density as well as compute power, that is greater than conventional systems. Additionally, the foregoing modular approach may accommodate design longevity and flexibility with respect to module design and implementation (or "bring up"). In particular, data store 143/343 may be closely coupled to compute resources (such as compute array 142/342), while remaining physically separated, such as by deployment on opposing sides of an interface card 200 or PCB, which facilitates cooling and minimizes thermal effects that are typically associated with implementing mass storage in proximity to high power processing resources or system on chip (SoC) hardware.

[0070] FIG. 4 is a functional block diagram illustrating components of an implementation of a compute node having utility in connection with a data processing and analytics system such as that depicted in FIG. 1. The FIG. 4 example represents one implementation of a compute node 140n such as illustrated in FIG. 1, and generally comprises communications link 149n, a programmable logic block 410, a processing system 430, and node memory 450.

[0071] In the FIG. 4 example, compute node 140n (in tier "n" of a particular stack of compute nodes 140) may be communicatively coupled, via communications link 149n to one or more additional compute nodes 140 connected in series at different successive tiers, *i.e.*, to compute node 140n - 1 and compute node 140n + 1. As described above, communications link 149n may be based upon or operative in accordance with standard serial interfaces or protocols, or may employ proprietary serial link technologies, for example, as a function of or influenced by the functionality and operational characteristics of compute array 142.

[0072] In the FIG. 4 implementation, programmable logic block 410 is illustrated as generally comprising, in addition to compute array 142 and DMA 146a, a storage interface component 411n. As noted above, compute array 142 may employ or comprise any of various commercial embodiments of an FPGA, and many alternatives may be equally suitable or adequate, depending upon design considerations such as power consumption, thermal factors, throughput and processing capabilities, overall requirements of system 100, price constraints, or a combination of these and other boundaries conditions that are typical in electronics design. DMA 146a may be employed to handle data traffic between processing system 430 or node memory 450, on the one hand, and communications link 149n, on the other hand, as illustrated in FIG. 4. In particular, DMA 146a may be embodied in or comprise a data transfer apparatus, device controller, or other hardware or firmware module that is configured and operative to facilitate bi-directional data

communications between electronic devices, either independently or in cooperation with control signals or other instructions received from processing system 430 or specific components thereof. Any of various commercially available hardware- or firmware- supported datamover implementations may be suitable for DMA 146a, provided that a particular implementation enables data transfer operations and throughput bandwidths sufficient to support required or desired functionality of compute node 140n, in particular, and system 100, in general.

[0073] Storage interface component 411n may be one of a number, "n," of such components employed in connection with a particular programmable logic block 410, as represented by the stack on the right side of FIG. 4. While a number of similar or identical such storage interface components may be integrated with or implemented in cooperation with programmable logic block 410, only one storage interface component 411n is described; it is noted that every such storage interface component in the illustrated stack need not be identical, and that various alternatives to the specific storage interface component 411n may be suitable for a particular implementation of programmable logic 410. The functionality and specific architecture of storage interface component 411n may vary as a function of data store 143 requirements, the operational characteristics and processing capabilities of compute array 142, or a combination of these and other factors. As noted above, a particular data store 143 having utility in connection with system 100 may be embodied in or comprise a mass data storage component, such as an EEPROM store. In the FIG. 4 embodiment, data store 143 may be a Flash memory, and accordingly, data store interface 145n may employ an ONFI protocol. In operation, data store interface 145n may manage access (*e.g.*, read and write access) to data store 143, as is generally known in the art. Other types of controller technology may be substituted as a function of the particular technology employed at data store 143, for example.

[0074] Storage interface component 411n may also comprise DMA 146b. As with DMA 146a, those of skill in the art will appreciate that any of various commercially available hardware- or firmware- supported datamover implementations (such as a data transfer apparatus, device controller, or other hardware or firmware module) may be suitable for use at or in conjunction with DMA 146b. In operation, DMA 146b may facilitate bi-directional data communications between compute array 142 or processing system 430 (and node memory 450), on the one hand, and storage interface component 411n, on the other hand. DMA 146b may function independently or autonomously, for example, or in cooperation with control signals or other instructions received from processing system 430 or specific components thereof, to effectuate data transfer operations

sufficient to support operation of compute array 142 with respect to read/write or other access to data store 143.

[0075] Storage interface component 411n may also comprise a routing block 413n. Routing block 413n may be implemented as a first in / first out (FIFO) buffer or similar memory structure temporarily to store one or both of data transferred (via data store interface 145n) between compute array 142 and data store 143, and data transferred (via DMA 146b) between compute array 142 and node memory 450. Various commercial implementations of a FIFO buffer structure are available (or may be developed and operative in accordance with known principles) to provide the functionality of routing block 413n. In some implementations, routing block 413n may comprise hardware logic, or may be operative in accordance with software instruction sets, sufficient to allow routing block 413n to access or to target specific memory locations in data store 143, either independently or in cooperation with compute array 142, processing system 430, data store interface 145n, or a combination of these and other components. The disclosed subject matter is not intended to be limited by the nature, structural configuration, or operational characteristics of routing block 413n, as many alternatives may be employed as a function of design considerations and operational requirements of system 100, in general, and compute node 140n, in particular.

[0076] Processing system 430 may generally comprise a first set of one or more processors 432a and 432b and a second set of one or more processors 439a, 439b, 439c, through 439n. Bus structures, power sources, controllers, internal memory, and other attendant and supporting hardware and communications paths have been omitted from FIG. 4 for clarity. It is noted that all of the processor blocks (432a, 432b, 439a, 439b, 439c, and 439n) depicted in FIG. 4 may be implemented as or generally comprise a microprocessor, a multi-core microprocessor, a microcontroller, an FPGA, a programmable logic controller (PLC), a programmable logic unit (PLU), or some similar or analogous hardware component suitable for data processing as is generally known in the art. In one implementation, for example, one or both of processors 432a and 432b may be embodied in or comprise a Cortex™ A5™ microprocessor core marketed by ARM Holdings PLC, and some or all of processors 439a, 439b, 439c, through 439n may be embodied in or comprise a Cortex A53™ microprocessor core marketed by ARM. Other configurations (*e.g.*, the number of processors employed) and specific types of processing devices are contemplated, and may readily be implemented, as a function of, or influenced by, design goals, processing overhead, application-specific requirements for compute node 140n,

and a variety of other factors. It is also noted that the example of processing system 430 illustrated in FIG. 4 represents only one implementation of processing system 148 depicted in FIG. 1, and that the FIG. 4 embodiment may comprise more or fewer processing resources than those shown.

[0077] In operation, processors 432a and 432b may manage or otherwise influence operation of DMAs 146a and 146b, respectively. Also as depicted in FIG. 4, each of processors 432a and 432b may employ on chip memory (OCM) to facilitate interoperation with DMAs 146. In that regard, DMAs 146a and 146b may be configured and operative to transfer data between and amongst components of compute node 140n as illustrated in FIG. 4 under control of or responsive to instructions from processors 432a and 432b; some or all of such instructions may be resident in or accessible by OCM components or hardware. Additionally or alternatively, OCM may also host or maintain status updates and attendant data acquired from or associated with operation of mailbox 452n described below. In some instances, OCM may have a lower latency than external memory, such as a DDR RAM component, and by employing OCM to manage mailbox 452n updates, memory 450 bandwidth may be reserved for payload traffic, rather than administrative traffic.

[0078] In some implementations, operation of processors 432a and 432b may be influenced by interrupts received, for example, from DMAs 146a and 146b. For example, one or both of DMAs 146a and 146b may be operative to issue an interrupt following expiration of a predetermined or dynamically selected duration (*i.e.*, a "timeout"), or upon the occurrence of a predetermined or pre-established event (such as completion of a certain threshold number of transactions). In this example, one or both of processors 432a and 432b may be operative to initiate exchange of mailbox 452n status information, or to execute some other administrative or substantive function, upon receipt of such an interrupt. Additionally or alternatively, processors 432a and 432b may be configured and operative to schedule or otherwise to influence operation of processors 439a, 439b, 439c, through 439n, *e.g.*, via interprocessor interrupts or other communications protocols or instructions.

[0079] Specifically, processors 439a, 439b, 439c, through 439n may be responsible for a variety of tasks which, under some circumstances, may be requested or required by processors 432a and 432b, or which may be initiated autonomously or otherwise without direction from external components such as processors 432a and 432b. In various implementations, processors 439a, 439b, 439c, through 439n may perform some or all of the following functions, or a subset

thereof, as requirements of system 100, in general, or node 140n, dictate: data transformation and storage (*e.g.*, with respect to node memory 450, memory store 143, or both); data dispatch for write/read operations (again, with respect to node memory 450, memory store 143, or both); command interpretation and execution (*e.g.*, with respect to data processing and algorithm compilation and execution); algorithm load and compute; and memory exception handling. Those of skill in the art will appreciate that various other functions may be assigned to or attributable to processing system 430 as a function of desired operability of compute node 140, and that any of numerous hardware arrangements, computing architectures, and distributed processing resources may be employed at processing system 430 to enable its functionality in the context of supporting desired capabilities of compute node 140.

[0080] Processing system 430 may cooperate with programmable logic block 410 for data transfer and read/write operations with respect to data store 143. In particular, flexible FIFO or other buffer structures and multiplexers may be employed to provide multiple data paths to and from each memory (such as Flash) channel. Some examples of specific data paths and supporting hardware are described below with reference to FIGS. 5 through 7.

[0081] In the FIG. 4 example, node memory 450 is one representation of the node memory 144 illustrated and described above with reference to FIG. 1. In that regard, node memory 450 may be implemented as or comprise DDR4 RAM, or other volatile or non-volatile memory suitable to support operations of compute node 140n and processing system 430. In some implementations, node memory 450 may generally comprise one or more mailboxes (*i.e.*, resource-specific data storage cells or memory locations) 452n, as well as one or more functional programming (FP) tables 454n, algorithm buffers 456n, and FP buffers 458n. These components may generally be operative to store data and instructions necessary or desirable for processing system 430 to perform write/read/execute operations in support of programmable logic block 410 as is generally known. An example is provided below.

[0082] In order to minimize the need for processor 432a intervention during write transactions from host compute system 199 to compute node 140n, a largely autonomous mailbox 452n and DMA 146a strategy may handle the entire transfer (or a portion thereof) of incoming data packets to DDR4 (or similar or analogous) buffers at compute node 140n, in general, and at node memory 450, in particular. In some implementations, it may be desirable that the inbound buffers are "content agnostic" and are treated generically. In that regard, a plurality (such as one hundred

twenty eight, for instance) of mailboxes 452n may be initialized for this purpose, and their status may be set to "empty" at boot time.

[0083] In programmable logic block 410, inbound DMA 146a may also be initialized and its internal control logic may be set to "full availability." In this instance, when data packets arrive, the first available mailbox 452n address may be retrieved (*e.g.*, from a local list maintained in node memory 450), and the incoming packed may be posted to that particular mailbox 452n. The local list may then be updated to reflect that a particular mailbox 452n is occupied, and that a next mailbox 452n in series, or otherwise selected or determined in accordance with processing system logic, should be used for a next inbound data packet or series of data packets. At that point, two processes may be initiated, as noted briefly above. First, a transaction counter or other meter may be incremented (or decremented, as a matter of design choice); once a message count threshold is reached, attaining the threshold may trigger an update of the occupied mailbox list by processing system 430. Second, a timer may be set; once a predetermined period of time has elapsed (*i.e.*, a timeout event), expiration of the time period may trigger an update, just as if the message count had been reached. This second process may ensure that single or last messages will be captured, even in the event that a message or transaction counter does not reach a predetermined or minimum threshold to trigger an update in accordance with the first process. In some implementations, during such an update, programmable logic block 410 may also receive an updated list (*e.g.*, a message or instruction representative of emptied mailboxes 452n since a last transaction, or some other relevant data) as prepared by processing system 430.

[0084] With the foregoing overview of system 100 and compute node 140n architecture in mind, the following addresses one example of algorithm load and execute initiation and functionality, by way of example only, and not by way of limitation. In one implementation, initialization of each compute node 140n may begin by booting from a local golden bit stream; this initialization may generally provide a common (*i.e.*, across all compute nodes 140) infrastructure in each respective programmable logic block 410. For example, initialization may define data paths (such as those described below with reference to FIGS. 5 through 7), protocols for communications link 149n, identification and initialization of various I/O controllers, and the like – generally establishing and defining the operating environment for each compute node 140n ("compute node environment"). In one embodiment, the initial bit stream may include an algorithm block or other information concerning particular algorithm execution, though this may occur after

the initial bit stream, if necessary or desired. In some circumstances, initialization may also provide boot code or other initial parameters for the various processors in processing system 430.

[0085] Upon initialization of compute node 140n, new or updated algorithm bit streams may then be transmitted, *e.g.*, by host compute system 199, via router module 120 and communications link 149n. In some implementations, algorithm bit streams may contain both algorithm logic and the same compute node environment used in the local golden bit stream employed during initialization. When an algorithm bit stream is received from (*e.g.*, from host compute system 199), some or all of the following processes may ensue: the algorithm bit stream may be provided to all compute nodes 140 simultaneously, or substantially simultaneously or concomitantly; each respective compute node 140n may complete any outstanding data transactions; normal compute node 140n operation may be briefly paused while the new bit stream is loaded into all compute arrays 142; normal compute node 140n operations may be resumed and all compute nodes 140 may begin executing the algorithm (and continue to do so until it completes or times out); and upon completion, computational results may be returned along with status, and the algorithm bit stream may be left in place for potential re-use at a later time.

[0086] Regarding some of the data paths supporting the foregoing functionality, it is noted that FIGS. 5 through 7 are diagrams illustrating examples of data routing strategies to route data through portions of the compute node depicted in FIG. 4.In the illustrated example, each storage interface component 411n may generally comprise a group of FIFO buffers or similar structures (such as represented by routing block 413n) connected with a versatile routing fabric. During operation, the routing fabric allows for run time configuration of data flow paths through routing block 413n and (1) to/from node memory 450 or OCM buffers associated with components of processing system 430, (2) to/from data store interface 145n, (3) to/from programmable logic block 410 compute pipelines, (4) to/from programmable logic block 410 path to communications link 149n.

[0087] In some embodiments, all such data paths may be implemented with 16-bit width and follow the AXI stream protocol. In that regard, an AXI stream is generally an address-less transport mechanism that requires a minimal amount of control signals (*e.g.*, only READY, VALID and LAST). Using this example approach, data flow may begin as soon as an intended target of a data transaction indicates "READY" and a source of the data transaction indicates "VALID." Conversely, data flow may be suspended when one of the foregoing two conditions is no longer

true. The "LAST" indicator may be used by logic in the target to detect packet or data stream boundaries, for example.

[0088] In some implementations of interface routing fabric employed at routing block 413n, one or more 4:1 multiplexers may have particular utility. For instance, a 4:1 multiplexer may cooperate well with a 6:1 (four data signals, two control signals) logical unit (LUN) data structure (described below), and therefore may maximize use of the basic elements found in some viable options for FPGA structures of compute array 142. Other types of multiplexers may be employed, for instance, to match a fabric used in a commercial implementation of compute array 142, to combine certain functionality set forth below into fewer components, or as a result of a variety of other considerations. A 4:1 multiplexer also provides a reasonable multiplexing strategy, to the extent that it can devote one input section to each of the four sources noted above, which may generally cover all use cases in most practical applications and hardware configurations.

[0089] In addition to the 4:1 input section, each multiplexer output may be configured and operative to drive multiple targets simultaneously as each path may be implemented as 4:1 source select. So, in such an example, one implementation of a routing paradigm may focus on the source of the data, rather than the component to which the data are to be transmitted. This paradigm allows each source to provide data to multiple targets simultaneously (or substantially simultaneously), and it makes an illegal or conflicting configuration (*e.g.*, two or more sources trying to drive one target simultaneously) impossible.

[0090] In some implementations, some AXI stream flow control signals may be treated with some extra logic applicable to the architecture and methodologies described herein. For instance, some logic rules for AXI stream flow control may be expanded as follows: READY signals of data sinks that are connected to a common source may be ANDed (*i.e.*, combined); READY signals for destinations that not selected (*i.e.*, inactive) may be set to "1" by default; and VALID status may only be presented by a data source when it sees the ANDed READY from all possible destinations (data sinks) to prevent a single sink from initiating the dataflow. This mechanism is very flexible and resource efficient. It may generally enable a versatile interconnect with respect to the compute pipeline, in accordance with which one source value may need to feed multiple compute engines in parallel (*e.g.*, count a value "x," square "x," and multiply "x" by a different value "y"). By instantiating multiple concurrent "listening ports" (*i.e.*, compute elements) to monitor one data source, it is possible to implement an arbitrary number of concurrent calculations in

parallel, subject to the architectural and processing limitations of the hardware embodying or employed at compute array 142, for example.

[0091] FIGS. 5 through 7 show such an arrangement with two parallel FIFOs or other buffer structures, each of which is fed by an input multiplexer 511 coupled to all source interconnects, and selectively outputs to any of three paths to programmable logic block 410 for compute. The actual count of paths to programmable logic block 410 compute resources may depend, for example, upon the loaded algorithm, the corresponding compute pipeline, and a variety of other factors.

[0092] In particular, each of FIGS. 5 through 7 show four inputs to input multiplexers 511: from data store 143 (*e.g.*, "Flash Channel" via data store interface 145n); from node memory 450 (*e.g.*, via DMA 146a); a first input from the compute pipeline (*i.e.*, programmable logic block 410 compute array 142); and a second input from the compute pipeline (*i.e.*, programmable logic block 410 path to communications link 149n). These inputs are concatenated at input multiplexers 511; output from the respective FIFOs may be concatenated at output multiplexers 512.

[0093] FIG. 5 illustrates one data path from data store 143 (*e.g.*, "Flash Channel" via data store interface 145n) to node memory 450 (*e.g.*, via DMA 146a), and a separate data path from node memory 450 (*e.g.*, via DMA 146a) to data store 143 (*e.g.*, "Flash Channel" via data store interface 145n). FIG. 6 illustrates one data path from data store 143 (*e.g.*, "Flash Channel" via data store interface 145n) to one of the compute pipelines, and a separate data path from node memory 450 (*e.g.*, via DMA 146a) to a different one of the compute pipelines. FIG. 7 illustrates one data path from data store 143 (*e.g.*, "Flash Channel" via data store interface 145n) to one of the compute pipelines, and a separate data path from one of the compute pipelines to node memory 450 (*e.g.*, via DMA 146a). Those of skill in the art will appreciate that various modifications and alterations to the implementations shown in FIGS. 5 through 7 are possible, and that numerous data paths between the various components may be defined as necessary or useful for operation of a particular compute node 140n.

[0094] FIG. 8 is a diagram illustrating one implementation of a memory architecture having utility in connection with a compute node such as that depicted in FIG. 4, and FIG. 9 is a diagram illustrating one implementation of a portion of the memory architecture depicted in FIG. 8.

[0095] In some implementations, each compute node 140n may employ sixteen physical channels (or buses) for access to data store 143. In FIG. 8, these physical channels (reference

numeral 899) are depicted in vertical columns labeled, from left to right, 0, 1, . . . , 15. On each respective channel 899, one arrangement may employ one or two memory chips, each of which may comprise one or more logical units (LUNs) (reference numeral 810). In FIG. 8, LUNs associated with channel 0 are labeled 810a, LUNs associated with channel 1 are labeled 810b, and LUNs associated with channel 15 are labeled 810n. It is noted that more or fewer channels 899 may be employed, and more or fewer LUNs 810 per channel may be implemented, depending upon overall arrangement and processing functionality of a particular compute node 140n, the type of memory employed at data store 143, and a variety of other factors.

[0096] In the context of the disclosed architecture and data path approaches set forth herein, it will be appreciated that a LUN 810 may execute more than one memory access (such as a Flash access) operation concurrently, even on the same Flash (or other memory) chip. In addition to LUNs 810, each chip in the FIG. 8 arrangement may generally feature multiple planes (as indicated by the stacks of LUNs 810 – each LUN 810 in a particular stack resides at a different respective plane in the architecture). By utilizing multiple (in this case, four) planes concurrently, the available access to a given memory page, in effect, quadruples in size, thereby improving the ratio of data transfer to command bandwidth.

[0097] In the discussion that follows, LUNs 810, in particular, and data store 143, in general, are referred to in terms of Flash memory for convenience; as noted above, however, the disclosed subject matter is not intended to be limited to the particular type of memory used for data store 143, except to the extent that changing a memory type or employing a different memory technology at data store 143 may influence selection and implementation of suitable or appropriate interface mechanisms, such as at data store interface 145. In any event, references to Flash memory are by way of example only, and not by way of limitation.

[0098] In some instances, data in each LUN 810n may be stored in the form of Flash pages, one example of which may generally comprise 16Kb of data payload, plus some overhead associated with error correction code (ECC) and header information. Pages may be organized into blocks, for instance, or sets of 1024 pages. This is illustrated best in FIG. 9, which depicts a LUN 810n comprising two blocks 811 of data; more than two blocks 811 per LUN 810 are contemplated. Similarly, while each block 811 is depicted as comprising only three discrete pages 812, it may be desirable that a block 811 comprise 1024 pages, 2048 pages, or more. Each page 812, in turn, may generally comprise an index 813, which may record metadata, information, or metrics or statistics associated with the data payload of the page 812. For instance, index 813

may include ECC information and a data type indicator, a first index, a record count, an incarnation counter, various statistics (such as a minimum data value, a maximum data value, a local average value, *etc.*) and other information, or a combination of these and other types of information, with respect to data in a particular page 812. Additionally, LUN 810n may include a cache 819, which may serve as temporary storage for a page 812, or a portion thereof, for various purposes such as read operations.

[0099] For example, in order to read a given page 812, a read command is issued (for example, by compute array 142 or processing system 430, either individually or in cooperation). Such a read command may generally cause LUN 810n to copy page 812 to (on-chip) cache 819, from where it may be read rapidly across data store interface 145 and communicated to compute array 142.

[00100] As is generally known in the art, a Flash page 812 may only be written in its entirety in the context of any particular write operation; specifically, a particular Flash page 812 must be erased, in its entirety, before any new contents or data may be written to that particular page 812. This is true even if only relatively few data values (as compared to the overall contents of the particular page 812) are sought to be changed; in every case, manipulating data in a page 812 typically requires that the entire page 812 be erased and subsequently re-written with modified contents. To make matters worse, particularly in the context of Flash memory, not only must a page 812 be erased before any new value may be re-written (or altered) in that page 812, but the entire block 811 containing the affected page 812 must also be re-written.

[00101] In one implementation of LUN 810n and a method of interacting with Flash memory, erase granularity may be set as one block 811 (*e.g.*, a set of 1024 pages) as noted above; in accordance with a novel approach to an old problem, one method of updating a page 812 is to read its contents, selectively to modify whatever data are to be changed, and subsequently to write a new copy of that page 812 to a new location in memory – typically in the same block 811, or at least in the same LUN 810n, but this may not always be practical or even desired – and with an updated identifier to facilitate identification of the new copy as the definitive, updated version. As it is not economical to erase an entire block 811 (and to move all the unaltered pages 812 that may reside in the same block 811 to a new location) each time a page 812 is updated or erased, the old page 812 may remain unaltered and in its original location within the block 811.

[00102] To manage the resulting multiple copies of a page 812, each specific page 812 may be provided with an incarnation counter (or similar indicium or identifier) in its metadata; as noted above, it may be useful that this incarnation counter is updated each time a page 812 is copied during an update or modification process. In some implementations, such an incarnation number count or other identifier may be read during boot and used to populate a page pointer index (not shown) or look up table that resides in, or is accessible by, node memory 450 or processing system 430, for example. In one approach, only the copy of a particular page 812 having the highest value in the incarnation counter may be identified in the pointer index or look up table, as it is this copy that represents the active and valid copy of that page 812 – all other copies contain data that have not been updated to reflect new values. In some embodiments, the page pointer index or look up table may also be updated when a page 812 is copied and its incarnation counter is incremented. The incarnation counter may be selected to have a range that is large enough (*e.g.*, 24 bit, double check) so that wrap-around conditions (*i.e.*, a reset from the highest value in the range back down to the lowest) occur infrequently.

[00103] Upon wrap around of a page incarnation counter, two methods of exception handling may be applied. In a brute force approach, all remaining stale copies of the page 812 may be eliminated; it will be appreciated that this approach may trigger a great number of page moves, even in the case where neighboring page data are still valid. In an alternative approach, an algorithm that closely interacts with garbage collection and block recycling (described below) may be employed to track the correct highest count even after wrap around has occurred. Many such algorithms may be devised to keep track of a most recently modified version of a page 812 after a counter wraps around, and the present disclosure is not intended to be limited by any particular approach or methodology.

[00104] In some embodiments, blocks 811 may be recycled efficiently by tracking a valid page (VP) count identifier and a continuity factor (CF), where VP count may be an indication of a number of valid pages 812 written to a block 811, and CF may be an indication of a number, or percentage, of pages 812 within a block 811 that no longer contain up to date data. In accordance with this paradigm, a block 811 having a low VP and a low CF is at the beginning of its life span – not all pages 812 have been written yet, and relatively few pages 812 are out of date – and hence this particular block 811 may be assigned a low priority for recycling operations. A block 811 with a high VP but a low CF may be nearly full or totally full – most, if not all, pages 812 have been written – but the block 811 as a whole contains relatively few pages 812 that have been

subsequently invalidated by more updated versions; such a block 811 may be assigned a medium or middling priority for recycling. A block 811 with a high CF, on the other hand, may be considered a candidate for recycling having a relatively higher priority, since many, if not most, of its pages 812 no longer contain valid data.

[00105] In accordance with one strategy, as long as there are sufficient blocks 811 available in a given LUN 810n to accommodate short term copying of pages 812 without rewriting or excessive wrap around activity, no old block 811 should be forcefully recycled as long as it has space to allocate to more than one unwritten page 812; this strategy may minimize, or at least reduce, unnecessary page writes. When free resources in a given block 811 become sparse, the block(s) 811 with the highest CF and the lowest VP may be freed by moving the remaining valid pages 812 to other blocks 811 having free space, and then erasing the block 811 from which all remaining valid pages 812 have been moved.

[00106] In most conventional Flash applications, retrieving (reading) a page 812 has a latency of about 100µs, as the page 812 is first copied to an on-chip cache 819 as set forth above; this typically occurs before the page 812 is ready to be transferred across a bus (*e.g.*, via data store interface 145n) to compute array 142. The data transfer of one page 812 from that on-chip cache 819 to the compute array 142, in contrast, requires less than a third of that overall time – *i.e.*, around 32 µs.

[00107] In order to mask the foregoing latency, each physical channel 899 in FIG. 8 may accommodate multiple LUNs 810, each of which may be retrieving pages 812 to cache 819 concurrently. Once an array of LUNs 810 (or targets) is loaded with read commands, further read commands may be interleaved while data from pending reads are retrieved from cache 819. With sufficient targets (for example, four or more) on a channel 899, the available bandwidth for a given channel 899 may be fully saturated with data moves (streaming). As the LUN 810 count increases, it becomes possible to read partial pages 812 (for example, a ½ page or a ¼ page) without any substantial loss in channel utilization (interleaving the additional read commands has minor impact on overall throughput). Reading partial pages 812 efficiently becomes a valuable feature in more advanced data mapping methods (see the discussion of data structures with reference to FIGS. 10 through 14 below).

[00108] Specifically, executing commands against a LUN 810 on a Flash chip will always result in latencies and minimum (but not deterministic) wait times before a previously issued command

is executed and the LUN 810 is available for further exchanges. As noted above, the biggest latencies are related to commands that retrieve a page 812 (around 100µs) and write a page 812. During those wait times, many features of a LUN 810 are generally unavailable. The status of a LUN 810 may be polled by issuing a status request command (for instance, from storage interface component 411n or elements thereof such as routing block 413n).

[00109] As also noted above, in accordance with the disclosed architecture, payload data and command sequences share the same bus or physical channel 899. Accordingly, to maximize throughput on a given channel 899, LUN 810 access may be interleaved across many targets (*i.e.*, pages 812), and an estimate of availability of each LUN 810 may be determined and maintained in order to minimize interrogation overhead. This functionality, and that set forth below, may be provided by storage interface component 411n, in general, and routing block 413n and data store interface 145n (either individually or in combination), in particular.

[00110] In that regard, storage interface component 411n or its components may take all of these requirements into consideration, and may be configured and operative to provide enough processing functionality to reduce or to eliminate the requirement for central processing unit (CPU) intervention (either by compute array 142, processing system 430, or both) by providing hardware resources necessary or sufficient to effectuate some of all of the following (*e.g.*, in routing block 413n): provide or control a dedicated command buffer per LUN 810 of a given physical channel 899; provide or control a dedicated command execution timer per LUN 810 (this timer may be set with a minimum latency of the Flash chip after issuing a command, such that status polling only begins after the minimum delay time has elapsed, and therefore keeps the channel 899 less congested with status requests); provide an autonomous status request command generator, so that status polling does not require CPU intervention; and provide a session identifier buffer for each command that is posted by processing system 430 into the various command FIFOs in routing block 413n (this may allow processing system 430 to "set and (almost) forget" issued commands, as storage interface component 411n may post completed (or failed) transaction updates in message mailboxes 452n along with a corresponding session identifier). In that regard, mailboxes 452n may have the utility set forth above with reference to FIG. 4.

[00111] It is also noted that routing block 413n, memory interface 145n, or a combination of these and (possibly or optionally) other components of storage interface component 411n may also provide hardware, firmware, or software resources necessary or sufficient to effectuate

selective storage of data in a "co-located" manner. In this context, the terms "co-located" or "co-location" generally refer to the operational characteristic (or its practical effect) of maintaining data records that are related to each other in or at memory addresses such as are reflected by or implemented in blocks 811, pages 812, or both that are relatively proximate to each other (*e.g.*, in physical space, logical space, or both) as compared to other blocks 811 or pages 812 in data store 143. Header information or metadata associated with particular data may be used to designate or otherwise to identify certain data as related to or relevant with respect to other data for this purpose. As set forth above, data may be supplied to system 100 with information (that may be recorded in index 813, for example) that may include a data type index or identifier, various statistics, and other information that may be used, among other things, to co-locate particular data with respect to other related data in a particular page 812; in the foregoing manner, related data may be stored in adjacent or contiguous pages 812 or blocks 811, facilitating retrieval of related data using reduced or minimal read operations.

[00112] One particular implementation that may benefit from co-located data may be configured and operative to organize large datasets acquired, by way of example, from a STARE (Surveillance Technology Advancement & Replacement for Ebsicons) data acquisition system. A typical STARE system employs a variety of sensors (*e.g.*, nine or more independent sensors, typically deployed in a satellite or high-altitude surveillance aircraft) that acquire optical or other electromagnetic data across a large geographic area and at successive points in time, such as for satellite reconnaissance, weather observations, time-lapse photographic applications, and the like. While the following discussion addresses such a STARE system, the disclosed subject matter is equally applicable to other types of large datasets, as will generally be appreciated by those of skill in the art. For example, on a micro-scale, the types of data storage and analytics techniques employed in a typical STARE application may be equally applicable to computer tomography (CT) scans, computer aided detection (CAD) mammography analyses, and other medical technologies involving optical or image data, as well as to design verification and quality assurance methodologies (such as machine vision inspection techniques, for instance) in the semiconductor, micro-electrical-mechanical (MEM) machines, and other small scale electrical and mechanical industries.

[00113] In accordance with one implementation, a respective data stream from each of a plurality of data sources (for instance, each respective sensor in a STARE array or each respective channel in a medical imaging device) may be ingested as a dedicated dataset for a

given point in time (*e.g.*, a temporal STARE index). In this example, the respective output of each respective sensor or channel may be considered an independent data source, while the data from each respective independent data source may be considered (and identified, designated, or otherwise characterized as) "related" to all the other data acquired from others of the independent data sources. For instance, data acquired by a specific sensor operating in a specific spectral band on a given satellite may be considered a specific, independent data source; if that satellite employs multiple sensors, each operating in a respective (though, perhaps, overlapping) spectral band such that the satellite, as a whole, acquires and provides data across multiple spectral bands, each such independent data stream may be ingested by system 100 (and, in particular, by compute node 140n) as different, independent, but related datasets. As noted above, the relationship between such independent datasets may be encoded in metadata or header information, for example, such that related datasets may be co-located when stored or written to data store 143 or other memory component.

[00114] In the foregoing manner, and in the context of a STARE system, each independent dataset may represent a collection of data points from a specific sensor observing or dedicated to a specific geographic region at one point in time. In the case of an arthroscopic, mammographic, CT, or other procedure involving medical imagery or other data collection application, for instance, each independent dataset may represent a time-specific or temporally varying collection of data points from a specific type of sensor covering a specific anatomical region or that is dedicated to collecting a particular type of data. Applications include automotive diagnostics, time-dependent stresses in aerospace structures, inventory and logistics tracking, machine vision and machine learning implementations, and any other application in which large datasets and analytics techniques are relevant.

[00115] Returning to the STARE example, each record in a particular independent dataset may include fields to designate, for instance, a STARE index and sensor data sufficient to identify a particular data source (*i.e.*, the sensor from which the data are acquired), a position of each sample in the data stream (*e.g.*, a time stamp or other unique indicium), and any related data streams to facilitate co-location of data during storage. It is noted that some or all data acquired by some or all of the plurality of data sources may be multi-dimensional (*e.g.*, intensity, frequency, directionality or orientation, altitude or distance from sensor to imaged object, and the like, depending upon the application and the type of data collected). Some or all of these factors may be encoded into, included in, or referenced by header information or other metadata associated

with each data packet provided by the independent data source or otherwise derived by other components as a function of or dependent upon the independent data source and *a prior* or predetermined knowledge of its operational characteristics or deployment.

[00116] In some implementations, a user or administrator of system 100 may choose to divide data associated with independent data sources which cover large geographic areas (or other areas or regions of interest) with high resolution into multiple smaller datasets, each of which may include or be specifically directed to a fraction of the overall geographic area or region of interest that is covered by the data source as a whole. If this is done with a suitable amount of spatial overlap, boundary conditions that may arise from distributing the smaller datasets to multiple compute nodes 140n may be mitigated.

[00117] Those of skill in the art will appreciate that the foregoing functionality may be readily implemented, for example, via suitable programming in an application programming interface (API) and/or in logic at compute array 142, routing block 143n, or both, such that related data may be recognized as such and ingested and reproduced as indexed and related datasets in connection with STARE or other applications. In some implementations, permanent, non-intrusive detailed instrumentation may be integrated into programmable logic block 410, for instance, and cooperate with an API to effectuate appropriate hardware configurations, analysis interfaces, and setup methods to accommodate this embodiment supporting co-location of related data.

[00118] Some of the same factors that enable co-location of data also facilitate implementations in which algorithms may operate on multiple datasets. In that regard, it will be appreciated that typical algorithms employ only a single dataset for input; this is both conventional and sensible, since a typical algorithm seeks to apply the same set of rules or instruction sets to data that are related in some way, and since the relationship between particular data points is conventionally defined by the fact that the data points reside in the same dataset. Where, as set forth above, different, independent data streams may be identified as related (*e.g.*, by header information or metadata) across independent datasets to support co-location, however, such relatedness may be leveraged to enable a particular algorithm to execute on multiple datasets in a single instantiation of the algorithm.

[00119] By way of example, a typical API call to execute an algorithm only allows the use of fields that belong to a single (open or active) dataset; multiple dataset computations are difficult

to implement and require extensive programming and computational (and data record access) overhead. However, using the above techniques that contemplate importing and storing data streams as dedicated datasets from independent data sources, and since those multiple sources may be considered for thresholding calculations and other house-keeping and initiation procedures, an API or programmable logic block 410, or both, may readily be modified to accommodate use of multiple datasets by a single instantiation of an algorithm. In that regard, programmable logic block 410 may be implemented such that it allows for each field which provides data for computations to be a member of any dataset that is available (*e.g.*, stored in data store 143n or other applicable memory structure). In a STARE implementation, such algorithms may be embodied in or comprise Connected Component Labelling (CCL) algorithms or other implementations, though these and other algorithms or heuristic approaches may have equal utility in image processing and other applications as noted above.

[00120] FIG. 10 is a diagram illustrating a conventional data record structure, and FIG. 11 is a diagram illustrating one implementation of a data record structure having utility in connection with a compute node operative in accordance with the present disclosure. Further, FIGS. 12 through 14G are diagrams illustrating additional implementations of data record structures having utility in connection with a compute node operative in accordance with the present disclosure.

[00121] In most Big Data and high performance computing applications, many computations may be performed on columns of data, or vectors, rather than on a single data point or value. This typical mode of operation may indicate or suggest that one useful way to store data in Flash is to map columns to dedicated Flash pages for efficient retrieval. On the contrary, however, some applications may benefit from organizing data in records organized in such a manner that a Flash page holds multiple fields (columns) of a common record. Prior data management and data processing systems address these competing requirements by storing the data twice: once, organized in columns; and then again, organized in records.

[00122] In accordance with the disclosed architecture of system 100, since each compute node 140 may have sixteen or more physical channels 899 that may operate concurrently, a sophisticated data mapping method, examples of which are set forth below, may allow for certain features that are typically unavailable in the context of single channel storage solutions.

[00123] The following examples, and the illustrations in FIGS. 10 through 14G, are based upon a hypothetical dataset of eight records, each record consisting of eight fields or columns. In the

drawing figures, a respective color (represented by a unique hatching pattern) is used to indicate a respective column or field type in a record, and a unique number is used to indicate a row or index for each unique record. In the context of the present disclosure, the terms "record" and "page" may be used interchangeably, and the terms "field" and "column" may be used interchangeably.

[00124] In the prior art data record structure depicted in FIG. 10, for example, each distinct row (1 through 8) represents a distinct data record (numbered 1 through 8); as indicated by the legend "Records 1 – y," it will be appreciated that such a data record structure may include more or fewer records than shown. Each column in FIG. 10 represents a particular field, each of which is distinct from the others as indicated by the respective color hatching; as indicated by the legend "Fields 1 – x," it will be appreciated that each record (1 through 8) may include more or fewer fields than shown.

[00125] In accordance with the disclosed subject matter, however, when staged or reformatted for storage in Flash (such as data store 143), the data may be, in effect, rotated by 90 degrees, such that all original fields in the same original column are stored in a new dedicated page (or pages), *i.e.*, a new dedicated page comprises fields, each comprising a single field type. This is illustrated in FIG. 11. Specifically, each respective row in FIG. 11 represents a new respective record, 1 through x (eight, in this example); *i.e.*, one new record for each discrete field in the original data structure. Each new record comprises a number of fields (eight, in this example) 1 through y, corresponding to the number of records in the original data structure of FIG. 10, and each field is of the same field type mapped from the same column in the multiple records in the original data structure.

[00126] In the foregoing manner, original records may be re-mapped into columns, and original fields may be re-mapped into discrete records or pages 812, each comprising a column from the original data structure; each new record (*i.e.*, representing an original column) may be stored in a dedicated Flash page 812 of any channel 899. In the FIG. 11 example, page 812 retrieval may be executed efficiently, as set forth above, though fetching related data may be further optimized with some additional manipulation.

[00127] For example, if only one channel 899 were available, one additional mechanism to facilitate optimization may be to spread the newly formatted pages 812 across multiple LUNs 810 so that access to the various pages 812 may be executed in an interleaved manner. As noted

above, such an interleaving strategy may have a positive effect on overall throughput with respect to the transfer time associated with writing a page 812 to cache 819 in the Flash array; in particular, this specific latency type of may be masked by use of interleaving, in accordance with which other data types or records from a second LUN 810 may be transferred across routing block 413n to compute array 142 while a first cache 819 associated with a first LUN 810 is still filling. This arrangement is illustrated in FIG. 12, which depicts the data structure of FIG. 11 distributed across multiple LUNs 810a, 810b, 810c, ... 810n.

[00128] In some implementations in which each compute node 140 accommodates sixteen (or more) parallel channels 899, it may be useful to distribute data traffic across all available channels 899 as described above with reference to FIGS. 4 through 9. In such an embodiment, it may be useful to distribute the data structure of FIG. 11 across the available channels 899 in the manner illustrated in FIG. 13 (only eight channels 899 are shown, for clarity). As illustrated in FIG. 13, a LUN 810a may comprise newly formatted pages 812 (each having an index 813) as set forth above; in this example, however, the newly formatted pages 812 are distributed across each channel 899 available to handle data transfers to and from the data store 143. It will be readily appreciated from examination of FIGS. 13A through 13H that the manner in which pages 812 are stored may be manipulated in accordance with a particular one of the channels 899 with which those pages 812 are associated. For each successive channel 899, for instance, a location of pages 812 associated with the same field type from the original data structure is incremented by an offset value, and each successive channel includes data from the next successive eight original records.

[00129] In the illustrated implementation, the order in which pages 812 are stored (*i.e.*, from record 1 through record 8) by original field type remains constant, but the first record 812 associated with a particular channel 899 increments by an offset value (for example, 1) for each successive channel 899. For example, a first channel 899 may be assigned or associated with a dataset in LUN 810a that begins with a first record or page 812 that represents a first field from the original data structure shown in FIG. 10, a second channel 899 may be assigned or associated with a dataset in LUN 810a that begins with a second record or page 812 that represents a second field from the original data structure, and so on. This is represented in FIGS. 13A through 13H, which depict a respective dataset for each respective one of eight channels 899 that are useable by LUN 810a (as indicated by the dashed lines in FIG. 13).

[00130] It is worth noting here that the actual (*i.e.*, physical) order in which pages 812 are stored in data store 143 (*e.g.*, a Flash array) need not change from channel 899 to channel 899 in many applications; for example, the same effect may be achieved logically or virtually by associating each series of pages 812 with an index or a look up table value that identifies a first page 812 for each channel 899. As long as a read operation (for example, requested by compute array 142 in cooperation with routing block 413n) is apprised of an index and an offset value for each channel 899, the specific order in which pages 812 are stored for any given channel 899 may be irrelevant. In one implementation, such indices and offset values may be maintained in node memory 450 (*e.g.*, in FP tables 454n) and communicated as necessary to routing block 413n via DMA 146b substantially as described above, though other alternatives may have similar or analogous utility. The depiction in FIG. 13 is provided by way of example only, and for purposes of visualization, but those of skill in the art will appreciate that the illustrated structure may be achieved virtually without undue experimentation. It is also noted that the data record structure shown in FIGS. 13 and 13A through 13H may be altered or modified to accommodate the number of physical channels 899 available to a particular LUN 810n, the bandwidth or processing capabilities of compute array 142, the number of records, fields, or both in the original data structure, or a combination of these and a variety of other factors.

[00131] In particular, the diagonal striping depicted in FIG. 13 is indicative of a unique feature provided by this approach: data may be read massively parallel as records (*i.e.*, from the original data structure in FIG. 10), on the one hand, and as pure columns (*i.e.*, from the original data structure in FIG. 10), on the other hand. The type of read may be selectively adjusted or determined by identifying a page 812 selection for each channel 899. As noted above, such a read operation may be requested by compute array 142, either independently or in cooperation with routing block 413n, and may be supported by data retrieved from node memory 450, in some applications.

[00132] If a read request for all channels 899 points to the same page 812 location (*i.e.*, the number of spaces measured from the top down, or from the bottom up, in FIG. 13), then horizontally across all channels 899, this is represented by the row at reference numeral 1399. Reading all channels 899 across the entirety of row 1399 will result in retrieval of complete records for the entire dataset in LUN 810a in parallel. This method may be useful when an application or algorithm needs to check for many conditions concurrently. As noted above, the same result may be effectuated by using a page index and offset value for each respective channel 899,

irrespective of location of a particular page 812 in a data structure maintained for that particular channel 899.

[00133] Additionally or alternatively, the patterned data record structure of FIG. 13 may also be effective in finding all individual records or pages 812 with a specific condition (or field type), for example, by setting an offset value in each channel 899 differently (*i.e.*, diagonally across) so that all channels 899 only deliver that specific condition or field type in parallel. For this search or read strategy, no reformatting of the data structure is required; both record-wide and column-centric selection may be equally effective in the context of effective storage bandwidth utilization.

[00134] To facilitate reassembly of related fields (from the newly formatted data structure) into complete original records after the different fields have been mapped to different Flash pages 812, each index 813 for a new page 812 may include, in addition to the information and metadata noted above, an indication of the field type (*e.g.*, a column number or other identifier from the original data structure in FIG. 10) and record (again, from the original data structure) with which a given page 812 should be associated, as well as an index number or other identifier of both the first and last record (from the original data structure) from which the fields contained in page 812 originated. In some embodiments, compute nodes 140 may assign a local system index to each page 812, for example, as each page 812 is received or otherwise acknowledged (*e.g.*, during boot or initialization). This index may be globally unique if combined with a node identifier and a card identifier, for instance, which uniquely identify compute node 140n and its card 200 in the context of system 100.

[00135] In one simple implementation employing eight compute nodes 140 each having sixteen memory channels, it is possible to spread the various field types across all eight compute nodes 140 for a total of one hundred twenty eight parallel field types. In the absence of sophisticated mesh connectivity, however, it may be cumbersome to accommodate compound conditions across multiple compute nodes 140 in this context, and even one hundred twenty eight field types or columns may not be sufficient to reflect many real-world applications. Accordingly, it may be appropriate or desirable in some circumstances to utilize Flash pages 812 that hold, maintain, reflect, or represent multiple field types (*i.e.*, columns from the prior art data structure of FIG. 10). This is depicted in simplified form in FIGS. 14A through 14G.

[00136] Specifically, the illustrated approach may be employed in some situations to mitigate or to eliminate shortcomings generally associated with current known limitations of block RAM

(BRAM) technologies implemented by many commercially available FPGA structures, for example, such as those having utility in connection with compute array 142. In applications requiring storage and/or retrieval of a vast number of datatypes (*i.e.*, fields) in parallel to facilitate examination or analysis of many conditions concurrently, conventional BRAM-based systems simply do not (and cannot, as a practical matter) comprise enough local storage to store or otherwise to accommodate many records with possibly thousands of fields at the same time.

[00137] Enabling multiple field types to be populated in a single page 812 of memory, and realizing benefits from reduced bandwidth overhead associated with partial page reads (made possible by deployment of multiple LUNs 810 in a high density constellation as illustrated and described herein), allows a compute node 140 to achieve higher throughput rates as compared to a strategy that requires reading specific entries from freshly retrieved entire pages 812 for each read request.

[00138] In some implementations, only minor adjustments to logic (*e.g.*, at routing block 413n, compute array 142, or both) or a few additional instructions (*e.g.*, in FP tables 454n at node memory 450) may be useful to enable partial page read functionality. For example, with mixed column types in a page 812, a normal page header or index 813 as described above may be modified such that it refers to a secondary page header describing the contents of a page 812 and a boundary between field types; using this mechanism, and by apprising relevant logic or instructions sets that partial page reads are possible, fractional compound pages such as those illustrated in FIGS. 14A through 14G may be maintained without altering the above-mentioned methods for selecting pages, pre-sorting, sorting, and so forth.

[00139] Many different applications and algorithmic processing strategies may benefit from partial page reads, particularly in situations where only data from some of the subpages, or a portion thereof, are relevant for a particular computation to be executed; by using a partial page read strategy, other (irrelevant) field types in a particular page may be ignored (*i.e.*, may not be retrieved) to economize on bandwidth and computational or other processing resources. In some such embodiments, it may be desirable to implement a strategy to rotate between active LUNs 810, and their associated resources, to maximize or otherwise to increase throughput (as in the example above addressing a single field type per page).

[00140] FIG. 15 is a diagram illustrating one strategy for data communications between a host compute system and a data processing and analytics system as depicted in FIG. 1. FIG. 15

illustrates, in simplified form, one example of certain data communications that may support the foregoing functionality in the context of a data processing operation or compute task requested by a host compute system 199. Specifically, communications (calls/receipts) operations for host compute system 199 are illustrated on the left of FIG. 15, communications passing through router module 120 are illustrated in the middle of FIG. 15, and communications operations for compute node 140 are illustrated on the right of FIG. 15.

[00141] Among the other things noted above with specific reference to FIG. 1, host compute system 199 may generally comprise a shared library 1510. In some implementations, shared library 1510 may be configured and operative to implement an appropriate application programming interface (API) to enable bi-directional data communications between host compute system 199 and router module 120 as described above and as depicted in FIG. 15. In particular, an API at shared library 1510 may be embodied in or comprise a procedural interface implemented on one or more of common or popular operating systems typical for host compute system 199; examples of such an operating system include Linux™, Windows™, and Mac™ OS™, though others may be available currently, or may be developed in the future. As is generally known, an API binding typically provides basic functionality to authenticate API requests, to enable or to facilitate text encoding/decoding, and to provide generic support for read and write requests; in some implementations, the API at shared library 1510 may employ C™ and Python™ bindings, though others may also be employed. Further, the API may parse dataset descriptions and perform basic parameter validation, as is also generally known.

[00142] As set forth above, the API may also have utility in storing and retrieving datasets in a co-located manner. For example, shared library 1510 may be configured and operative (such as by the foregoing parsing of dataset descriptions, parameter validations, or other techniques) in cooperation with the API to identify specific datasets and to characterize same as related to other distinct datasets. Such a relatedness determination, which may be binary or effectuated in accordance with a weighted scale (*e.g.*, to provide a relatedness "score" or other value for a dataset relative to one or more selected other datasets) may be used to influence a physical or logical location in shared library 1510 at which data are maintained. For example, it may be beneficial in some circumstances to store related datasets in adjacent, continguous, or alternating memory blocks in shared libary 1510. The same may also be true for memory store 143n, as noted above, and so, additionally or alternatively, components at host compute system 199 may communicate datasets to compute nodes 140 (such as illustrated in FIG. 15) with such

relatedness information encoded in metadata or header information, or even embedded within the datastream itself.

[00143] In some embodiments, shared library 1510 may be implemented as a binary libary (for example, on Windows, it may be in the form of a dynamic link library (DLL)) and may be provided as part of a simple software development kit (SDK) that includes appropriate header files, in a manner that is similar to support provided for any other generic C or Python library. In use, shared library 1510 may generally be apprised of, and may maintain, information related to open datasets for each individual user of host compute system 199 (*i.e.*, it may be desirable that information within shared library 1510 is not shared between individual users, but is rather shared between discrete resources of system 100).

[00144] As indicated in FIG. 15, host compute system 199 may further comprise a driver 1520. In operation, functionality of driver 1520 may be called by shared library 1510. In that regard, driver 1520 may employ standard input/output (I/O) calls, such as read and write, and may be operable in cooperation with Linux, Windows, Mac OS, or a combination of these and other operating systems (depending, for instance, upon the operating system governing functionality of host compute system 199 as a whole). Driver 1520 may translate dataset names to internal dataset identifiers, for example, as may be necessitated by host compute system 199 operating system or other resources, and may also provide full parameter validation, security and sharing management functionality, and administrative tasks as are generally required by operating systems and other components in host compute system 199. Specifically in the context of the disclosed subject matter, driver 1520 may manage activity of compute nodes 140 (for instance, in cooperation with individual processing systems 430) by sending commands and receiving (and, optionally, aggregating) responses as necessary during run-time. In that regard, it is noted that some of the functionality of driver 1520 may be provided in cooperation with one or more of the components of processing systems 148/430, and may influence or benefit from data in node memory 144/450, described above with reference to FIGS. 1 and 4.

[00145] Driver 1520 may also maintain and organize open datasets for all registered or authorized users of host compute system 199, maintain dataset identification data or indicia for same, and enforce administration privileges or other access rights to components or modules that are fundamental to operation of host compute system 199.

[00146] Compute node 140 may generally comprise, in addition to the components described above, a firmware module 1590 or instruction set repository as indicated in FIG. 15. Firmware 1590 may be embodied in a dedicated memory component, for example, or may reside in or be maintained by an element of processing system 430 or node memory 450, in some instances. The disclosed subject matter is not intended to be limited by the manner in which firmware 1590 is stored or deployed in the context of the various hardware and software components of compute node 140. In operation, firmware 1590 may generally implement an appropriate data transfer command/response protocol sufficient to enable the communications and functionality set forth herein; in that regard, firmware 1590 may execute or accommodate an asynchronous command packet / response packet interface to allow the communications depicted in FIG. 15 and otherwise to facilitate data transfer between compute node 140 and router module 120 in accordance with the foregoing methodologies and principles. As set forth above, this feature may also comprise one or more mechanisms supporting col-location of related datasets, either independently or in cooperation with information received from host compute system 199 and associated with the data transfer.

[00147] Specifically, firmware 1590 may, in some embodiments, be configured and operative to manage data transfers between compute array 142 and data store 143 (*e.g.*, a Flash array). For example, firmware 1590 may be responsible for, manage, or otherwise facilitate some or all of the following functionality: handling boot-up initializations for compute node 140; building and managing indices or look up tables indicative of data locations for all dataset records managed by compute node 140; creating and managing datasets and any associated dataset descriptors (including co-location indications or relatedness scores across datasets); optimizing data storage by field type per dataset; managing page 812 or block 811 write optimization protocols or strategies (including, but not limited to, co-location or dataset density approaches); managing page 812 replacement and page 812 aging strategies; loading and execution of algorithms in compute array 142 (*e.g.*, such as may be embodied in an FPGA fabric); loading relevant parameters needed or desired for algorithm execution (including whether a particular algorithm may execute on multiple, related datasets); collating and, optionally, passing data (*e.g.*, read data and/or algorithm results) between and amongst components in compute node 140; and generating dataset identifications or other indicia for dataset creation and co-location or relatedness scores, and coordinating same across all compute nodes 140 in a particular distributed processing system 100.

[00148] In the context of all datasets, firmware may maintain and organize all names, access strings, identifiers, descriptors, indices, relevant relatedness scores or co-location preferences, if applicable, and the like. For a particular dataset, firmware 1590 may also maintain a record of data pages 812 that contain dataset data and are located or that reside on or in association with a particular compute node 140. Additionally, firmware may also maintain data associated with a list of outstanding read or write requests (or both) for data that are currently being transferred. As noted above with respect to driver 1520, it will be appreciated that some of the functionality of firmware 1590 may be provided in cooperation with one or more of the components of processing system 148/430, and may influence or benefit from data stored in node memory 144/450.

[00149] Turning now specifically to the data flow illustrated in FIG. 15, a user's application program may make an API call at host compute system 199 to initiate a data processing operation. In some implementations, the API may include different routines for different functions, such as CREATE, OPEN, CLOSE, WRITE, READ, EXECUTE ALGORITHM, DELETE RECORDS, DELETE DATASET, and the like. Typically, the API expects that a dataset to be accessed will or must be specified by name and access string (password) or in accordance with some other unique identifier or identifying indicium.

[00150] In this example, shared library 1510 may field the API call and check, authenticate, verify, or otherwise confirm basic parameters. For example, if the request targets a particular dataset, then shared library 1510, either individually or in cooperation with other components of system 100, may check to ensure that host compute system 199 has that dataset open or is in the process of opening it (*e.g.*, such that the target dataset, or a relevant portion thereof, is accessible in, or via, RAM or via another short term storage medium). As another example, if the request were to create a dataset, shared library 1510, either individually or in cooperation with other components of system 100, may parse and verify a relevant dataset descriptor or other information related to creation of such a dataset. If the request were to access dataset records, on the other hand, shared library 1510 may verify valid record numbers and ranges, again, either autonomously or in cooperation with other system 100 components.

[00151] In some implementations, shared library 1510 may be responsible for building the basis for the requested or required command, for example: the dataset may be specified by name; relevant command codes and other parameters may be set; a "write" command may be formatted to specify a number, say from 1 to n, of data records to be written; a "read," "delete record," or other command may be formatted to specify a starting record number (or an index and offset as

set forth above, for instance) and a number of records to be accessed or affected; and so forth. The types of commands recognized and constructed by shared library 1510 are myriad, and may be application-specific or dynamically created, for example, as a function of the processing operation sought to be executed.

[00152] In that regard, shared library 1510 may build an "algorithm load" or "algorithm execute" command, or both, as necessary or as instructed by host compute system 199, and may retrieve, or request retrieval of, an algorithm bit stream or other pertinent instruction sets from an algorithm library or other memory storage facility maintained, for example, at host compute system 199. Upon specification of a suitable bit stream or other data and information associated with a particular algorithm to be executed, as well as selection of required or desired filters, boundary conditions, relevant variables, or a combination of these and other command parameters, shared library 1510, or a proxy, may then make a call to driver 1520, which call may include a dataset name and other relevant command parameters sufficient to initialize driver 1520 with appropriate data to begin data transfer and processing operations.

[00153] Such "I/O" calls and "Read" and "Write" data transfers are illustrated in FIG. 15 at the vertical line representing a boundary between user operations and kernel (or low level, operating system) operations in host compute system 199.

[00154] Driver 1520 may field calls from the user domain (*i.e.*, across the "I/O" call boundary illustrated in FIG. 15), such as may be embodied in or comprise a request or command string from shared library 1510. Responsive to such a command or request, driver 1520 may execute, or may ensure that some other component executes, a full validation or verification of each and every parameter in a particular command or request. For example, many commands use a dataset identifier or unique indicium or indicia for efficiency; in such instances, driver 1520 may translate a dataset name to an associated identifier for downstream use at compute node 140. As another example involving security checks and administrative privileges, driver 1520 may maintain a record of authorized users, and thus may be apprised of which users are administrative users that have attendant administrative rights or privileges. In situations in which a particular command or request is one that requires administrative rights, driver 1520 may enforce administrative access rules as dictated or suggested by other elements of host compute system 199; alternatively, driver 1520 may ignore such rules involving regulated access to system resources in instances where a particular command or request does not implicate administrative access rules or policies.

[00155] Similarly, driver 1520 may be configured and operative to maintain data indicative of which datasets, system- or enterprise-wide, that are open and currently accessible; further, driver 1520 may also maintain data or other information related to open datasets with respect to which sharing is or may be allowed, and under what circumstances (*e.g.*, sharing may be permitted if so declared when the dataset was opened or if a dataset is related to an open dataset, for instance, or when the dataset was created), as well as which records are valid within a particular dataset (*e.g.*, including a total number of valid records and a percentage of valid records as compared to all records, valid and invalid or out of date).

[00156] If driver 1520 determines that all examined criteria and parameters are within acceptable ranges (*i.e.*, a command or request is properly constructed and formatted), then a cyclic redundancy check (CRC) may be calculated for the command packet (e.g., creating a validated or authenticated command packet), and the newly validated command packet may be transmitted to compute node 140 via host interface 126 (which may be, for example, a PCIe bus in some implementations, as noted above). This data transfer is illustrated in FIG. 15 at the vertical line (labeled "PCIe Bus") representing a boundary between kernel operations in host compute system 199, on the one hand, and router module 120, on the other hand. Router module 120 may then forward the verified command packet to one or more compute nodes 140 (*e.g.*, via communications links 143) substantially as described above.

[00157] At a compute node 140, firmware 1590 may receive a verified command packet and may execute authentication procedures to verify the CRC for the packet; as described above with respect to driver 1520, firmware 1590 may also execute a full parameter check in the compute node 140 domain. Given valid parameters and successful CRC check, firmware 1590 may then execute the command specified by the verified command packet.

[00158] It is noted that a particular command may include zero, one, or a plurality of page read and/or write operations, and instructions for these may be parsed by firmware 1590 and transmitted to elements of compute array 142, processing system 430, or both for execution.

[00159] For algorithm load and algorithm execute commands, the algorithm bit stream provided by shared library 1510 may be loaded into compute array 142, processing system 430, or both, and executed substantially as set forth above with reference to FIGS. 1 and 4. For example, selected data records may be streamed to and from compute array 142, generally in cooperation with routing block 413n, and multi-channel algorithm results may be combined into a

single result (for example, by compute array 142, routing block 413n, or both, either independently or in cooperation with processing system 430).

[00160] A response packet (illustrated at the top center of FIG. 15) generated at compute node 140 may generally describe what happened during data processing operations, and may provide, include, or reference a result (or an analysis of results) generated during computational execution of a relevant algorithm. Data payload in a response packet may include some or all of the following: a status of an ongoing processing operation (*e.g.*, percent complete, estimated time to completion, estimated compute cycles, estimated node resources that must or may be brought online, *etc.*); how many and which particular dataset record(s) are involved or may be implicated or affected by an ongoing processing operation; and what (interim) results have been obtained so far, along with an accounting or other report of such results. Most command packets generally expect a single response, but that is not true for all command packets; in some implementations in which command packets may expect or allow for multiple responses, those multiple responses may be aggregated, either at compute node 140 or at driver 1520, for example, as a function of system design choice or simply to decrease or otherwise to optimize data transfer efficiency or bandwidth utilization.

[00161] In any event, a CRC may be calculated for the whole response packet as generated by elements of compute node 140. Firmware 1590, or a surrogate or delegated component, may then transmit response packets back to host compute system 199 (*e.g.*, via router module 120, across host interface 126 bus, and to driver 1520). Upon receipt of a response packet (*e.g.*, from host interface 126), driver 1520 may then verify the CRC associated with or included in the response packet, ensure that the response matches or is coordinated with outstanding or pending command packets, and manage these and other operations in the context of a multi-access environment in which multiple compute nodes 140 are communicating with a single host compute system 199. In that regard, driver 1520 may aggregate responses, for instances, from a single compute node 140 or from multiple compute nodes 140, and may return such responses, as well as any appropriate or attendant status information to shared library 1510. At this point in some implementations, shared library 1510 may then take control of program execution, update its data structures based upon information in received responses, and send new or updated data to a particular user's buffer or other temporary, user-specific data storage area. Program control may then be returned to a user's program or other software application that requested the data processing operations in the first place.

[00162] FIG. 16 is a functional flow diagram illustrating aspects of one implementation of a method of reformatting a data structure. As indicated at block 1601, a method of reformatting a data structure may begin by receiving a block of data, the block of data comprising a number, y, of original records wherein each original record comprises a number, x, of original fields. See, *e.g.*, FIG. 10 for an illustration of one example of such an original data record. The receiving operation depicted at block 1601 may be executed, for instance, by compute array 142 as illustrated in FIGS. 1 and 4; additionally or alternatively, this operation may be executed by, or supported by, one or more components of processing system 148/430 as described above with reference to FIGS. 1 and 4.

[00163] The method may proceed to an iterative loop as indicated at block 1602 and decision block 1620. In accordance with this iterative procedure, a number, x, of new data records may be defined, each respective new record comprising y new fields, wherein each respective new field corresponds to a respective original field from a respective original record. If a determination is made (*e.g.*, at block 1620) that additional iterations are required, the method may loop back to block 1602; otherwise, the method may proceed to block 1603. At block 1603, the block of data may be reformatted, responsive to the defining operation (at block 1602) such that a reformatted block of data comprises the new data records, each comprising a single field type. See FIG. 11 for an illustration of one example of a data structure that may result from the operations depicted at blocks 1602, 1620, and 1603. In particular, the operations depicted at these blocks are set forth in detail above with specific reference to FIGS. 10 through 14.

[00164] In this example, the method may continue by associating each of the new data records with an index operative to identify a particular new data record in the reformatted block of data. These indices are illustrated at reference numeral 813 in FIG. 11, for example. The method may then proceed by writing the reformatted block of data to memory (block 1605), whereupon the method may conclude (block 1699).

[00165] To summarize the operations depicted in FIG. 16, particularly in the context of FIGS. 1, 4, and 11, it will be appreciated that a method reformatting a data structure may result in a newly formatted data structure that enables flexible and efficient processing operations and data access in cooperation with conventional memory hardware architectures; for example, vector processing or traditional processing may be selectively implemented with minimal administrative overhead.

[00166] It is noted that the arrangement of the blocks and the order of operations depicted in FIG. 16 are not intended to exclude other alternatives or options. For example, the operations depicted at blocks 1603 and 1604 may be reversed in order, or they may be made to occur substantially simultaneously in some implementations. Further, one or more of these operations may occur substantially simultaneously with the operations depicted at blocks 1602 and 1605 in instances where it may be desirable to do so, *e.g.*, for efficiency, where processing resources are sufficient, when multithreaded or parallel operations are accommodated by the compute array 142 and attendant hardware, and the like. Those of skill in the art will appreciate that the foregoing subject matter is susceptible of various design choices that may influence the order or arrangement of the operations depicted in FIG. 16.

[00167] FIG. 17 is a functional flow diagram illustrating aspects of one implementation of a method of executing data processing operations. As indicated at block 1701, a method of executing data processing operations may begin by providing a compute node communicatively coupled to a host computer and operative to manage data processing operations independent of the host computer. Such a compute node may be embodied in or comprise elements described above with reference to compute node 140; similarly, the communicative coupling may be embodied in or comprise elements of router module 120 such as described in the examples above.

[00168] If a determination is made (*e.g.*, at decision block 1720) that additional nodes are required or desired for a particular data processing operation or application, then the method may loop back to block 1701; otherwise, the method may proceed to block 1702.

[00169] The method may continue by providing a programmable logic component in each compute node as indicated at block 1702; each programmable logic component may be configured and operative to execute data processing operations in cooperation with a first memory component. In some disclosed embodiments, the programmable logic component (e.g., compute array 142 in FIGS. 1 and 4) may be embodied in or comprise a microprocessor, a multicore microprocessor, an FPGA, or other digital data processing component suitable to execute the functionality set forth above. Also as noted above, the first memory component may be, or have access to, a non-volatile data storage device such as EEPROM or Flash memory, though other types of memory are contemplated.

[00170] In some disclosed implementations, the method may proceed by providing a data mover component in the compute node that is configured and operative to facilitate data communications between the programmable logic component and a second memory component (block 1703). The second memory component may include any of various types of solid state storage hardware, such as RAM, SRAM, DRAM, DDR RAM, DDR4 RAM, and the like. The data mover component may utilize hardware and firmware components that are operable substantially as set forth above with reference to FIGS. 1 and 4.

[00171] The method may proceed, as indicated at block 1704, by providing a set of instructions that enable the programmable logic component to reformat a block of data comprising original records, each original record comprising a plurality of fields having different field types, such that a reformatted block of data comprises new records, each new record comprising a plurality of fields of a single field type from the original records. Aspects of an example of such reformatting operations are described above with reference to FIGS. 10 through 14.

[00172] The method may then proceed by utilizing a plurality of communications channels to transfer data associated with the reformatted data block between the programmable logic component and the first memory component (block 1705), whereupon the method may conclude (block 1799).

[00173] As with FIG. 16, the arrangement of the blocks and the order of operations depicted in FIG. 17 are not intended to exclude other alternatives or options. For example, it will be appreciated that in accordance with one embodiment, the operations depicted at blocks 1702, 1703, and 1704 may be executed in any order without materially impacting results of the method or the functionality of any particular hardware implementation utilized to execute the method. In addition to the alternatives set forth in detail above, various design choices that may influence the order or arrangement of the operations depicted in FIG. 17 will be readily apparent to those of skill in the art.

[00174] FIGS. 18 and 19 are functional block diagrams illustrating components of various implementations of a compute node having utility in connection with a data processing and analytics system. It will be appreciated that the architectures illustrated in FIGS. 18 and 19 may be implemented as variations of compute node 140, and that node 1800 may be similar in overall architecture to compute node 140 depicted in, and described in detail with specific reference to, FIGS. 1 and 4. As illustrated in FIG. 19, node 1900 is similar to node 1800, though the former

may omit a data store component (reference numeral 143 in FIGS. 1 and 1800) that is included in or accessible by the latter.

[00175] As set forth above with reference to FIG. 1, a communications link (not illustrated in FIGS. 18 and 19) may be employed to couple compute nodes such as 140, 1800, and 1900 at different successive tiers in series. As noted above, such a communications link (reference numeral 149 in FIGS. 1 and 4) may be based upon and operative in accordance with standard serial interfaces or protocols, though some proprietary serial links may also be used as a design choice. For instance, it may be desirable that a communications link employ or comprise a Xilinx Aurora™ communication link. In any event, such a communications link may be selected, configured, and operative as a function of, or otherwise taking into consideration, the hardware and operation of a compute array 142 (or "algorithm framework"). Those of skill in the art will appreciate that many options exist for both compute array 142 and a communications link, and that specific commercial embodiments have been identified by way of example only, and not by way of limitation.

[00176] The "Flash Mass Storage" functional block illustrated in FIG. 18 may be embodied in or comprise the data store 143 described above with reference to FIG. 1. In that regard, data store 143 may be embodied in or include Flash memory or other EEPROM or non-volatile storage medium. A suitable interface, such as data store interface 145 in FIG. 1 (not shown in FIG. 18), may employ a protocol that is necessary or desirable to allow compute array 142 and algorithm program logic 1890 to communicate with (*e.g.*, manage read and write access to) data store 143 as is generally known in the art.

[00177] The "Firmware Workspace" components illustrated in FIGS. 18 and 19 may be identical, similar, or analogous to node memory 144 described above with specific reference to FIGS. 1 and 4. In that regard, node memory 144 may be implemented as or comprise DDR4 RAM or other volatile or non-volatile memory suitable to support operations of node 1800 or 1900, in general, and processing system 148, in particular. As noted above, node memory 144 may be employed to support algorithm execution in some compute-intensive situations. It may be desirable or useful in certain circumstances that node memory 144 comprise, or at least have access to, dedicated or predetermined addresses or blocks of memory (such as algorithm memory 1841) that may be used, *e.g.*, in an "on demand" or "ad hoc" manner, as processing loads or bandwidth requirements dictate as set forth below.

[00178] In that regard, algorithm memory 1841 may be implemented as pre-allocated addresses, blocks, or sectors of node memory 144, for instance, or it may be dynamically allocated on an "as available" basis as processing requirements vary over time. The latter approach may not reserve or pre-allocate large chunks of node memory 144 when it is not needed, but may also ensure that sufficient algorithm memory 1841 may be available to support complex computational tasks. In the illustrated examples, algorithm memory 1841 may generally comprise resource-specific data storage cells or memory locations (such as the FP tables 454n, algorithm buffers 456n, and FP buffers 458n illustrated in FIG. 4), and may generally be operative to store data and instructions necessary or desirable for processing system 148 to perform write/read/execute operations in support of programmable logic block 410 as is generally known.

[00179] Those of skill in the art will appreciate that the present disclosure is not intended to be limited to any particular memory management techniques that may enable or facilitate allocation of algorithm memory 1841 in the context of using node memory 144 in connection with particular drivers, memory controllers, or address allocation methodologies.

[00180] In the illustrated architectures, processing system 148 (which may be similar or analogous to processing system 148 of FIGS. 1 and 4) may generally handle read/write commands and other memory access with respect to data store 143 (*e.g.*, via a suitable data store interface), receive and act upon interrupt requests (*e.g.*, with respect to operations executing at compute array 142), and regulate, control, or otherwise influence intra-node data transfers across the various components of nodes 1800 and 1900. For instance, in one embodiment described above, processing system 148 may comprise a set of one or more data processing elements or hardware components to manage all interrupt handling as well as AXI or other intra-node data transfers, on the one hand, and a different set of one or more processing components to manage general system administration (or "house-keeping") and management tasks for a node 1800, 1900 as well as algorithm execution support for compute array 142, on the other hand. While aspects of this implementation are described above with reference to FIG. 4, FIGS. 18 and 19 illustrate that different component arrangements and task distributions are within the scope and contemplation of the disclosed subject matter.

[00181] As depicted in FIGS. 18 and 19, processing system 148 may comprise processors 439 (or processing "cores") such as described above as well as additional processing cores 1871 that may be used for algorithm processing or execution support as set forth in more detail below. In that regard, processors 1871 may be analogous to (or implemented in addition to) processors

439c and 439d in FIG. 4, and may generally be embodied in or comprise digital processing hardware and firmware that is similar to or identical to that employed in processors 439, though it may be desirable in some instances to use different hardware having different processing capabilities or operational characteristics for processors 1871 as compared to processors 439, particularly in the case where processors 1871 are dedicated to algorithm execution support (as opposed to a situation where a generic core may be used for algorithm execution only on a part-time or as needed basis). It is noted that bus structures, power sources, controllers, internal memory, and other attendant and supporting hardware and communications paths have been omitted from FIGS. 18 and 19 for clarity. Processors 1871 may be implemented as or generally comprise a microprocessor, a multi-core microprocessor, a microcontroller, an FPGA, a PLC, a PLU, or some similar or analogous hardware component suitable for data processing as is generally known in the art and described above with reference to processors 439.

[00182] In operation, processors 439 and 1871 may be responsible for a variety of tasks which, under some circumstances, may be requested or required by processors 432a and 432b (see FIG. 4), or which may be initiated autonomously or otherwise without direction from external components. In various implementations, processors 439 and 1871 may perform some or all of the following functions, or a subset thereof, as requirements of system 100, in general, or node 1800, 1900, dictate: data transformation and storage (*e.g.*, with respect to node memory 144, memory store 143, or both); data dispatch for write/read operations (again, with respect to node memory 144, memory store 143, or both); command interpretation and execution (*e.g.*, with respect to data processing and algorithm compilation and execution); algorithm load and compute; and memory exception handling. Those of skill in the art will appreciate that various other functions may be assigned to or attributable to processing system 148 as a function of desired operability of node 1800, 1900, and that any of numerous hardware arrangements, computing architectures, and distributed processing resources may be employed at processing system 148 to enable its functionality in the context of supporting desired capabilities of node 1800, 1900. Specifically, processors 1871 may be dedicated (either on a full-time or as needed basis) to support the algorithm execution functionality described in more detail below.

[00183] Further, and to support algorithm execution functionality as necessary or desired, processing system 148 may additionally comprise or have access to a memory component 1870. Memory component 1870 may be dedicated to facilitate processing system 148 functionality, and may be embodied in or comprise an Embedded Multimedia Card (EMMC) firmware Flash device

or other EEPROM or non-volatile storage medium. By redirecting read/write operations from node memory 144 to a more local and dedicated memory component 1870 for computationally intensive algorithm execution operations, processing system 148 may realize performance advantages in high-demand applications.

[00184] As with the FIG. 4 approach, programmable logic block 410 is illustrated in FIGS. 18 and 19 as generally comprising a compute array 142 or algorithm framework, which may in some instances be communicatively coupled to data store 143 by a suitable hardware interface (such as storage interface component 411n in FIG. 4) that is omitted from FIG. 18 for clarity. Aspects of the interoperability of programmable logic block 410 and data store 143 are described above with reference to FIG. 4, and it is noted that the FIG. 18 arrangement is not intended to be limited to any particular memory controller or communications protocol.

[00185] In the implementations of FIGS. 18 and 19, compute array 142 is depicted as further comprising or embodying an algorithm program logic functional block (or "APL," reference numeral 1890). As noted above, compute array 142 may employ or comprise any of various commercial embodiments of an FPGA that may be selected in accordance with any of a number of factors or boundaries conditions that are typical in electronics design. In this case, compute array 142 may be selected to accommodate operation of APL 1890 as described below, as well as to support other required or desired functionality of compute node 1800, 1900.

[00186] In operation, processing system 148 may cooperate with programmable logic block 410 for data transfer and read/write operations with respect to data store 143 (FIG. 18) or, more generally, to support algorithm execution at node 1800, 1900.

[00187] FIG. 20 is a functional block diagram illustrating components of an implementation of a management node having utility in connection with a data processing and analytics system. A management node 2000 such as depicted in FIG. 20 may be one implementation of router module 120 such as described above with reference to FIG. 1. In that regard, management node 2000 may generally comprise a router 122 portion, a router memory 124 portion (which, as noted above, may include one or more router memory components 124a, 124b, 124c, ... 124n), and one or more node interfaces or communications links (not illustrated, but described above with reference to FIG. 1). It is noted that attendant power supplies, bus structures, and memory controller elements are omitted from FIG. 20 for clarity, but are well-known in the art.

[00188] Management node 2000, in general, may also include a host interface 126 to communicate with a host compute system 199. As set forth above, host interface 126 may be operative in accordance with PCI or PCIe hardware and associated communications protocols such as Ethernet™, SATA, or other suitable communications technologies. The specific hardware and communication strategy embodied by or incorporated into host interface 126 may be selected in accordance with bandwidth requirements, cooperating hardware at host compute system 199, functionality and operational characteristics of router 122 portion of management node 2000, or a combination of these and a variety of other factors.

[00189] As set forth above with reference to router module 120, one or more node interfaces or communications links may be implemented as a network interface card, a wireless or inductive hardware component, or a communications port, jack, or plug, for instance, and may enable router 122 portion of management node 2000 to communicate with a respective compute node 140, 1800, and 1900 as is generally known in the art. In use, router 122 portion may be operable to transmit data packets between host compute system 199 (via host interface 126) and a respective compute node 140, 1800, 1900 (via an appropriate node interface) in cooperation with routing information (destination address, packet header data, flood domains, *etc.*) and other short term data (look up tables, resource availability, performance or load balancing data, *etc.*) stored in or accessible by router memory 124 portion. Router memory 124 portion may be embodied in or comprise, by way of example, volatile memory such as RAM, SRAM, DRAM, DDR RAM, and the like. Router memory 124 portion, in general, or individual memory components such as 124a – 124n, in particular, may employ dedicated or dynamically allocated memory storage areas or buffers (such as result buffers 2091) to support algorithm processing in high-demand applications. As noted above, the disclosed subject matter is not to be limited to any particular memory management or resource allocation strategy, as many techniques and hardware architectures may be employed to facilitate the functionality described herein.

[00190] In one implementation, router 122 portion may execute all routing functionality (such as described above with reference to FIG. 1) using, or in cooperation with, algorithm program logic (APL) 2022 embedded in an algorithm hardware and/or firmware framework 2021. As with APL 1890 described above, APL 2022 may employ or comprise any of various commercial embodiments of an FPGA that may be selected in accordance with any of a number of factors or design choices. In this case, APL 2022 may be selected to accommodate operation of node 2000 in connection with algorithm processing as described below, in addition to support other required

or desired functionality of router 122 portion set forth above. As many types of hardware implementations are generally available, the structure and operation of router 122 portion and router memory 124 portion of management node 2000 are not described further.

[00191] FIG. 21 is a functional block diagram illustrating one implementation of an algorithm processing system using a management node. FIG. 21 depicts one arrangement utilizing a matrix (of size n X t) of distributed processing resources, wherein the matrix comprises an arbitrary number of processing pipelines (1, 2, ... n), each of which includes an arbitrary number of processing tiers (1, 2, ... t). Other arrangements, for instance, in which the pipelines do not have the same number of tiers, may also be employed in some circumstances as a design choice or as a function of computational requirements or other system parameters. In some implementations, the number of pipelines may be limited to four, eight, or sixteen, for example, depending upon the nature and operational characteristics of the nodes 140, 1800, 1900, and 2000, for instance, or in accordance with processing capabilities, power consumption, size requirements, economic, or other constraints.

[00192] As illustrated in FIG. 21, each processing pipeline (1 through n) may employ a Flash/Compute node 1800 at tier 1 (in some of the description that follows, such a Flash/Compute node 1800 may be referred to as a "memory-supported" compute node, owing to the fact that it has access to, or may incorporate, memory store 143). In some implementations, the tier 1 node 1800 in each pipeline is assigned a particular algorithm processing operation, and may use an associated memory store 143 to support such an assigned operation. To facilitate such processing, however, in addition to making use of available local resources and processing capabilities, node 1800 may employ (*e.g.*, via delegation of tasks, passing of interim results, or using alternative resources downstream in its particular pipeline) additional resources in downstream tiers (tiers 2 through t, for example). In the example illustrated in FIG. 21, each pipeline employs Compute nodes 1900 in tiers 2 through t; as noted above, each node 1900 may omit on-board or accessible memory stores 143, while still having computational capabilities sufficient to support activities delegated or requested by a respective upstream node 1800 (or node 1900, as the case may be) in its particular pipeline. It will be appreciated that a node 1900 at a particular tier in a particular pipeline may be replaced with or supplemented by a node 1800 having access to memory store 143 where desirable or appropriate for a given application or system configuration. Further, it is noted that FIG. 21 has been simplified to show only a single set of slaved, serialized Compute nodes 1900 per Flash/Compute node 1800 in a particular

pipeline; this may not always be the case, as a single Flash/Compute node 1800 may support multiple, parallel pipelines of Compute nodes 1900 in some circumstances, depending upon, for example, the nature, operational characteristics, and bandwidth of the communications links 149 employed, the sophistication of compute arrays or algorithm frameworks 142, 2021 and their respective encodings, or a combination of these and a variety of other system parameters.

[00193] At the terminus of each processing pipeline, a terminal node (*i.e.*, at tier t in FIG. 21) may provide results to a management node 2000 as illustrated in FIG. 21. Though the illustrated arrangement depicts each respective pipeline feeding results to a single management node 2000, other possibilities are contemplated. Similarly, the management node 2000 at the terminus of each pipeline may be the same management node 2000 that originally assigned the processing tasks to the Flash/Compute nodes 1800 at tier 1 of the pipelines, or it may also be a different management node 2000 in other applications or hardware architectures.

[00194] FIG. 22 is a functional flow diagram illustrating aspects of one implementation of a method of executing data processing operations. As indicated in FIG. 22, one method of executing an algorithm in a distributed processing compute environment may begin by providing an initiating compute node comprising a memory store and a first node memory (see block 2201). In one such implementation, an initiating node may be embodied in or comprise a Flash/Compute node 1800 at tier 1 of a pipeline as described above with reference to FIGS. 18 and 21. This initiating compute node may engage in bi-directional data communication with a host compute system 199 substantially as set forth above.

[00195] A method may continue, as indicated a block 2202, by serially coupling one or more additional compute nodes to the initiating compute node to establish an execution pipeline comprising the initiating compute node and the one or more additional compute nodes, each respective one of the one or more additional compute nodes comprising a respective second node memory. In some implementations, the one or more additional compute nodes may not include or have access to a memory store; in such circumstances, these additional compute nodes may be embodied in or comprise Compute nodes 1900 such as those described above with reference to FIGS. 19 and 21 or similar hardware constructs.

[00196] The execution pipeline may be initiated by loading a bit stream comprising an instruction set related to execution of the algorithm into a respective programmable logic space of each respective compute node in the execution pipeline (block 2203). As noted above, loading

a bit stream into each compute node responsible for execution of the algorithm may be desirable in many instances, providing efficiencies and increasing data flow through a particular pipelined architecture. In that regard, it is noted that information in the bit stream may be used in several operations depicted in FIG. 22, as well as in other methods operative in accordance with the disclosed subject matter.

[00197] For algorithm processing, a method may continue with executing a first operation associated with the algorithm at the initiating node using information in the bit stream and data retrieved from the memory store, and passing results of the first operation to a first additional compute node in the execution pipeline (block 2204). As noted above, some implementations may execute this functionality without storing the results of the first operation in the first node memory.

[00198] As described above with specific reference to FIG. 21, the first additional compute node and each subsequent additional compute node in the execution pipeline may execute a respective additional operation associated with the algorithm using information in the bit stream and results of a respective preceding operation, and may then pass respective additional results of the respective additional operation to a next additional compute node in the execution pipeline (block 2205). As with block 2204, in some circumstances, it may be desirable that the operations depicted at block 2205 occur without storing the additional results in the respective second node memory of each respective additional compute node.

[00199] In accordance with the disclosed subject matter, a method may conclude by terminating execution of the algorithm at a management node that receives the additional results from a terminal additional compute node in the execution pipeline, the terminating comprising executing a final operation to produce a final result of the algorithm (block 2206). The management node may be embodied in or comprise a Management node 2000 substantially as described above with reference to FIGS. 20 and 21, and may execute the functionality described above.

[00200] As with FIGS. 16 and 17, the arrangement of the blocks and the order of operations depicted in FIG. 22 are not intended to exclude other alternatives or options. For example, it will be appreciated that in accordance with one embodiment, the operations depicted at blocks 2201 and 2202 may be executed substantially simultaneously, or may be integrated into a single operation. As another example, the operation depicted at block 2203 may be executed

concomitantly with the operations at block 2201, 2202, or both, where compute nodes are selected in real-time or near real-time as a function of execution requirements, for example, which may be determined by the sophistication of the bit stream. Further, those of skill in the art will appreciate that the operations depicted at blocks 2204 and 2205 may occur substantially simultaneously, as a subsequent or downstream compute node is executing a next step while one or more upstream nodes are executing preceding steps in an algorithm processing operation. These and other such alternatives may readily be effectuated without materially impacting results of the method or the functionality of any particular hardware implementation utilized to execute the method. In addition to the alternatives set forth in detail above, various design choices that may influence the order or arrangement of the operations depicted in FIG. 22 will be readily apparent to those of skill in the art.

[00201] Those of skill in the art will appreciate that the foregoing distributed processing system may be particularly suited to complex algorithm processing operations, and that the platform architecture may provide performance and efficiency advantages in many applications. In particular, the disclosed subject matter may have utility in connection with a pipeline architecture machine in accordance with which algorithm execution parameters may be programmed as processing pipelines using various facilities implemented in hardware and firmware resident at or accessible by the nodes 140, 1800, 1900, and 2000 described herein. In some implementations, for example, data may be streamed from a memory component (such as memory store 143) into an execution pipeline (such as depicted in FIG. 21) generally at the maximum clock rate of the algorithm framework 142, 2021 processing components that form or support the nodes in the pipeline.

[00202] As set forth above, an aspect of the disclosed subject matter that facilitates implementing an algorithm to execute in a particular processing node is the compute array or algorithm framework integrated into the programmable logic 410 space (see FIGS. 4 and 18-20) of the Flash/Compute nodes 140, 1800, or, when a module or card (see FIGS. 3B and 3C) is so configured, its Compute nodes 1900 or management node 2000. As noted above, nodes 1900 and 2000 may simply be implemented as Flash/Compute nodes 140, 1800 with no memory store 143 component, and the former may serve in such an arrangement as additional compute accelerators.

[00203] In operation of an algorithm processing application, most single pass algorithms may be distributed as binary bit streams to be loaded and executed in the programmable logic 410

space of one or more nodes 1800 (see FIGS. 4 and 18).More complex algorithms may also utilize other facilities or resources of a particular node or a combination of nodes (140, 1800, 1900, or 2000) such as is illustrated in the pipelines of FIG. 21. These other facilities or resources may include: programmable logic 2022 space of a management node 2000; node memory 144 at nodes 1800, 1900, router memory 124 at node 2000, or both; one of more processors 439, 1871 at a node 1800, 1900; or a combination of the foregoing or other resources.

[00204] Thus, to execute a given algorithm, a bit stream loaded into each node 1800, 1900 may also be accompanied by a bit stream for a management node 2000 that may be tasked with overseeing or managing the relevant or related pipeline (see FIG. 21) and/or executable code to be processed by a local processing system 148, 430. Whether implemented by a single node based bit stream or by a more complex combination of node 1800, 1900 and management node 2000 facilities, all algorithms may be managed by firmware and a proprietary algorithm framework (PAF) code that is part of every bit stream and complements an algorithm's specific bit stream in the programmable logic 410, 2022 space in a given node. For example, FIG. 20 illustrates some functional blocks of a management node 2000 and its facilities, including APL 2022 integrated with or embedded in algorithm framework 2021. These facilities may be operative in accordance with the present disclosure to support the functionality described herein, as will be appreciated by those of skill in the art.

[00205] From an application programmer point of view, the complexities of this approach to algorithm processing may be completely ignored. In that regard, application programmers may simply specify the algorithm to be executed, provide parameters for the algorithm (including the name and selection criteria for the dataset upon which the algorithm is to be executed), and request execution. It is noted that an application programmer may also identify datasets that are related to the named dataset, or that the disclosed algorithm execution methodologies may identify such related datasets without input, depending upon the manner in which the datasets are identified as related in the memory store paradigm. In this manner, a single instantiation of an algorithm execution process may operate on multiple datasets efficiently and without undue programming or experimentation on the part of the application programmer. The facilities or system resources used or required by that algorithm, how nodes 140, 1800, 1900, and 2000 may interact to access data and to execute the algorithm, how much memory is required or where the appropriate memory hardware resides, and other functional and operational details may be entirely transparent to application programmers.

[00206] Some use cases are provided below.

[00207] In the context of the present disclosure, the term "Level 1" algorithm is intended to mean an algorithm that may be implemented in accordance with the present system and method using just the programmable logic 410 space of one or more Flash/Compute nodes 140, 1800 (*i.e.*, those nodes incorporating or having access to a memory store 143). These Level 1 algorithms may make use of a single processing pipeline (say, pipeline n-1 in FIG. 21), which may be replicated and executed in parallel across a plurality of nodes 140, 1800 in the pipeline. A Level 1 algorithm may be complex, large (in terms of number of execution steps), or both; a Level 1 algorithm may even be recursive, but it may generally be characterized by the fact that it has little or no need for intermediate data storage. With the foregoing as context, it will be appreciated that a Level 1 algorithm may interface with a memory store 143 and return results via the PAF (*e.g.*, compute array 142 in FIGS. 4 and 18) under the control of firmware resident in the programmable logic 410 space.

[00208] Examples of such Level 1 algorithms are: basic mathematics and statistics functions such as min, max, average, *etc.*; complex data searches and selections; and more complex mathematical computations such as Fast Fourier Transforms and the like.

[00209] In the context of the present disclosure, the term "Level 2" algorithm is intended to mean an algorithm that generally requires one or more blocks of significant amounts of data storage for intermediate results between the various steps in an associated processing pipeline. It is noted that "significant amounts" of data storage may vary as a function of overall processing requirements, hardware capabilities, throughput or bandwidth thresholds, and other system parameters. In that regard, a typical Level 2 algorithm may be characterized as such because, in order to satisfy or to exceed some performance threshold or processing bandwidth requirement, which may be pre-determined or dynamically calculated or adjusted, the algorithm must or beneficially should: pass intermediate results to slaved nodes 1900; store those intermediate results in a temporary data store for use in connection with subsequent steps within the overall algorithm execution; or both. Specifically, a Level 2 algorithm in this context may be characterized as a series of execution layers in which each execution layer essentially operates on a Level 1 algorithm, the output (*e.g.*, a significant quantity of data) of which is necessary for operations to occur at a subsequent execution layer. In accordance with the disclosed subject matter, such Level 2 algorithms may be executed in a variety of ways.

SUBSTITUTE SPECIFICATION
Version with Markings to Show Changes Made

[00210] In one implementation, for example, a Level 2 algorithm may be approached as a series of simple (*i.e.*, Level 1) algorithms stored within the programmable logic 410 space of a Flash/Compute node 140, 1800, each of which may be configured and operative to feed its output into node memory 144 (of the associated node 140, 1800); the input for each such simple algorithm at each tier in a particular pipeline may be stored results read from either the associated memory store 143 or the node memory 144 at a previous layer (*i.e.*, from the node 140, 1800 in a preceding tier in the pipeline).

[00211] As another example of implementation, a Level 2 algorithm may be approached as a series of simple algorithms stored with the programmable logic 410 space of a Flash/Compute node 140, 1800, and also stored in the programmable logic 410 space of one or more Compute nodes 1900; in this embodiment (illustrated in FIG. 21), each node 1800, 1900 may feed intermediate results or data to the next to form a pipeline. Each node 1900 in a particular pipeline may be slaved to the node 140, 1800 at tier 1, and each particular pipeline may ultimately terminate at the APL 2022 of a management node 2000.The final result of the algorithm execution may be computed (*e.g.*, at management node 2000) and returned to a host compute system 199. In this case, instead of writing intermediate results to short-term storage (*e.g.*, at node memory 144), such results may be transmitted or handed off by the $i\text{-}1^{th}$ node (1800, 1900, as the case may be) to the i^{th} node in the slaved pipeline (see FIG. 21). As the cascade of intermediate results propagates through the pipeline, the tier 1 node 1800, for example, may then accept the next data element(s) (and any additional arguments) from data store 143 (or its equivalent) while the tier 2 node 1900 continues to execute concomitantly or substantially simultaneously, ultimately passing its output to the tier 3 node 1900 in the pipeline, whereupon the tier 2 node 1900 may then be ready to accept the next input from the initiating tier 1 node 1800, and so on. The terminal node in a given pipeline (*e.g.*, the tier t node 1900 in FIG. 21) may finally pass or transmit its results to the management node 2000 for any final computation steps (*e.g.*, using APL 2022), after which a final result may be buffered for returned to host compute system 199, and each relevant tier 1 node 1800 may be notified that a result is ready to be returned.

[00212] One example of a Level 2 algorithm as contemplated herein is a common inference algorithm for a neural network executing under TensorFlow™ (an open-source platform for machine learning, artificial intelligence training, and the like). As is generally known in the art, neural networks are typically organized into layers (or tiers): all the neurons (or nodes) in a given tier are tasked with executing processing operations using both input data for that tier (and per

node, as directed by the network) and a coefficient (as supplied by the network) for each node in the tier. The output data or results of the inference processing operations at each node may be quite large; typically, these data are transmitted or fed to the next subsequent tier downstream in the processing pipeline, after which they may be discarded by the node transmitting the results. The final or terminal tier's output (or a computation based upon output from all the nodes in the terminal tier) is the result of the inference, and may be returned to the application that requested the inference that instantiated the algorithm processing in the first place. In essence, then, the basic TensorFlow computational element may be considered a Level 1 algorithm that accepts primarily two inputs or arguments: a coefficient which is computed from results of original training of the particular relevant neural network; and an element of data, which is either taken from (in the case of the first layer or tier of the neural network) the dataset that is the target of the inference process, or from (in the case of a subsequent layer or tier of the neural network) output data from a previous tier in the neural network.

[00213] In the context of the present computing architecture such as illustrated in FIGS. 1 and 4, for instance, the foregoing pipelined processing methodologies may be implemented to run exclusively within a series of Flash/Compute nodes 140, 1800, for example, by storing intermediate results from processing within a given tier in node memory 144 of an applicable node 140, 1800. While this approach may achieve satisfactory performance for most applications, it may not achieve the best possible performance for computationally expensive pipelined processing since it may not, under certain circumstances, fully leverage a particular node's (or tier's) available assets. For example, since the executable code (in compute array 142 and processing system 148, for instance) must store and retrieve intermediate results from node memory 144, the streaming capabilities of the architecture may not be optimized to achieve maximum throughput, as reads from data store 143 must wait during node memory 144 access cycles. As noted above, in many processing scenarios, even in a Big Data context involving high performance computing and massive datasets, the architecture illustrated in FIGS. 1 and 4 may be adequate to provide sufficient performance and data throughput rates. For applications in which performance and data throughput are critical, however, it may be desirable to combine a Flash/Compute node 140, 1800 with a Compute node 1900, for example, such as in the architecture illustrated in FIG. 21.

[00214] As noted above with reference to FIG. 19, some embodiments of a Compute node 1900 may omit data store 143. In addition to economizing on silicon real estate, power

consumption, manufacturing costs, and the like, utilization of node 1900 may enable or facilitate implementation of an overall system architecture that resembles that depicted in FIG. 21. Such a system architecture or hardware arrangement may link a single, particular Flash/Compute node 1800 (which does incorporate or have access to memory store 143) to a pipeline of serially coupled Compute nodes 1900, each of which may be uniquely associated with that particular Flash/Compute node 1800. In the FIG. 21 example, tier 1 comprises node 1800 (*i.e.*, the "initiating node"), and every successive tier (2, 3, … t) comprises a node 1900. The illustrated architecture, where appropriately coded and leveraged to increase or to maximize data throughput, may have particular utility in connection with neural network processing operations.

[00215] In the context of processing the TensorFlow inference algorithm in connection with this hardware architecture, for example, for all nodes in a particular pipeline, the same bit stream implementing the algorithm may be loaded into the programmable logic 410 space of each node 1900, in addition to the programmable logic 410 space of Flash/Compute node 140, 1800. In some implementations, there may be minor differences in the data supplied to node 1800 as compared to those supplied to nodes 1900 in a particular pipeline. For instance, Flash/Compute node 140, 1800 may be programmed not to use its node memory 144 in connection with executing the algorithm, whereas each Compute node 1900 may be instructed or coded to do so before passing or transmitting its output to the next Compute node 1900 in the pipeline. These and any other desired operational parameters may be communicated to nodes 1800, 1900 via the bit stream.

[00216] In operation, the tier 1 initiating Flash/Compute node 1800 may retrieve input for the algorithm processing operation (*e.g.*, from memory store 143), execute required or desired computations for the first layer in the neural network, and transmit computational results or output data to the first Compute node 1900 in the pipeline. Once such computational results or output data are received from the initiating tier 1 node, the first Compute node 1900 in the pipeline (in this case, at tier 2), may begin processing operations at the next layer in the neural network; meanwhile, back at the tier 1 Flash/Compute node 1800, the next element of data to be used as input for the next algorithm processing operation may be retrieved (*e.g.*, from memory store 143) such that Flash/Compute node 1800 may already be executing the next subsequent required or desired computations or processing operations. Data processing responsibilities may propagate downstream through the pipeline, along with output from the previous tier, substantially as illustrated and described above with reference to FIG. 21.

[00217] Each Compute node 1900 in a particular pipeline may handle one or more layers of processing in the given algorithm. Eventually, upon completion of processing at a particular Compute node 1900 in a particular tier of the pipeline, that particular Compute node 1900 may then pass output to the next Compute node 1900 in the next subsequent tier. Those of skill in the art will appreciate that the number of layers for a particular algorithm that may be computed by a given Compute node 1900 in a particular pipeline may depend upon the coding provided in the bit stream, the complexity and structure of the neural network (including the overall architecture, pipeline length, specific implementations of nodes 1800, 1900, bandwidth of the communications links 149, read/write performance of node memory 144 or memory store 143, and the like), sophistication of the algorithm being processed, or a combination of these and other factors. In that regard, the number of Compute nodes 1900 employed in a given pipeline may also depend upon the complexity and structure of the neural network in the context of some or all of the factors listed above. In some instances, it may be useful to balance the ratio of layers (of algorithm processing) per Compute node 1900 with the number of Compute nodes 1900 in a given architecture in such a manner that the associated Flash/Compute node 1800 in a given pipeline may continue to accept data streamed from memory store 143 without needing to stall the read process.

[00218] As noted above, the terminal Compute node 1900 (*e.g.*, at tier t in FIG. 21) in a particular pipeline may transmit its data to a management node 2000; in some implementations, such a management node 2000 may also be supplied with a copy of the algorithm bit stream that instantiated the processing operations in the first place. In operation, management node 2000 generally may execute the final layer (or final several layers) of computation required by the algorithm to produce a result. In some implementations, it may be desirable that a management node 2000 comprises four independent channels to router memory 124; this architecture may enable a four-way parallel computation, for example, such that a given management node 2000 may support up to four pipelines. This is illustrated in FIG. 21, which has been simplified to show only a single set of slaved, serialized Compute nodes 1900 per Flash/Compute node 1800. Other embodiments comprising multiple pipelines per Flash/Compute node 1800 are contemplated as set forth above, and may be implemented as a design choice or as a function of overall system throughput requirements.

[00219] In operation, management node 2000 may buffer computational results or algorithm output for return to host compute system 199 when appropriate or required. When resources at

management node 2000 are taxed or reach a capacity threshold that may be pre-determined or dynamically computed or adjusted (*e.g.*, when result buffers 2091 are full or exceed a threshold or other pre-determined or dynamically computed limit), management node 2000 may signal the algorithm initiating nodes 140, 1800 under its control that computed results are ready for transmission. In some embodiments, one or more applicable Flash/Compute nodes 140, 1800 may initiate the return, *e.g.*, by commanding or instructing management node 2000 to transmit buffered results to host compute system 199; alternatively, management node 2000 may initiate such a data transfer in accordance with coded instructions, *e.g.*, unbidden or otherwise without instructions or requests supplied by any other system component.

[00220] The following description addresses certain hardware and functional details of some of the system components and implementations described above. In that regard, recall that FIGS. 2A through 2C are functional block diagrams illustrating one example of a hardware environment accommodating components of a data processing and analytics system. These hardware environments may be referred to as "cards" in some instances, such as in the description of FIGS. 2A through 2C; additionally, those of skill in the art will appreciate that such cards may also be referred to as "modules" in some industries or for some applications. In the description that follows, the term "module" may be used as a synonym for "card" as that term is used in reference to FIGS. 2A through 2C.

Program Logic 410 Space

[00221] In some implementations, a single implementation of a card or module may generally comprise two types of FPGAs or processing components.

[00222] The first may be embodied in or comprise a straightforward FPGA or other processing component that handles all the functionality of a management node 2000 (FIG. 20) or router module 120 (FIG. 1) set forth above. In the illustrated implementations, there is only one such processing component per module. To accommodate the described functionality, this processing component (for example, algorithm framework 2021 in router 122 portion) may comprise or incorporate a large (in terms of overall silicon real estate or area) programmable logic space (such as APL 2022, for instance) and a large number of I/O lanes. In operation, this FPGA or processing component may generally implement appropriate protocols for messaging to/from the host compute system 199, vector messaging to/from pipelined nodes 140, 1800, and 1900, and other routing and housekeeping functionality substantially as described above.

[00223] The second is an FPGA or other processing component that may generally be implemented as a Multi-Processor System on a Chip (MPSoC). In the illustrated embodiments, every module comprises at least one of these; in some embodiments such as described above, a module or card may comprise eight of such processing components, which may operate in parallel (though other configurations are also possible, and may reflect design choices, cost constraints, desired processing characteristics, and the like). In embodiments, each such processing component includes a reasonably large programmable logic 410 space, four or more (*e.g.*, 64 bit) processors or cores 439, two or more (*e.g.*, 32 bit) processors or cores 432, and a sufficient number of I/O lanes to enable both communications with a management node 2000 via communications links 149 as well as high speed connections to node memory 144 and memory store 143 elements, if applicable. Note that some nodes 1900 may also employ the same or similar processing components, but do not need to implement a connection to a memory store 143 component.

[00224] Within each type of FPGA or processing component, an algorithm's use of the available resources while residing in the programmable logic 410 space may be managed by the algorithm framework (142 in FIGS. 18 and 19 or 2021 in FIG. 20). In accordance with some methodologies, other than restrictions imposed by the algorithm framework 142, 2021, an algorithm may be configured and operative to use any or all available logic within the programmable logic space.

[00225] In operation, local firmware (*e.g.*, executing on one of more of the embedded processors 439, 1871) may initialize the programmable logic 410 space upon system boot or as otherwise requested, and may then load a "golden image" or template containing the appropriate or requested algorithm framework 142, 2021. Once loaded and initialized as is generally known, the local firmware may then load and execute algorithms or attendant or associated instruction sets as dictated by messages received from the host compute system 199.

Algorithm Framework 142, 2021

[00226] The algorithm framework 142, 2021 may generally provide two basic categories of services: system utilities; and a data selection engine.

[00227] In some implementations, system utilities may generally include the following: communications link 149 messaging system protocols and support; control of access (*i.e.*, driver support, controller functionality, *etc.*) to node memory 144 and memory store 143 elements, if

applicable; data movers (described above) that may queue and initiate read/write activities for node memory 144 and memory store 143 elements, if applicable; start/stop/resume control for algorithm execution processes; and management of any unhandled condition (such as interrupts and the like) arising from algorithm processing operations.

[00228] In some implementations, data selection engine services may be requested or utilized in cooperation with the data movers, for instance, to implement a flexible selection capability when an algorithm operates on only a subset of data from a target dataset. For example, selections of targeted portions or subsets of data may be made using any of a variety of fields in a dataset, irrespective of whether those particular data are from the fields on which a particular algorithm will ultimately be operating. In such cases, selection of specific subsets of data may be based upon specific fields in the target dataset, fields in a different dataset that includes indices to the target dataset, both, or a combination of these and other factors or system parameters related to or arising from the applicable algorithm and processing boundary conditions. It is noted that co-location of data or datasets, as well as providing an option for a particular algorithm to execute on multiple datasets, complement these features and may result in increased computational and data access efficiencies in may applications.

[00229] In some embodiments, it may be desirable that the only permitted interaction between algorithms and the algorithm framework facilities of a node 1800, 1900 may be via the algorithm framework 142, 2021 hardware and firmware implementation. In accordance with the disclosed subject matter, this restricted permission paradigm may be a deliberate design choice, as the algorithm framework 142, 2021 may be responsible for preventing algorithms and the constituent hardware and firmware components from corrupting system data and user data maintained in node memory 144 and memory store 143, if applicable. Other alternative embodiments may not be so restricted, however, as will be appreciated by those of skill in the art.

Node Memory 144

[00230] The majority of node memory 144 located on or accessible by a Flash/Compute node 142, 1800 may be dedicated to or otherwise used for managing access to memory store 143, as well as for storing system information required by local firmware that manages operation and overall functionality of the node 142, 1800. Even in light of the foregoing, however, several gigabytes (or more, in some implementations) of node memory 144 may be available for use by Level 2 algorithms, if needed (this is represented by reference numeral 1841 in FIG. 18). Access

to such algorithm memory 1841 may be achieved via the PAF or via interaction with one or more cores 439 or 1871.

[00231] In some embodiments of a Compute node 1900, a minority or relatively small portion of local node memory 144 (typically, only a few gigabytes) may be used for system information and intra-node messaging operations required by local firmware. The remainder of the available capacity (typically, 16 gigabytes or more) of a node memory 144 storage component (*i.e.*, algorithm memory 1841) may thus be made available for Level 2 algorithm execution operations, for example: buffering intermediate results for use within the node 1900 itself; buffering a final intermediate result for transmission to the next node 1900 in the pipeline or to the management node 2000; storing data relevant to a previous or subsequent neural network processing layer; and the like.

[00232] In typical Flash/Compute node 1800 as set forth herein, an algorithm using local node memory 144 generally may slow down the streaming of data from memory store 143 into the programmable logic 410 space as required during algorithm processing operations. Such an intentional governing of data flow may be implemented as a design choice, since the data movers (described above) represent shared resources for both functions (*i.e.*, retrieving data from memory store 143 and managing intra-node data transfers with respect to node memory 144 and other components of node 1800). In some situations, this may affect throughput in a manner that is measureable or significant, which is one of the motivations to implement the architecture depicted in FIG. 21 for high-throughput, computationally expensive applications in which it may be useful to regulate or to limit access to memory store 143.

Processing Cores

[00233] Local firmware running on either a Flash/Compute node 140, 1800, on the one hand, or on Compute node 1900, on the other hand, may generally utilize one or more processors 439, 1871 or cores (see processing system 430 in FIG. 4). As is generally known in the art, such firmware may employ multiple cores, for instance, to increase performance under normal loads, or it may throttle back and dedicate fewer cores to intra-node work in the event that a Level 2 algorithm requires the use of one or more processors 439, 1871 for high load situations.

[00234] In some implementations, it may be desirable to run a host Linux operating system in connection with any processor 432, 439, 1871 that is being used by, or that is managing, an algorithm processing operation or any application-specific code. Access to the various facilities

of an applicable node 142, 1800, 1900, and 2000 may be strictly controlled by such a host operating system, and Linux is one operating system that provides certain protections for user and system data that are located in node memory 144 and memory store 143 that are not available in other commercially available operating systems. Specifically, using Linux, direct access to node memory 144 and memory store 143 elements may be strictly regulated in a manner that is not readily implemented using other operating systems. In some circumstance, the EMMC memory component 1870 integrated into or accessible by a Flash/Compute node 1800 and a Compute node 1900 may be utilized for maintaining data and instruction sets associated with the host operating system, for instance, and to act as a "hard drive" for the local node 1800, 1900 when needed or desired. Such activity may be limited to a specific partition on the memory component 1870 if desired, as is generally known in the art.

Management Node 2000 Facilities

[00235] A management node 2000, as previous described, may have a distinct architecture and specific facilities that are available to accommodate processing of Level 2 algorithms in some situations. In some implementations, for instance, a management node 2000 may include or have access to up to four times (or more) the local memory capacity (*e.g.*, router memory 124) of other nodes (such as nodes 140, 1800, and 1900) and no memory store 143 component; additionally or alternatively, it may also not have any local processors or processing cores.

[00236] In the illustrated embodiments (provided by way of example only) a management node 2000 may generally be controlled by its own local version of the algorithm framework 2021, which may include extensions or appropriate modifications suitable to enable bi-directional data communication with the host compute system 199 (as set forth above with reference to FIG. 1) and any necessary functional blocks to implement necessary PCIe or other communications protocols (or other applicable protocols for interfacing with any other communication fabric that is generally known in the art or developed in accordance with known principles).

[00237] In the context of algorithm processing operations utilizing the programmable logic 2022 space of a management node 2000, for example, a single FPGA or other processing component resident at or integrated into the management node 2000 may be made available via algorithm framework 2021, and may be loaded and made available in the event that a Level 2 algorithm, which may require use of its resources, is selected for execution.

[00238] In the illustrated arrangement, router memory 124 integrated into a management node 2000 may be implemented as four or more separate DRAM modules (reference numerals 124a through 124n in FIGS. 1 and 20), though other configurations may have utility in some situations. In one implementation, each router memory 124 module may be sixteen gigabytes in capacity, and may be coupled to the local algorithm framework 2021 via a dedicated or individual direct memory access (DMA) or other channel. In that regard, a particular management node 2000 may alternatively function effectively as four (or more) virtual Compute nodes 1900 that each have programmable logic 2022 space only. In addition, the memory usage requirements for a given management node 2000 (in terms of overall system operation) may be generally low, so the vast majority of the total memory capacity resident at a management node 2000 may be made available to an algorithm executing in its programmable logic 2022 space.

[00239] A management node 2000 may also be capable of communicating directly with any and all other nodes 140, 1800, and 1900 connected to it via a pipeline or other distributed architecture hardware construct. These communications facilities may generally be used at any point during algorithm execution, though it is most likely that inter-node communications involving a management node 2000 will be related to or associated with a final or terminal operation at the end of execution or when data buffers or other temporary data storage components (such as result buffers 2091) reach some pre-determined capacity, at which point it may be necessary or desirable to initiate a return of data to the host compute system 199.

[00240] Several features and aspects of a system and method have been illustrated and described in detail with reference to particular embodiments by way of example only, and not by way of limitation. Those of skill in the art will appreciate that alternative implementations and various modifications to the disclosed subject matter are within the scope and contemplation of the present disclosure. Therefore, it is intended that the present disclosure be considered as limited only by the scope of the appended claims.

SUBSTITUTE SPECIFICATION
Version with Markings to Show Changes Made

What is claimed is:

CLAIMS

1. A data processing system operative in cooperation with a host compute system, the data processing system comprising:

a router module comprising a host interface and a node interface, wherein the router module is communicatively coupled to the host compute system via the host interface; and

multiple parallel first tier compute nodes coupled to one or more additional compute nodes in series, each compute node comprising:

a communications link, wherein each first tier compute node is communicatively coupled to the node interface at the router module via the communications link, at different successive tiers, each compute node in a series being communicatively coupled, via its communications link to one or more additional compute nodes connected at different successive tiers;

a data store comprising records associated with a data processing operation;

a programmable logic component to execute the data processing operation in cooperation with the data store;

a node memory comprising data and instructions to support operation of the programmable logic component in connection with the data processing operation and to facilitate data communications via the communications link;

a data mover component to facilitate data communications between the programmable logic component and the node memory; and

a storage interface component to facilitate data communications between the programmable logic component and the data store;

wherein the storage interface component utilizes a plurality of communications channels to transfer data between the programmable logic component and the data store.

2. The data processing system of claim 1 wherein the communications link employs a serial communications protocol.

Formatted: Line spacing: 1.5 lines

3. The data processing system of claim 1 wherein the data store comprises an Electronically Erasable Programmable Read Only Memory device.

4. The data processing system of claim 3 wherein the storage interface component employs and Open NAND Flash Interface protocol.

5. The data processing system of claim 1 wherein the programmable logic component comprises a field programmable gate array.

6. The data processing system of claim 1 further comprising a processor to manage operation of the data mover component.

7. The data processing system of claim 6 wherein the data mover component comprises a device controller.

8. The data processing system of claim 1 wherein the host compute system executes instructions to influence operation of the compute node.

9. The data processing system of claim 1 wherein the router module and the compute node are integrated into an interface card communicatively coupled to the host compute system.

Formatted: Line spacing: 1.5 lines

Formatted: Line spacing: 1.5 lines

Formatted: Line spacing: 1.5 lines

Formatted: Line spacing: 1.5 lines

Formatted: Line spacing: 1.5 lines

Formatted: Line spacing: 1.5 lines

SUBSTITUTE SPECIFICATION
Version with Markings to Show Changes Made

ABSTRACT

A system and method employing a distributed hardware architecture, either independently or in cooperation with an attendant data structure, in connection with various data processing strategies and data analytics implementations are disclosed. A compute node may be implemented independent of a host compute system to manage and to execute data processing operations. Additionally, an unique algorithm architecture and processing system and method are also disclosed. Different types of nodes may be implemented, either independently or in cooperation with an attendant data structure, in connection with various data processing strategies and data analytics implementations.

Electronic Acknowledgement Receipt

EFS ID:	47315245
Application Number:	18075576
International Application Number:	
Confirmation Number:	1074
Title of Invention:	Analytics, Algorithm Architecture, and Data Processing System and Method
First Named Inventor/Applicant Name:	Robert Bismuth
Customer Number:	27799
Filer:	Victor Jay Castellucci/Maria Lima
Filer Authorized By:	Victor Jay Castellucci
Attorney Docket Number:	FERM0003CON200564319
Receipt Date:	04-JAN-2023
Filing Date:	06-DEC-2022
Time Stamp:	14:23:31
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Transmittal Letter	1FINAL_FERM0003CON2_564319_FOR_FILING.pdf	47261 1826f03420c976d8364d0693e9ce25240a92cf5b	no	6

Warnings:

Information:					
2	Application Data Sheet	FINAL_Corrected_ADS_Corrected.pdf	131515 cebf615fd1fe01135f65e37937b962e7031264bc	no	7
Warnings:					
Information:					
This is not an USPTO supplied ADS fillable form					
3	Specification	FINAL_564319_Specification_EDITED_MARKED_FINAL_FOR_FILING.pdf	825392 8b5f55c4ffa5ce2a99c8232f55b03b0b91175fa3	no	75
Warnings:					
Information:					
4	Specification	FINAL_FERM0003CON2_564319_Specification_EDITED_CLEAN_FINAL_FOR_FILING.pdf	788106 8ed87c3ac1930216149bb2cbc9355017420147a3	no	75
Warnings:					
Information:					
		Total Files Size (in bytes):	1792274		

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 a 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.

DOCKET NO: FERM0003CON2.564319 **PATENT**
Application No. 18/075,576
Reply Under 37 CFR § 1.111

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:	**Robert Bismuth et al.**	Confirmation No.:	**1074**
Serial No.:	**18/075,576**	Group Art Unit:	**_NA_**
Filing Date:	**December 6, 2022**	Examiner:	**_NA_**
Docket No.:	**FERM0003CON2.564319**		
For:	ANALYTICS, ALGORITHM ARCHITECTURE, AND DATA PROCESSING SYSTEM AND METHOD		

FILED VIA EFS-WEB
Date Filed: January 4, 2023

Commissioner for Patent
P.O. Box 1450
Alexandria, VA 22313-1450

Commissioner:

Applicant Response to Pre-Exam Formalities Notice

In response to the outstanding Notice to File Corrected Application Papers dated December 21, 2022 (the "Formalities Notice") and the Response to Request for Corrected Filing Receipt dated December 21, 2022 (the "Benefit Claims Notice"), reconsideration is respectfully requested in view of the amendments and/or remarks as indicated below:

A **Listing of Claims** begins on page 2;
Remarks begin on page 4.

The following listing of claims supersedes all other listings of claims in the present application. Prior to examination on the merits, please amend the claims as follows:

Listing of Claims

What is claimed is:

1. (Original) A data processing system operative in cooperation with a host compute system, the data processing system comprising:

a router module comprising a host interface and a node interface, wherein the router module is communicatively coupled to the host compute system via the host interface; and

multiple parallel first tier compute nodes coupled to one or more additional compute nodes in series, each compute node comprising:

a communications link, wherein each first tier compute node is communicatively coupled to the node interface at the router module via the communications link, at different successive tiers, each compute node in a series being communicatively coupled, via its communications link to one or more additional compute nodes connected at different successive tiers;

a data store comprising records associated with a data processing operation;

a programmable logic component to execute the data processing operation in cooperation with the data store;

a node memory comprising data and instructions to support operation of the programmable logic component in connection with the data processing operation and to facilitate data communications via the communications link;

a data mover component to facilitate data communications between the programmable logic component and the node memory; and

a storage interface component to facilitate data communications between the programmable logic component and the data store;

wherein the storage interface component utilizes a plurality of communications channels to transfer data between the programmable logic component and the data store.

2. (Original) The data processing system of claim 1 wherein the communications link employs a serial communications protocol.
3. (Original) The data processing system of claim 1 wherein the data store comprises an Electronically Erasable Programmable Read Only Memory device.
4. (Original) The data processing system of claim 3 wherein the storage interface component employs and Open NAND Flash Interface protocol.
5. (Original) The data processing system of claim 1 wherein the programmable logic component comprises a field programmable gate array.
6. (Original) The data processing system of claim 1 further comprising a processor to manage operation of the data mover component.
7. (Original) The data processing system of claim 6 wherein the data mover component comprises a device controller.
8. (Original) The data processing system of claim 1 wherein the host compute system executes instructions to influence operation of the compute node.
9. (Original) The data processing system of claim 1 wherein the router module and the compute node are integrated into an interface card communicatively coupled to the host compute system.

REMARKS

Status of the Claims

Claims 1-9 are all the claims pending in the application. As indicated in the Listing of Claims above, claims 1-9 are all "Original," and have not been amended; no new matter has been added. The claims section of the specification, however, has been amended so as to place the claims in compliance with the Office's spacing and formatting requirements. Due to an inadvertent formatting error, the original specification was submitted with the claims section originally presented in single space (1.0) formatting; as "amended," the claims are now presented in the Substitute Specification (submitted herewith) in space and a half (1½) formatting.

Applicant respectfully submits that the claims are allowable as written, and requests consideration and confirmation of same at the Office's earliest convenience.

The Benefit Claims Notice

The Benefit Claims Notice dated December 21, 2022, indicates that the application reflects improper benefit claims to prior-filed provisional applications. An amended Application Data Sheet (ADS) is submitted herewith to reflect proper benefit to the three expired provisional applications identified in both the original ADS and in paragraph [0001] of the originally filed specification. It is also noted that paragraph [0001] of the specification has been amended (*see* the Substitute Specification – Version with Markings to Show Changes Made) to make clear which co-pending application originally claimed the earlier filing dates of the three expired provisional applications out of an abundance of caution, but was correct as originally filed.

Applicant requests that the Office acknowledge the proper priority claims and issue a Filing Receipt reflective of the originally filed paragraph [0001] and the newly amended ADS.

The Formalities Notice

The Formalities Notice identified two items that the Office wanted to be corrected (*see* page 1: "Specification Items" and "Drawing Items").

Regarding the "Specification Items"

The Office noted that the line spacing for the claims section was not "1½ or double spaced" as a result of the inadvertent formatting error noted above. In that regard, the Formalities Notice was not exactly clear as to how an amendment to the claims section might be received in the context of 37 CFR § 1.121 (*i.e.*, "Manner of Making Amendments"), noting only that a "proper claim listing, having the proper line spacing throughout, is required to satisfy this requirement."

For the avoidance of doubt, Applicant has both (a) submitted a new Listing of Claims as set forth above at pages 2 and 3, identifying each of the claims as "Original" and presenting the entire set of claims in 1½ line spacing format and (b) presented an entire claim section at the end of a Substitute Specification (presented both as "Clean" and in a "Version with Markings to Show Changes Made," both of which are) submitted herewith for the Office's review and approval. Applicant reiterates that neither the text nor the substance of the claims has changed, and that consequently no new matter has been added. Further, Applicant submits that the claims are now in compliance with the Office's formatting requirements, and are in condition for examination and allowance as presented.

Regarding the Drawing Items

The Office noted that " . . . the specification . . . is missing the description of Figure(s) 14 as required . . . ." Without conceding this point, Applicant notes that while this is not a deficiency in the drawing figures, it is readily addressable through a minor amendment to the specification at paragraph [0030]; newly amended paragraph [0030] mirrors the language in original paragraph [0029] (which was not objected to), and merely places the written description in a form that is commensurate in scope with the originally filed drawing figures (so no new matter has been added). Applicant submits that the Substitute Specification submitted herewith (presented both as "Clean" and in a "Version with Markings to Show Changes Made") places the specification in compliance with the various rules identified by the Office.

Conclusion

Based upon the remarks set forth above and the documents submitted in connection with this Reply (including the Substitute Specification and the amended ADS), it is respectfully submitted that the present application is now in condition for allowance. Prompt and favorable action to this effect and early passing of the application to issue are respectfully requested. The Examiner is invited to contact the undersigned to discuss this application if the Examiner believes that such discussion may be productive in advancing the application.

No fees are believed to be necessary at this time. Should any fees be required in connection with this Reply, however, the Office is authorized to charge any underpayment or to credit any overpayment to Deposit Account No. 50-3111.

Date: January 4, 2023

Respectfully submitted,

/Victor Castellucci/

Victor Castellucci
Attorney for Applicant
Reg. No. 43,535

COZEN O'CONNOR

1200 19th Street NW
Washington, DC 20036

SUBSTITUTE SPECIFICATION
Clean Version

Analytics, Algorithm Architecture, and Data Processing System and Method

CROSS-REFERENCE TO RELATED APPLICATIONS

[0001] This application is a continuation of, and claims priority to, co-pending United States patent application Serial No. 17/666,049, filed February 7, 2022, which is a continuation of United States patent application Serial No. 16/729,727, filed December 30, 2019, which claims the benefit of the following United States provisional patent applications: Serial No. 62/801,839, filed February 6, 2019; Serial No. 62/853,764, filed May 29, 2019; and Serial No. 62/882,044, filed August 2, 2019. The disclosures of each of the foregoing applications are hereby incorporated herein by reference in their entireties.

FIELD OF THE DISCLOSURE

[0002] Aspects of the disclosed subject matter relate generally to high performance data processing, and more particularly to a system and method employing a distributed hardware architecture, either independently or in cooperation with an attendant data structure, in connection with various data processing strategies and data analytics implementations; additionally or alternatively, aspects of the disclosed subject matter relate to an unique algorithm architecture and processing system and method that may be implemented, either independently or in cooperation with an attendant data structure, in connection with various data processing strategies and data analytics implementations.

BACKGROUND

[0003] Recently, "Big Data," high performance computing, and solid state device technologies have become increasingly important in many contexts, such as in connection with machine learning and artificial intelligence projects, for instance. In accordance with conventional computing strategies, solid state devices (such as Flash™ memory and other data storage hardware media), having no moving parts or motors and having relatively short access/read/write times as compared to traditional spinning media, have been gaining popularity in cycle-intensive applications that benefit from decreased latency in memory access. On the other hand, with the explosion of data available to such systems (as a result, for example, of the nascent Internet of Things ("IoT"), distributed memory systems, and other processing paradigms involving devices sharing data with other devices), the sheer volume of available data to process is increasing faster than traditional hardware and software systems are able to evolve in order to process those data in a meaningful and efficient manner.

[0004] Further, most conventional systems designed for high throughput data processing and analytics rely upon exhaustive (or "brute force") approaches that attempt to overpower the magnitude of the challenge with overwhelming computational resources, at the expense of cycle time and power consumption. In many instances, the dataset size creates a situation in which a practical or commercially reasonable solution is impossible, or at least prohibitively expense. As a practical matter, it will be appreciated that for as long as the rate at which new data become available for processing continues to outpace the rate at which processing methodologies advance to accommodate the increased size of a given dataset, it will continue to take longer to solve bigger and more complex data processing problems.

[0005] Therefore, there is a need for an improved system and method employing a distributed hardware architecture that facilitates high throughput data processing and analytics solutions for resource-intensive applications; as set forth below, some implementations of such a distributed hardware architecture may be configured and operative to employ a data structure that departs from traditional approaches. Further, there is a need for an improved algorithm architecture and processing system and method for high throughput data processing and analytics solutions for resource-intensive applications. As with the distributed hardware architecture noted above, such an algorithm architecture and processing system may have utility in connection with a non-traditional data structure.

SUMMARY OF THE DISCLOSURE

[0006] The following presents a simplified summary of the disclosure in order to provide a basic understanding of some aspects of various embodiments disclosed herein. This summary is not an extensive overview of the disclosure. It is intended neither to identify key or critical elements of the disclosed embodiments nor to delineate the scope of those embodiments. Its sole purpose is to present some concepts of the disclosed subject matter in a simplified form as a prelude to the more detailed description that is presented later.

[0007] The present disclosure describes a system and method employing a distributed hardware architecture to manage and to execute data processing operations, either independently or in cooperation with an attendant data structure; in the context of this description, a discrete component of such a distributed hardware architecture is generally referred to as a "compute node," though other terms may be used by those of skill in the art. In some implementations, one or more compute nodes may be utilized to manage and to execute data

processing operations independent of, though communicatively coupled with, a host compute system. In addition, the present disclosure describes an algorithm architecture and processing system and method that may be implemented, either independently or in cooperation with an attendant data structure, in connection with various data processing strategies and data analytics implementations that may be distributed across or amongst various compute nodes in a distributed processing environment or paradigm.

[0008] In accordance with one aspect of the disclosed subject matter, for example, a method of reformatting a data structure may generally comprise: receiving a block of data, the block of data comprising a number, y, of original records wherein each original record comprises a number, x, of original fields; for i = 1 to x, defining a new data record, each respective new record comprising y new fields, wherein each respective new field corresponds to a respective i^{th} original field from a respective original record; reformatting the block of data responsive to the defining such that a reformatted block of data comprises the new data records, each comprising a single field type; associating each of the new data records with an index operative to identify a particular new data record in the reformatted block of data; and writing the reformatted block of data to memory.

[0009] In accordance with some implementations, a reformatted data structure may be derived from an original data structure, and the reformatted data structure may generally comprise a plurality of new data records, each of the new data records comprising a plurality of new fields associated with a particular field type defined in the original data structure such that every new field in a particular new record is of the particular field type, wherein the reformatted data structure allows a processing system selectively to read the new data records as original records from the original data structure or as original fields from the original data structure.

[0010] In some disclosed implementations, a method of executing data processing operations may generally comprise: providing a compute node communicatively coupled to a host computer and operative to manage data processing operations independent of the host computer; providing a programmable logic component in the compute node that is configured and operative to execute data processing operations in cooperation with a first memory component; providing a data mover component in the compute node that is configured and operative to facilitate data communications between the programmable logic component and a second memory component; providing a set of instructions that enable the programmable logic component to reformat a block of data comprising original records, each original record comprising a plurality of fields having different

field types, such that a reformatted block of data comprises new records, each new record comprising a plurality of fields of a single field type from the original records; and utilizing a plurality of communications channels to transfer data associated with the reformatted data block between the programmable logic component and the first memory component.

[0011] In accordance with another aspect of the disclosed subject matter, a data processing system may be operative in cooperation with a host compute system, and the data processing system may generally comprise: a router module comprising a host interface and a node interface, wherein the router module is communicatively coupled to the host compute system via the host interface; and a compute node comprising: a communications link, wherein the compute node is communicatively coupled to the node interface at the router module via the communications link; a data store comprising records associated with a data processing operation; a programmable logic component to execute the data processing operation in cooperation with the data store; a node memory comprising data and instructions to support operation of the programmable logic component in connection with the data processing operation and to facilitate data communications via the communications link; a data mover component to facilitate data communications between the programmable logic component and the node memory; and a storage interface component to facilitate data communications between the programmable logic component and the data store; wherein the storage interface component utilizes a plurality of communications channels to transfer data between the programmable logic component and the data store.

[0012] Additionally or alternatively, a data processing system operative in cooperation with a host compute system may generally comprise: a router module comprising a host interface and a node interface, wherein the router module is communicatively coupled to the host compute system via the host interface; and a compute node comprising: a communications link, wherein the compute node is communicatively coupled to the node interface at the router module via the communications link; a data store comprising datasets associated with a data processing operation; a programmable logic component to execute the data processing operation in cooperation with the data store; a node memory comprising data and instructions to support operation of the programmable logic component in connection with the data processing operation and to facilitate data communications via the communications link; a data mover component to facilitate data communications between the programmable logic component and the node memory; and a storage interface component to facilitate data communications between the programmable logic component and the data store; wherein the programmable logic component

executes a set of instructions to reformat a block of data comprising original records, each original record comprising a plurality of fields having different field types, such that a reformatted block of data comprises new records, each new record comprising a plurality of fields of a single field type from the original records; and wherein the programmable logic component causes the new records to be stored as the datasets in the data store.

[0013] In accordance with another aspect of the disclosed subject matter, a data processing system operative in cooperation with a host compute system may be summarized as generally comprising: a management node comprising a host interface and a node interface, wherein the management node is communicatively coupled to the host compute system via the host interface; a memory-supported compute node comprising: (i) a communications link, wherein the memory-supported compute node is communicatively coupled to the node interface at the management node via the communications link; (ii) a data store comprising records associated with a data processing operation; (iii) a programmable logic component to execute the data processing operation in cooperation with the data store; and (iv) a node memory comprising data and instructions to support operation of the programmable logic component in connection with the data processing operation and to facilitate data communications via the communications link; and a pipeline of one or more additional compute nodes, each of the respective one or more additional compute nodes serially connected to the memory-supported compute node via a respective communications link and comprising a respective additional node memory and a respective programmable logic component to execute a respective data processing operation in cooperation with data received via the respective communications link.

[0014] In another aspect, a data processing system operative in cooperation with a host compute system may generally comprise: a management node comprising a host interface and a node interface, wherein the management node is communicatively coupled to the host compute system via the host interface; a memory-supported compute node comprising: (i) a communications link, wherein the memory-supported compute node is communicatively coupled to the node interface at the management node via the communications link; (ii) a data store comprising records associated with a data processing operation; (iii) a programmable logic component to execute the data processing operation in cooperation with the data store; and (iv) a node memory comprising data and instructions to support operation of the programmable logic component in connection with the data processing operation and to facilitate data communications via the communications link; and a pipeline of one or more additional compute nodes, each of the

respective one or more additional compute nodes comprising: a respective communications link, wherein the respective additional compute node is communicatively coupled to an upstream compute node and a downstream compute node via the communications link; and a respective additional node memory and a respective programmable logic component, the respective additional node memory comprising data and instructions to support operation of the respective programmable logic component in connection with a respective data processing operation and to facilitate data communications via the respective communications link; wherein each respective additional compute node executes the respective data processing operation using results from its respective upstream compute node, and passes respective results to its respective downstream compute node; in some implementations, it may be desirable that each respective additional compute node operates without storing the respective results in the respective additional node memory.

[0015] In accordance with yet another aspect of the disclosed subject matter, a method of executing an algorithm in a distributed processing compute environment may generally comprise: providing an initiating compute node comprising a memory store and a first node memory; serially coupling one or more additional compute nodes to the initiating compute node to establish an execution pipeline comprising the initiating compute node and the one or more additional compute nodes, each respective one of the one or more additional compute nodes comprising a respective second node memory; initiating the execution pipeline by loading a bit stream comprising an instruction set related to execution of the algorithm into a respective programmable logic space of each respective compute node in the execution pipeline; executing a first operation associated with the algorithm at the initiating node using information in the bit stream and data retrieved from the memory store, and passing results of the first operation to a first additional compute node in the execution pipeline; wherein the first additional compute node and each subsequent additional compute node in the execution pipeline executes a respective additional operation associated with the algorithm using information in the bit stream and results of a respective preceding operation, and passes respective additional results of the respective additional operation to a next additional compute node in the execution pipeline; and terminating execution of the algorithm at a management node that receives the additional results from a terminal additional compute node in the execution pipeline, the terminating comprising executing a final operation to produce a final result of the algorithm. As with the previous example, it may be desirable in some circumstances

that each respective additional compute node operates without storing the respective results in the respective additional node memory.

[0016] The foregoing and other aspects of various disclosed embodiments will be apparent through examination of the following detailed description thereof in conjunction with the accompanying drawing figures, in which like reference numerals are used to represent like components throughout, unless otherwise noted.

DESCRIPTION OF THE DRAWING FIGURES

[0017] FIG. 1 is a functional block diagram illustrating one implementation of a data processing and analytics system for use in connection with a host compute system;

[0018] FIGS. 2A through 2C are functional block diagrams illustrating one example of a hardware environment accommodating components of a data processing and analytics system;

[0019] FIGS. 3A through 3C are functional block diagrams illustrating certain hardware components of an implementation of a data processing and analytics system such as that depicted in FIG. 1;

[0020] FIG. 4 is a functional block diagram illustrating components of an implementation of a compute node having utility in connection with a data processing and analytics system such as that depicted in FIG. 1;

[0021] FIG. 5 is a diagram illustrating one example of a data routing strategy to route data through portions of the compute node depicted in FIG. 4;

[0022] FIG. 6 is a diagram illustrating another example of a data routing strategy to route data through portions of the compute node depicted in FIG. 4;

[0023] FIG. 7 is a diagram illustrating another example of a data routing strategy to route data through portions of the compute node depicted in FIG. 4;

[0024] FIG. 8 is a diagram illustrating one implementation of a memory architecture having utility in connection with a compute node such as that depicted in FIG. 4;

[0025] FIG. 9 is a diagram illustrating one implementation of a portion of the memory architecture depicted in FIG. 8;

[0026] FIG. 10 is a diagram illustrating a conventional data record structure;

[0027] FIG. 11 is a diagram illustrating one implementation of a data record structure having utility in connection with a compute node operative in accordance with the present disclosure;

[0028] FIG. 12 is a diagram illustrating another implementation of a data record structure having utility in connection with a compute node operative in accordance with the present disclosure;

[0029] FIGS. 13 and 13A through 13H are diagrams illustrating another implementation of a data record structure having utility in connection with a compute node operative in accordance with the present disclosure;

[0030] FIGS. 14 and 14A through 14G are diagrams illustrating yet another implementation of a data record structure having utility in connection with a compute node operative in accordance with the present disclosure;

[0031] FIG. 15 is a diagram illustrating one strategy for data communications between a host compute system and a data processing and analytics system as depicted in FIG. 1;

[0032] FIG. 16 is a functional flow diagram illustrating aspects of one implementation of a method of reformatting a data structure; and

[0033] FIG. 17 is a functional flow diagram illustrating aspects of one implementation of a method of executing data processing operations.

[0034] FIGS. 18 and 19 are functional block diagrams illustrating components of various implementations of a compute node having utility in connection with a data processing and analytics system;

[0035] FIG. 20 is a functional block diagram illustrating components of an implementation of a management node having utility in connection with a data processing and analytics system;

[0036] FIG. 21 is a functional block diagram illustrating one implementation of an algorithm processing system using a management node; and

[0037] FIG. 22 is a functional flow diagram illustrating aspects of one implementation of a method of executing data processing operations.

DETAILED DESCRIPTION

[0038] Certain aspects and features of the disclosed subject matter may be further understood with reference to the following description and the appended drawing figures. In

operation, a system and method employing a distributed hardware architecture, and optionally, an attendant data structure, may have utility in connection with various data processing strategies and data analytics implementations. Specifically, the present disclosure provides for an integrated data processing platform that may generally comprise hardware and software.

[0039] As set forth in more detail below, the present disclosure addresses design and implementation of an architectural framework that may employ high performance data processing elements (compute nodes) coupled with massive data storage (a data store) in a unique configuration that minimizes or eliminates typical processor/network wait states, optimizes instruction fetch memory cycles, and analyzes data that are streamed from an attached or associated data store at the maximum rate at which the data can be accessed or streamed by the storage subsystem. The architectural framework generally contemplates one or multiple compute nodes operating in parallel (and in series, in some cases, as described below), each of which may be configured as a pipeline of computational elements that can be configured and operative to implement a variety of algorithms or other data processing operations. In the configurations provided as examples, a fully parallelized dataflow environment, which exhibits extremely high performance for analyzing massive amounts of data, is described.

[0040] The disclosed architectural arrangement may generally include a set of programmable logic blocks that may be linked in various ways to create (as well as to execute) arbitrary algorithms. This strategy greatly reduces the learning curve for using programmable processors to implement computational algorithms. For example, a typical application programmer (*e.g.*, with a limited understanding of programmable logic hardware, *per se*) may nevertheless easily use the disclosed architecture without needing to understand the intricacies of using one of more programmable logic block hardware components and how their structures or interconnections might affect or otherwise relate to algorithm requirements of the application program.

[0041] Additionally, embodiments are disclosed of a system and method of morphing (*i.e.*, reorganizing or reformatting) data located in data records; as is generally known in the art, these data are typically stored or maintained in files or datasets (or “records” or “pages”). The disclosed system and method may be implemented for the purpose of increasing the performance of analytic algorithms being applied to the data, and may optimize or maximize the rate at which data may be correctly presented to an analytic algorithm.

[0042] It will be appreciated that the present disclosure addresses design and implementation of memory controller (such as a Flash controller) hardware and strategies capable of supporting optimized, overlapping data read and write requests from or to a memory component, while also allowing streaming of data from the same memory component for analysis by an algorithm or other data processing operation. In one implementation, scheduling of read access requests issued to memory component targets may be sequenced and overlapped in a strategic manner such that data output from the memory component is maximized, optimized, or otherwise increased as compared to conventional memory controller technologies; specifically, the disclosed systems and methods may decrease or eliminate wait times typically caused by Flash (or other memory) read delays or latency.

[0043] In accordance with the disclosed subject matter, it will be appreciated that a power management system may readily be implemented using the architecture set forth below (see, *e.g.*, FIG. 4) to implement a system of microcontroller- or microprocessor-based regulators that communicate with each other using an on-board local area network. This solution may allow power regulators to be sequenced as desired or required (for instance, by power consuming subsystems on a host hardware module), and also may allow the regulators to balance current load requirements. In some instances, this implementation may also provide a failover capability in the event that any single regulator or group of regulators fails. The result is a safe power environment that correctly starts up a module (such as a compute node or an entire interface card or other hardware component comprising or accommodating a plurality of compute nodes such as described below) without risk of damage due to uncoordinated power sequencing and also provides for a balanced, fault-tolerant running environment to facilitate or to improve reliability with respect to a particular hardware module operative in accordance with the disclosed subject matter. For instance, core voltage supplied to a processing unit in a router component as described below may be supplied via, or in cooperation with, a poly-phase, shared (*i.e.*, redundant) regulator to handle power sequencing, load balancing, and fault-tolerance; it will be appreciated that other voltage supply subsystems necessary or desirable to provide the following functionality may benefit from a similar or analogous voltage regulation strategy.

[0044] Turning now to the drawing figures, FIG. 1 is a functional block diagram illustrating one implementation of a data processing and analytics system for use in connection with a host compute system. As illustrated, one implementation of system 100 may generally comprise a host compute system 199, a router module 120, and a compute node 140.

[0045] Host compute system 199 may generally be embodied in or comprise a computer server, a desktop or workstation computer, a laptop or portable computer or tablet, or a combination of one or more of such components. In operation, host compute system 199 may be employed to initiate, instantiate, or otherwise to request data processing operations to be executed by system 100 substantially as set forth below. In that regard, host compute system 199 may include one or more microprocessors, field programmable gate arrays (FPGAs), microcontrollers, or other digital processing apparatus, along with attendant memory, controllers, firmware, network interface hardware, and the like. For example, host compute system 199 may generally comprise multiprocessor systems, microprocessor-based or programmable consumer electronics, personal computers ("PCs"), networked PCs, minicomputers, mainframe computers, and similar or comparable apparatus for general purpose or application-specific data processing. Various implementations of host compute system 199 may be deployed in distributed computing environments in accordance with which tasks or program modules may be performed or executed by remote processing devices, which may be linked through a communications network. Those of skill in the art will appreciate that any of various computer servers, work stations, or other processing hardware components or systems of components may be suitable for implementation at host compute system 199, and that the disclosed subject matter is not limited to any particular hardware implementation or system architecture employed at host compute system 199.

[0046] With respect to requesting, initiating, enabling, facilitating, and receiving results of the data processing functionality described herein, host compute system 199 may be communicatively coupled to router module 120, enabling bi-directional data transmission in accordance with any of various communications interfaces or telecommunications protocols generally known in the art or developed and operative in accordance with known principles. Router module 120, in turn, may also be communicatively coupled to compute node 140 such that data may be transmitted or transferred, via router module 120, between host compute system 199 and compute node 140 substantially as set forth below.

[0047] Specifically, router module 120 may generally comprise a router 122, one or more router memory components 124a, 124b, 124c, … 124n ("router memory 124") and a host interface 126; additionally, router module 120 may also include one or more node interfaces (not illustrated in FIG. 1) such as network interface cards, wireless or inductive hardware components, or communications ports, jacks, or plugs, for instance, to enable router 122 to communicate with a respective compute node 140 as is generally known in the art. In use, router 122 may be operable

to transmit data packets between host compute system 199 and a respective compute node 140 in cooperation with routing information (destination address, packet header data, flood domains, *etc.*) and other short term data (look up tables, resource availability, performance or load balancing data, *etc.*) stored in or accessible by router memory 124. In some instances, router 122 may distribute data packets and write commands to compute nodes 140 in accordance with a data-aware round robin strategy. Router memory 124 may be embodied in or comprise, by way of example, volatile memory such as random access memory (RAM) in any of its various forms, for instance, static RAM (SRAM), dynamic RAM (DRAM), double-data rate (DDR) RAM, and the like; in some applications, DDR4 RAM may be used as router memory 124. Attendant bus structures and memory controller elements are omitted from FIG. 1 for clarity, but are well-known in the art. In one implementation, router 122 may execute all routing functionality on or in cooperation with a Kintex™ KU095 FPGA platform marketed by Xilinx, Inc., but as many types of hardware implementations are generally available, the structure and operation of router 122 and router memory 124 are not described further.

[0048] In some implementations, host interface 126 may be operative in accordance with Peripheral Component Interconnect (PCI) or PCI Express (PCIe) hardware and associated communications protocols. Alternatively, other types of communications interface hardware and methodologies may be employed, such as Ethernet™ or Serial ATA (SATA). The specific hardware and communication strategy embodied by or incorporated into host interface 126 may be selected in accordance with bandwidth requirements, cooperating hardware at host compute system 199, functionality and operational characteristics of router 122, or a combination of these and a variety of other factors. The same is generally true of the hardware and communication protocol implemented at node interfaces, which may be configured and operative to communicate data packets bi-directionally between router 122 and compute nodes 140 as set forth below.

[0049] In that regard, router 122 may be communicatively coupled (via a respective node interface) to a respective compute node 140 via a communications link 149 integrated into each respective compute node 140. In the FIG. 1 illustration, only a single compute node 140 is illustrated, but an additional three in the first compute node tier are also depicted abstractly by the bi-directional arrows from router 122. It will be appreciated that each respective series of compute nodes 140 may employ a dedicated communications channel or “lane” for data transmission with router 122, such that the FIG. 1 architecture contemplates four parallel lanes, one for each of a series of compute nodes 140. In that regard, compute node 140 may be communicatively

coupled, via communications link 149, to one or more additional compute nodes 140 connected in series (*i.e.*, at different successive tiers), as illustrated at the top right side of FIG. 1.

[0050] In addition to communications link 149, in the FIG. 1 arrangement, each compute node 140 may generally comprise a compute array 142, a data store 143, a data store interface 145, node memory 144, one or more datamover apparatus (DMA) 146a and 146b ("DMAs 146"), and a processing system 148. These elements are described in detail below with specific reference to FIG. 4.

[0051] In some implementations, data store 143 may be embodied in or comprise a mass data storage component, such as a non-volatile data storage device, one example of which is an Electronically Erasable Programmable Read Only Memory (EEPROM) store. For example, data store 143 may be, or include, Flash memory, though other memory types having suitable or appropriate characteristics to facilitate the functionality set forth herein may be in use currently or developed in the future. Specifically, though all Flash is EEPROM (as a technical matter), it is noted that not all EEPROM is Flash. Typically, those of skill in the art may tend to interpret the term EEPROM as a very specific type of non-volatile memory, but the term is not so limited as employed in the context of this disclosure; examples using Flash are described below, but any non-volatile storage media and high-bandwidth interface protocol may have utility in the context of the disclosed subject matter. In situations in which data store 143 is implemented as Flash, data store interface 145 may employ an Open NAND Flash Interface (ONFI) protocol; in the case that data store 143 uses a different memory technology, data store interface 145 may be selected or modified accordingly. In operation, data store interface 145 manages access (*e.g.*, read and write access) to data store 143, as is generally known in the art.

[0052] As with router memory 124, node memory 144 may be implemented as or comprise DDR4 RAM, or other volatile or non-volatile memory suitable to support operations of compute node 140, in general, and processing system 148, in particular. In the illustrated architecture, processing system 148 may generally handle read/write commands and other memory access with respect to data store 143 (via data store interface 145), receive and act upon interrupt requests (*e.g.*, with respect to operations executing at compute array 142), and regulate, control, or otherwise influence data transfers managed by DMAs 146. For instance, in one embodiment, processing system 148 may comprise a set of one or more processing components to manage all interrupt handling as well as Advanced Extensible Interface (AXI) or other data transfers via DMAs 146, on the one hand, and a different set of one or more processing components to manage

general system administration (or "house-keeping") and management tasks for compute node 140 as well as algorithm execution support for compute array 142, on the other hand. While this implementation is described in more detail below with reference to FIG. 4, it is worth noting here that it is only one example, and that many different component arrangements and task distributions are within the scope and contemplation of the disclosed subject matter.

[0053] It is noted that router module 120 (including host interface 126) and the various compute nodes 140 described above, as well as associated busses, communications pathways, and attendant or supporting hardware infrastructure, may be incorporated into or disposed on a carrier board (such as a printed circuit board (PCB)) or interface card ("card," see reference numeral 200 in FIGS. 2A and 2B) that is configured and operative to cooperate with structural components of host compute system 199 as is generally known. Specifically, one or more cards 200 may be inserted into or otherwise engage receptacles, slots, jacks, plugs, *etc.* that are electrically and communicatively coupled to a motherboard, bus, or other interface component at host compute system 199; accordingly, system 100 may be scalable as a function of slots that are available at host compute system 199 to receive a card 200.

[0054] At a high level, the disclosed architecture of system 100 generally includes a star arrangement in accordance with which a central host interface 126 (*e.g.*, a PCIe interface) to/from host compute system 199 may be routed to four (or more) "riser" positions via router 122 (*e.g.*, an FPGA or microprocessor platform). At each riser position, a stack (*i.e.*, various tiers in FIG. 1) of compute nodes 140 may be communicatively coupled in series. One configuration may generally comprise two rows of four compute nodes 140 (tiers), whereas the FIG. 1 configuration includes four rows, each having an indeterminate number of tiers. In accordance with a PCIe implementation, each riser position or stack may employ eight discrete lanes of 8 Gb/s data links, *i.e.*, a total of 64 Gb/s, full duplex, capacity per riser position.

[0055] As noted above, communications link 149 may be employed to couple compute nodes 140 within a stack (*i.e.*, at different successive tiers) in series. In some implementations, communications link 149 may be based upon and operative in accordance with standard serial interfaces or protocols, though some proprietary serial links may also be appropriate, depending upon the functionality and operational characteristics of compute array 142, for instance. For instance, one example of compute array 142 may employ or comprise a Xilinx Zynq™ Ultrascale+™ ZU9EG FPGA, in which case, it may be desirable that communications link 149 employ or comprise a Xilinx Aurora™ communication link, as it functions well with transceivers

provided by the same manufacturer. In operation, communications link 149 may be selected, configured, and operative to provide a light-weight wrapper of the complex multi-gigabit transceivers deployed in the FPGA fabric of compute array 142, and so it may be desirable to select communications link 149 hardware as a function of, or otherwise taking into consideration, the hardware and operation of compute array 142. Those of skill in the art will appreciate that many options exist for both compute array 142 and communications link 149, and that specific commercial embodiments have been identified by way of example only, and not by way of limitation.

[0056] In connection with Big Data and other resource-intensive processing applications, it may generally be desirable to allow compute nodes 140 autonomously to handle much of the data ingest activities and distribution of processing responsibilities; accordingly, compute nodes 140 themselves may, in some cases, determine an appropriate method of round robin processing and tracking of write histories during complex data processing operations. Additionally or alternatively, it may be desirable that host compute system 199 provide directions or instructions regarding round robin processing and write operations.

[0057] For example, each compute node 140 may monitor and track, or log, its own local activity; as the level of detail or the quantity of activity to be tracked increases, however, the amount of node memory 144 required (to enable rapid access, for example) may generally increase, the latency for a given memory access operation (or for a given type or class of operations) will increase, or both. This is especially true if information or local activity sought to be tracked must be extracted from data store 143 prior to each new write transaction.

[0058] Further, communication with neighboring compute nodes 140 in the disclosed star architecture may generally become increasingly costly the further away a particular compute node 140 is from the neighboring compute node 140. Elapsed time, throughput degradation (if control or administrative paths are shared with payload data), programmable logic resource consumption, and energy consumption may all increase in proportion with or as a function of distance between respective compute nodes 140 that are to exchange information. One way to eliminate or to minimize such deleterious effects is to implement a mesh architecture (not shown), in accordance with which an “any to any” communications strategy may be integrated into the architecture of system 100. In such an embodiment, a bit of sideband control traffic may generally have minimal or negligible impact, and may be implemented effectively without having a materially adverse effect on payload data or overall performance of system 100.

[0059] One approach to round robin processing that may readily be implemented across multiple cards 200 is to allow host compute system 199 itself to direct write commands to specific data targets or memory addresses, down to the data store 143 level, for instance. In particular, host compute system 199 may generally be apprised of all the data (*i.e.*, system-wide) to be written, and is unique in that regard (as compared to other components of system 100); accordingly, host compute system 199 may be enabled to select, compute, or otherwise to determine a proper, appropriate, or optimal granularity of distribution (*e.g.*, after how many clock cycles may the round robin advance from one compute node 140 to another). Furthermore, host compute system 199 may keep track of last writes across the entirety of the topography of system 100, and then re-start the round robin at an appropriate point (*e.g.*, after a power cycle, crash, or other interruption).

[0060] Employing host compute system 199 in the foregoing manner may not scale well in terms of performance under certain circumstances, particularly in situations where host compute system 199 is limited to local resources and not distributed across multiple processing platforms. For applications or environments requiring high scalability, an alternative approach may include enabling host compute system 199 to inform each attached or coupled card 200 of an optimal granularity of the round robin method that it is to utilize (for example, to attain maximum or some desired level of efficiency). In the case of a system employing multiple cards 200, host compute system 199 itself may implement the first tier of round robin distribution. In this example, each compute node 140 may keep track of the last channel as well as the memory address or data record (per channel) that was used for each record or column of data that is written as set forth in more detail below.

[0061] If the first stage of the round robin mechanism (*e.g.*, on the FPGA implemented at router 122) has no non-volatile storage available, it may not be possible to store the last target of distribution across power cycles. To avoid overburdening the first compute node 140 as a target in the case of many power cycles (each of which may require round robin re-initialization), the round robin logic may be initialized with a random first target in some applications. If the round robin granularity is too small and the data are distributed across multiple compute nodes 140 without filling a full page on any of them, the performance of system 100 incurs partial page write penalties on all compute nodes 140 involved.

[0062] FIGS. 2A through 2C are functional block diagrams illustrating one example of a hardware environment accommodating components of a data processing and analytics system.

[0063] FIG. 2A depicts a top view of a carrier board (such as a PCB) or interface card ("card" 200) that may support or carry router module 120 and compute node 140 substantially as set forth above. Card 200 may comprise a typical or conventional form factor host interface 126 suitably dimensioned and configured to create electrical and/or power connections when engaged with a cooperating structure integrated into, associated with, or otherwise communicatively coupled to host compute system 199. As noted above, host interface 126 may be a PCIe, Ethernet, SATA, or other interface, and may be coupled to electrical and power connections on or integrated with card 200 via conventional PCB fabrication techniques. FIG. 2A also illustrates communications links 149a, 149b, 149c, through 149n (links 149), a respective one dedicated to each of a respective number "n" of stacks of compute nodes 140 coupled in series as set forth above. It is noted that card 200 may employ more or fewer than the four communications links 149 that are depicted in FIG. 2A.

[0064] FIG. 2B depicts a top view of a card 200 on which is disposed one example of a managing interface 202 component, and FIG. 2C depicts a top view of the managing interface 202 component of FIG. 2B. In some implementations, managing interface 202 may be embodied in or comprise router module 120 described above; in that regard, managing interface 202 may be an example of one embodiment of router module 120, such that router 122 maps to an FPGA 299 and router memory 124 components map to interface memory components 204a and 204b, each of which is depicted as having a number of discrete memory cells or memory modules 214, such as dual in-line memory modules (DIMMs) or other independently addressable memory structures. It is noted that, in this example, managing interface 202 may further comprise additional interface memory components (not shown) to correspond to the number of router memory 124 components employed in router module 120. As an alternative, router module 120 may be deployed in the FPGA 299 block in FIG. 2, and interface memory components 204a and 204b may support the functionality of router module 120 substantially as described above. Those of skill in the art will appreciate that other alternatives exist, and that the disclosed subject matter is not intended to be limited to the architecture or structural arrangement illustrated in FIGS. 2B and 2C.

[0065] FIGS. 3A through 3C are functional block diagrams illustrating certain hardware components of an implementation of a data processing and analytics system such as that depicted in FIG. 1. The left side of FIG. 3A depicts communications links 149a, 149b, 149c, through 149n (links 149) and a power coupling 399 in isolation, while the right side of FIG. 3A

depicts links 149 and power coupling 399 as they may be deployed on card 200. As noted above, each respective communications link 149 may be dedicated to each of a respective number "n" of stacks of compute nodes 140 coupled in series, and card 200 may employ more or fewer than the four communications links 149 that are depicted in FIG. 3A.

[0066] FIG. 3B is a first view of one side of a card or substrate upon which elements of compute nodes 340a, 340b, 340c, through 340n (compute nodes 340) may be disposed, and FIG. 3C is a second view of an opposite side of a card or substrate upon which other elements of compute nodes 340 may be disposed. The left side of each of FIGS. 3B and 3C illustrates a single compute node 340n in isolation, while the right side illustrates a number, "n," of such compute nodes implemented on card 200.

[0067] It is noted that compute nodes 340 may be integrated with, or disposed upon, a PCB or other carrier board, interface card, or similar structural substrate such as described above with reference to card 200. In the example of FIGS. 3B and 3C, for instance, some elements or components of compute nodes 340 may be disposed upon, attached to, or integrated with one side of such a card, whereas other components of compute nodes 340 may be disposed upon, attached to, or integrated with an opposing side. This particular structural arrangement, and appropriate manufacturing methods for effectuating same, are generally known in the art of electronics fabrication, and the disclosed subject matter is not intended to be limited to any particular fabrication techniques or process node methodologies.

[0068] Each compute node 340 may generally comprise communications link 149, a data store (depicted abstractly at reference numeral 343 in FIG. 3B), a compute array 342, and node memory 344.In some implementations, compute nodes 340 may map generally to compute nodes 140 illustrated in, and described with reference to, FIG. 1. In one example, data store 343, compute array 342, and node memory 344 may correspond, respectively, to data store 143, compute array 142, and node memory 144 described above; in such an example, compute nodes 340 may additionally comprise components that are equivalent or analogous to data store interface 145, DMAs 146, and processing system 148, though these are omitted from FIGS. 3B and 3C for clarity. In another embodiment using a different mapping, compute array 342 itself may incorporate some or all of the components illustrated and described with reference to compute node 140, in which case data store 343 and node memory 344 may be employed to support the functionality of compute array 342.

[0069] In the examples described above, system 100 may facilitate scalability, in terms of both memory or storage density as well as compute power, that is greater than conventional systems. Additionally, the foregoing modular approach may accommodate design longevity and flexibility with respect to module design and implementation (or "bring up"). In particular, data store 143/343 may be closely coupled to compute resources (such as compute array 142/342), while remaining physically separated, such as by deployment on opposing sides of an interface card 200 or PCB, which facilitates cooling and minimizes thermal effects that are typically associated with implementing mass storage in proximity to high power processing resources or system on chip (SoC) hardware.

[0070] FIG. 4 is a functional block diagram illustrating components of an implementation of a compute node having utility in connection with a data processing and analytics system such as that depicted in FIG. 1. The FIG. 4 example represents one implementation of a compute node 140n such as illustrated in FIG. 1, and generally comprises communications link 149n, a programmable logic block 410, a processing system 430, and node memory 450.

[0071] In the FIG. 4 example, compute node 140n (in tier "n" of a particular stack of compute nodes 140) may be communicatively coupled, via communications link 149n to one or more additional compute nodes 140 connected in series at different successive tiers, *i.e.*, to compute node 140n - 1 and compute node 140n + 1. As described above, communications link 149n may be based upon or operative in accordance with standard serial interfaces or protocols, or may employ proprietary serial link technologies, for example, as a function of or influenced by the functionality and operational characteristics of compute array 142.

[0072] In the FIG. 4 implementation, programmable logic block 410 is illustrated as generally comprising, in addition to compute array 142 and DMA 146a, a storage interface component 411n. As noted above, compute array 142 may employ or comprise any of various commercial embodiments of an FPGA, and many alternatives may be equally suitable or adequate, depending upon design considerations such as power consumption, thermal factors, throughput and processing capabilities, overall requirements of system 100, price constraints, or a combination of these and other boundaries conditions that are typical in electronics design. DMA 146a may be employed to handle data traffic between processing system 430 or node memory 450, on the one hand, and communications link 149n, on the other hand, as illustrated in FIG. 4. In particular, DMA 146a may be embodied in or comprise a data transfer apparatus, device controller, or other hardware or firmware module that is configured and operative to facilitate bi-directional data

communications between electronic devices, either independently or in cooperation with control signals or other instructions received from processing system 430 or specific components thereof. Any of various commercially available hardware- or firmware- supported datamover implementations may be suitable for DMA 146a, provided that a particular implementation enables data transfer operations and throughput bandwidths sufficient to support required or desired functionality of compute node 140n, in particular, and system 100, in general.

[0073] Storage interface component 411n may be one of a number, "n," of such components employed in connection with a particular programmable logic block 410, as represented by the stack on the right side of FIG. 4. While a number of similar or identical such storage interface components may be integrated with or implemented in cooperation with programmable logic block 410, only one storage interface component 411n is described; it is noted that every such storage interface component in the illustrated stack need not be identical, and that various alternatives to the specific storage interface component 411n may be suitable for a particular implementation of programmable logic 410. The functionality and specific architecture of storage interface component 411n may vary as a function of data store 143 requirements, the operational characteristics and processing capabilities of compute array 142, or a combination of these and other factors. As noted above, a particular data store 143 having utility in connection with system 100 may be embodied in or comprise a mass data storage component, such as an EEPROM store. In the FIG. 4 embodiment, data store 143 may be a Flash memory, and accordingly, data store interface 145n may employ an ONFI protocol. In operation, data store interface 145n may manage access (*e.g.*, read and write access) to data store 143, as is generally known in the art. Other types of controller technology may be substituted as a function of the particular technology employed at data store 143, for example.

[0074] Storage interface component 411n may also comprise DMA 146b. As with DMA 146a, those of skill in the art will appreciate that any of various commercially available hardware- or firmware- supported datamover implementations (such as a data transfer apparatus, device controller, or other hardware or firmware module) may be suitable for use at or in conjunction with DMA 146b. In operation, DMA 146b may facilitate bi-directional data communications between compute array 142 or processing system 430 (and node memory 450), on the one hand, and storage interface component 411n, on the other hand. DMA 146b may function independently or autonomously, for example, or in cooperation with control signals or other instructions received from processing system 430 or specific components thereof, to effectuate data transfer operations

sufficient to support operation of compute array 142 with respect to read/write or other access to data store 143.

[0075] Storage interface component 411n may also comprise a routing block 413n. Routing block 413n may be implemented as a first in / first out (FIFO) buffer or similar memory structure temporarily to store one or both of data transferred (via data store interface 145n) between compute array 142 and data store 143, and data transferred (via DMA 146b) between compute array 142 and node memory 450. Various commercial implementations of a FIFO buffer structure are available (or may be developed and operative in accordance with known principles) to provide the functionality of routing block 413n. In some implementations, routing block 413n may comprise hardware logic, or may be operative in accordance with software instruction sets, sufficient to allow routing block 413n to access or to target specific memory locations in data store 143, either independently or in cooperation with compute array 142, processing system 430, data store interface 145n, or a combination of these and other components. The disclosed subject matter is not intended to be limited by the nature, structural configuration, or operational characteristics of routing block 413n, as many alternatives may be employed as a function of design considerations and operational requirements of system 100, in general, and compute node 140n, in particular.

[0076] Processing system 430 may generally comprise a first set of one or more processors 432a and 432b and a second set of one or more processors 439a, 439b, 439c, through 439n. Bus structures, power sources, controllers, internal memory, and other attendant and supporting hardware and communications paths have been omitted from FIG. 4 for clarity. It is noted that all of the processor blocks (432a, 432b, 439a, 439b, 439c, and 439n) depicted in FIG. 4 may be implemented as or generally comprise a microprocessor, a multi-core microprocessor, a microcontroller, an FPGA, a programmable logic controller (PLC), a programmable logic unit (PLU), or some similar or analogous hardware component suitable for data processing as is generally known in the art. In one implementation, for example, one or both of processors 432a and 432b may be embodied in or comprise a Cortex™ A5™ microprocessor core marketed by ARM Holdings PLC, and some or all of processors 439a, 439b, 439c, through 439n may be embodied in or comprise a Cortex A53™ microprocessor core marketed by ARM. Other configurations (*e.g.*, the number of processors employed) and specific types of processing devices are contemplated, and may readily be implemented, as a function of, or influenced by, design goals, processing overhead, application-specific requirements for compute node 140n,

and a variety of other factors. It is also noted that the example of processing system 430 illustrated in FIG. 4 represents only one implementation of processing system 148 depicted in FIG. 1, and that the FIG. 4 embodiment may comprise more or fewer processing resources than those shown.

[0077] In operation, processors 432a and 432b may manage or otherwise influence operation of DMAs 146a and 146b, respectively. Also as depicted in FIG. 4, each of processors 432a and 432b may employ on chip memory (OCM) to facilitate interoperation with DMAs 146. In that regard, DMAs 146a and 146b may be configured and operative to transfer data between and amongst components of compute node 140n as illustrated in FIG. 4 under control of or responsive to instructions from processors 432a and 432b; some or all of such instructions may be resident in or accessible by OCM components or hardware. Additionally or alternatively, OCM may also host or maintain status updates and attendant data acquired from or associated with operation of mailbox 452n described below. In some instances, OCM may have a lower latency than external memory, such as a DDR RAM component, and by employing OCM to manage mailbox 452n updates, memory 450 bandwidth may be reserved for payload traffic, rather than administrative traffic.

[0078] In some implementations, operation of processors 432a and 432b may be influenced by interrupts received, for example, from DMAs 146a and 146b. For example, one or both of DMAs 146a and 146b may be operative to issue an interrupt following expiration of a predetermined or dynamically selected duration (*i.e.*, a "timeout"), or upon the occurrence of a predetermined or pre-established event (such as completion of a certain threshold number of transactions). In this example, one or both of processors 432a and 432b may be operative to initiate exchange of mailbox 452n status information, or to execute some other administrative or substantive function, upon receipt of such an interrupt. Additionally or alternatively, processors 432a and 432b may be configured and operative to schedule or otherwise to influence operation of processors 439a, 439b, 439c, through 439n, *e.g.*, via interprocessor interrupts or other communications protocols or instructions.

[0079] Specifically, processors 439a, 439b, 439c, through 439n may be responsible for a variety of tasks which, under some circumstances, may be requested or required by processors 432a and 432b, or which may be initiated autonomously or otherwise without direction from external components such as processors 432a and 432b. In various implementations, processors 439a, 439b, 439c, through 439n may perform some or all of the following functions, or a subset

thereof, as requirements of system 100, in general, or node 140n, dictate: data transformation and storage (*e.g.*, with respect to node memory 450, memory store 143, or both); data dispatch for write/read operations (again, with respect to node memory 450, memory store 143, or both); command interpretation and execution (*e.g.*, with respect to data processing and algorithm compilation and execution); algorithm load and compute; and memory exception handling. Those of skill in the art will appreciate that various other functions may be assigned to or attributable to processing system 430 as a function of desired operability of compute node 140, and that any of numerous hardware arrangements, computing architectures, and distributed processing resources may be employed at processing system 430 to enable its functionality in the context of supporting desired capabilities of compute node 140.

[0080] Processing system 430 may cooperate with programmable logic block 410 for data transfer and read/write operations with respect to data store 143. In particular, flexible FIFO or other buffer structures and multiplexers may be employed to provide multiple data paths to and from each memory (such as Flash) channel. Some examples of specific data paths and supporting hardware are described below with reference to FIGS. 5 through 7.

[0081] In the FIG. 4 example, node memory 450 is one representation of the node memory 144 illustrated and described above with reference to FIG. 1. In that regard, node memory 450 may be implemented as or comprise DDR4 RAM, or other volatile or non-volatile memory suitable to support operations of compute node 140n and processing system 430. In some implementations, node memory 450 may generally comprise one or more mailboxes (*i.e.*, resource-specific data storage cells or memory locations) 452n, as well as one or more functional programming (FP) tables 454n, algorithm buffers 456n, and FP buffers 458n. These components may generally be operative to store data and instructions necessary or desirable for processing system 430 to perform write/read/execute operations in support of programmable logic block 410 as is generally known. An example is provided below.

[0082] In order to minimize the need for processor 432a intervention during write transactions from host compute system 199 to compute node 140n, a largely autonomous mailbox 452n and DMA 146a strategy may handle the entire transfer (or a portion thereof) of incoming data packets to DDR4 (or similar or analogous) buffers at compute node 140n, in general, and at node memory 450, in particular. In some implementations, it may be desirable that the inbound buffers are “content agnostic” and are treated generically. In that regard, a plurality (such as one hundred

twenty eight, for instance) of mailboxes 452n may be initialized for this purpose, and their status may be set to "empty" at boot time.

[0083] In programmable logic block 410, inbound DMA 146a may also be initialized and its internal control logic may be set to "full availability." In this instance, when data packets arrive, the first available mailbox 452n address may be retrieved (*e.g.*, from a local list maintained in node memory 450), and the incoming packed may be posted to that particular mailbox 452n. The local list may then be updated to reflect that a particular mailbox 452n is occupied, and that a next mailbox 452n in series, or otherwise selected or determined in accordance with processing system logic, should be used for a next inbound data packet or series of data packets. At that point, two processes may be initiated, as noted briefly above. First, a transaction counter or other meter may be incremented (or decremented, as a matter of design choice); once a message count threshold is reached, attaining the threshold may trigger an update of the occupied mailbox list by processing system 430. Second, a timer may be set; once a predetermined period of time has elapsed (*i.e.*, a timeout event), expiration of the time period may trigger an update, just as if the message count had been reached. This second process may ensure that single or last messages will be captured, even in the event that a message or transaction counter does not reach a predetermined or minimum threshold to trigger an update in accordance with the first process. In some implementations, during such an update, programmable logic block 410 may also receive an updated list (*e.g.*, a message or instruction representative of emptied mailboxes 452n since a last transaction, or some other relevant data) as prepared by processing system 430.

[0084] With the foregoing overview of system 100 and compute node 140n architecture in mind, the following addresses one example of algorithm load and execute initiation and functionality, by way of example only, and not by way of limitation. In one implementation, initialization of each compute node 140n may begin by booting from a local golden bit stream; this initialization may generally provide a common (*i.e.*, across all compute nodes 140) infrastructure in each respective programmable logic block 410. For example, initialization may define data paths (such as those described below with reference to FIGS. 5 through 7), protocols for communications link 149n, identification and initialization of various I/O controllers, and the like – generally establishing and defining the operating environment for each compute node 140n ("compute node environment"). In one embodiment, the initial bit stream may include an algorithm block or other information concerning particular algorithm execution, though this may occur after

the initial bit stream, if necessary or desired. In some circumstances, initialization may also provide boot code or other initial parameters for the various processors in processing system 430.

[0085] Upon initialization of compute node 140n, new or updated algorithm bit streams may then be transmitted, *e.g.*, by host compute system 199, via router module 120 and communications link 149n. In some implementations, algorithm bit streams may contain both algorithm logic and the same compute node environment used in the local golden bit stream employed during initialization. When an algorithm bit stream is received from (*e.g.*, from host compute system 199), some or all of the following processes may ensue: the algorithm bit stream may be provided to all compute nodes 140 simultaneously, or substantially simultaneously or concomitantly; each respective compute node 140n may complete any outstanding data transactions; normal compute node 140n operation may be briefly paused while the new bit stream is loaded into all compute arrays 142; normal compute node 140n operations may be resumed and all compute nodes 140 may begin executing the algorithm (and continue to do so until it completes or times out); and upon completion, computational results may be returned along with status, and the algorithm bit stream may be left in place for potential re-use at a later time.

[0086] Regarding some of the data paths supporting the foregoing functionality, it is noted that FIGS. 5 through 7 are diagrams illustrating examples of data routing strategies to route data through portions of the compute node depicted in FIG. 4.In the illustrated example, each storage interface component 411n may generally comprise a group of FIFO buffers or similar structures (such as represented by routing block 413n) connected with a versatile routing fabric. During operation, the routing fabric allows for run time configuration of data flow paths through routing block 413n and (1) to/from node memory 450 or OCM buffers associated with components of processing system 430, (2) to/from data store interface 145n, (3) to/from programmable logic block 410 compute pipelines, (4) to/from programmable logic block 410 path to communications link 149n.

[0087] In some embodiments, all such data paths may be implemented with 16-bit width and follow the AXI stream protocol. In that regard, an AXI stream is generally an address-less transport mechanism that requires a minimal amount of control signals (*e.g.*, only READY, VALID and LAST). Using this example approach, data flow may begin as soon as an intended target of a data transaction indicates “READY” and a source of the data transaction indicates “VALID.” Conversely, data flow may be suspended when one of the foregoing two conditions is no longer

true. The "LAST" indicator may be used by logic in the target to detect packet or data stream boundaries, for example.

[0088] In some implementations of interface routing fabric employed at routing block 413n, one or more 4:1 multiplexers may have particular utility. For instance, a 4:1 multiplexer may cooperate well with a 6:1 (four data signals, two control signals) logical unit (LUN) data structure (described below), and therefore may maximize use of the basic elements found in some viable options for FPGA structures of compute array 142. Other types of multiplexers may be employed, for instance, to match a fabric used in a commercial implementation of compute array 142, to combine certain functionality set forth below into fewer components, or as a result of a variety of other considerations. A 4:1 multiplexer also provides a reasonable multiplexing strategy, to the extent that it can devote one input section to each of the four sources noted above, which may generally cover all use cases in most practical applications and hardware configurations.

[0089] In addition to the 4:1 input section, each multiplexer output may be configured and operative to drive multiple targets simultaneously as each path may be implemented as 4:1 source select. So, in such an example, one implementation of a routing paradigm may focus on the source of the data, rather than the component to which the data are to be transmitted. This paradigm allows each source to provide data to multiple targets simultaneously (or substantially simultaneously), and it makes an illegal or conflicting configuration (*e.g.*, two or more sources trying to drive one target simultaneously) impossible.

[0090] In some implementations, some AXI stream flow control signals may be treated with some extra logic applicable to the architecture and methodologies described herein. For instance, some logic rules for AXI stream flow control may be expanded as follows: READY signals of data sinks that are connected to a common source may be ANDed (*i.e.*, combined); READY signals for destinations that not selected (*i.e.*, inactive) may be set to "1" by default; and VALID status may only be presented by a data source when it sees the ANDed READY from all possible destinations (data sinks) to prevent a single sink from initiating the dataflow. This mechanism is very flexible and resource efficient. It may generally enable a versatile interconnect with respect to the compute pipeline, in accordance with which one source value may need to feed multiple compute engines in parallel (*e.g.*, count a value "x," square "x," and multiply "x" by a different value "y"). By instantiating multiple concurrent "listening ports" (*i.e.*, compute elements) to monitor one data source, it is possible to implement an arbitrary number of concurrent calculations in

parallel, subject to the architectural and processing limitations of the hardware embodying or employed at compute array 142, for example.

[0091] FIGS. 5 through 7 show such an arrangement with two parallel FIFOs or other buffer structures, each of which is fed by an input multiplexer 511 coupled to all source interconnects, and selectively outputs to any of three paths to programmable logic block 410 for compute. The actual count of paths to programmable logic block 410 compute resources may depend, for example, upon the loaded algorithm, the corresponding compute pipeline, and a variety of other factors.

[0092] In particular, each of FIGS. 5 through 7 show four inputs to input multiplexers 511: from data store 143 (*e.g.*, "Flash Channel" via data store interface 145n); from node memory 450 (*e.g.*, via DMA 146a); a first input from the compute pipeline (*i.e.*, programmable logic block 410 compute array 142); and a second input from the compute pipeline (*i.e.*, programmable logic block 410 path to communications link 149n). These inputs are concatenated at input multiplexers 511; output from the respective FIFOs may be concatenated at output multiplexers 512.

[0093] FIG. 5 illustrates one data path from data store 143 (*e.g.*, "Flash Channel" via data store interface 145n) to node memory 450 (*e.g.*, via DMA 146a), and a separate data path from node memory 450 (*e.g.*, via DMA 146a) to data store 143 (*e.g.*, "Flash Channel" via data store interface 145n). FIG. 6 illustrates one data path from data store 143 (*e.g.*, "Flash Channel" via data store interface 145n) to one of the compute pipelines, and a separate data path from node memory 450 (*e.g.*, via DMA 146a) to a different one of the compute pipelines. FIG. 7 illustrates one data path from data store 143 (*e.g.*, "Flash Channel" via data store interface 145n) to one of the compute pipelines, and a separate data path from one of the compute pipelines to node memory 450 (*e.g.*, via DMA 146a). Those of skill in the art will appreciate that various modifications and alterations to the implementations shown in FIGS. 5 through 7 are possible, and that numerous data paths between the various components may be defined as necessary or useful for operation of a particular compute node 140n.

[0094] FIG. 8 is a diagram illustrating one implementation of a memory architecture having utility in connection with a compute node such as that depicted in FIG. 4, and FIG. 9 is a diagram illustrating one implementation of a portion of the memory architecture depicted in FIG. 8.

[0095] In some implementations, each compute node 140n may employ sixteen physical channels (or buses) for access to data store 143. In FIG. 8, these physical channels (reference

numeral 899) are depicted in vertical columns labeled, from left to right, 0, 1, . . . , 15. On each respective channel 899, one arrangement may employ one or two memory chips, each of which may comprise one or more logical units (LUNs) (reference numeral 810). In FIG. 8, LUNs associated with channel 0 are labeled 810a, LUNs associated with channel 1 are labeled 810b, and LUNs associated with channel 15 are labeled 810n. It is noted that more or fewer channels 899 may be employed, and more or fewer LUNs 810 per channel may be implemented, depending upon overall arrangement and processing functionality of a particular compute node 140n, the type of memory employed at data store 143, and a variety of other factors.

[0096] In the context of the disclosed architecture and data path approaches set forth herein, it will be appreciated that a LUN 810 may execute more than one memory access (such as a Flash access) operation concurrently, even on the same Flash (or other memory) chip. In addition to LUNs 810, each chip in the FIG. 8 arrangement may generally feature multiple planes (as indicated by the stacks of LUNs 810 – each LUN 810 in a particular stack resides at a different respective plane in the architecture). By utilizing multiple (in this case, four) planes concurrently, the available access to a given memory page, in effect, quadruples in size, thereby improving the ratio of data transfer to command bandwidth.

[0097] In the discussion that follows, LUNs 810, in particular, and data store 143, in general, are referred to in terms of Flash memory for convenience; as noted above, however, the disclosed subject matter is not intended to be limited to the particular type of memory used for data store 143, except to the extent that changing a memory type or employing a different memory technology at data store 143 may influence selection and implementation of suitable or appropriate interface mechanisms, such as at data store interface 145. In any event, references to Flash memory are by way of example only, and not by way of limitation.

[0098] In some instances, data in each LUN 810n may be stored in the form of Flash pages, one example of which may generally comprise 16Kb of data payload, plus some overhead associated with error correction code (ECC) and header information. Pages may be organized into blocks, for instance, or sets of 1024 pages. This is illustrated best in FIG. 9, which depicts a LUN 810n comprising two blocks 811 of data; more than two blocks 811 per LUN 810 are contemplated. Similarly, while each block 811 is depicted as comprising only three discrete pages 812, it may be desirable that a block 811 comprise 1024 pages, 2048 pages, or more. Each page 812, in turn, may generally comprise an index 813, which may record metadata, information, or metrics or statistics associated with the data payload of the page 812. For instance, index 813

may include ECC information and a data type indicator, a first index, a record count, an incarnation counter, various statistics (such as a minimum data value, a maximum data value, a local average value, *etc.*) and other information, or a combination of these and other types of information, with respect to data in a particular page 812. Additionally, LUN 810n may include a cache 819, which may serve as temporary storage for a page 812, or a portion thereof, for various purposes such as read operations.

[0099] For example, in order to read a given page 812, a read command is issued (for example, by compute array 142 or processing system 430, either individually or in cooperation). Such a read command may generally cause LUN 810n to copy page 812 to (on-chip) cache 819, from where it may be read rapidly across data store interface 145 and communicated to compute array 142.

[00100] As is generally known in the art, a Flash page 812 may only be written in its entirety in the context of any particular write operation; specifically, a particular Flash page 812 must be erased, in its entirety, before any new contents or data may be written to that particular page 812. This is true even if only relatively few data values (as compared to the overall contents of the particular page 812) are sought to be changed; in every case, manipulating data in a page 812 typically requires that the entire page 812 be erased and subsequently re-written with modified contents. To make matters worse, particularly in the context of Flash memory, not only must a page 812 be erased before any new value may be re-written (or altered) in that page 812, but the entire block 811 containing the affected page 812 must also be re-written.

[00101] In one implementation of LUN 810n and a method of interacting with Flash memory, erase granularity may be set as one block 811 (*e.g.*, a set of 1024 pages) as noted above; in accordance with a novel approach to an old problem, one method of updating a page 812 is to read its contents, selectively to modify whatever data are to be changed, and subsequently to write a new copy of that page 812 to a new location in memory – typically in the same block 811, or at least in the same LUN 810n, but this may not always be practical or even desired – and with an updated identifier to facilitate identification of the new copy as the definitive, updated version. As it is not economical to erase an entire block 811 (and to move all the unaltered pages 812 that may reside in the same block 811 to a new location) each time a page 812 is updated or erased, the old page 812 may remain unaltered and in its original location within the block 811.

[00102] To manage the resulting multiple copies of a page 812, each specific page 812 may be provided with an incarnation counter (or similar indicium or identifier) in its metadata; as noted above, it may be useful that this incarnation counter is updated each time a page 812 is copied during an update or modification process. In some implementations, such an incarnation number count or other identifier may be read during boot and used to populate a page pointer index (not shown) or look up table that resides in, or is accessible by, node memory 450 or processing system 430, for example. In one approach, only the copy of a particular page 812 having the highest value in the incarnation counter may be identified in the pointer index or look up table, as it is this copy that represents the active and valid copy of that page 812 – all other copies contain data that have not been updated to reflect new values. In some embodiments, the page pointer index or look up table may also be updated when a page 812 is copied and its incarnation counter is incremented. The incarnation counter may be selected to have a range that is large enough (*e.g.*, 24 bit, double check) so that wrap-around conditions (*i.e.*, a reset from the highest value in the range back down to the lowest) occur infrequently.

[00103] Upon wrap around of a page incarnation counter, two methods of exception handling may be applied. In a brute force approach, all remaining stale copies of the page 812 may be eliminated; it will be appreciated that this approach may trigger a great number of page moves, even in the case where neighboring page data are still valid. In an alternative approach, an algorithm that closely interacts with garbage collection and block recycling (described below) may be employed to track the correct highest count even after wrap around has occurred. Many such algorithms may be devised to keep track of a most recently modified version of a page 812 after a counter wraps around, and the present disclosure is not intended to be limited by any particular approach or methodology.

[00104] In some embodiments, blocks 811 may be recycled efficiently by tracking a valid page (VP) count identifier and a continuity factor (CF), where VP count may be an indication of a number of valid pages 812 written to a block 811, and CF may be an indication of a number, or percentage, of pages 812 within a block 811 that no longer contain up to date data. In accordance with this paradigm, a block 811 having a low VP and a low CF is at the beginning of its life span – not all pages 812 have been written yet, and relatively few pages 812 are out of date – and hence this particular block 811 may be assigned a low priority for recycling operations. A block 811 with a high VP but a low CF may be nearly full or totally full – most, if not all, pages 812 have been written – but the block 811 as a whole contains relatively few pages 812 that have been

subsequently invalidated by more updated versions; such a block 811 may be assigned a medium or middling priority for recycling. A block 811 with a high CF, on the other hand, may be considered a candidate for recycling having a relatively higher priority, since many, if not most, of its pages 812 no longer contain valid data.

[00105] In accordance with one strategy, as long as there are sufficient blocks 811 available in a given LUN 810n to accommodate short term copying of pages 812 without rewriting or excessive wrap around activity, no old block 811 should be forcefully recycled as long as it has space to allocate to more than one unwritten page 812; this strategy may minimize, or at least reduce, unnecessary page writes. When free resources in a given block 811 become sparse, the block(s) 811 with the highest CF and the lowest VP may be freed by moving the remaining valid pages 812 to other blocks 811 having free space, and then erasing the block 811 from which all remaining valid pages 812 have been moved.

[00106] In most conventional Flash applications, retrieving (reading) a page 812 has a latency of about 100µs, as the page 812 is first copied to an on-chip cache 819 as set forth above; this typically occurs before the page 812 is ready to be transferred across a bus (*e.g.*, via data store interface 145n) to compute array 142. The data transfer of one page 812 from that on-chip cache 819 to the compute array 142, in contrast, requires less than a third of that overall time – *i.e.*, around 32 µs.

[00107] In order to mask the foregoing latency, each physical channel 899 in FIG. 8 may accommodate multiple LUNs 810, each of which may be retrieving pages 812 to cache 819 concurrently. Once an array of LUNs 810 (or targets) is loaded with read commands, further read commands may be interleaved while data from pending reads are retrieved from cache 819. With sufficient targets (for example, four or more) on a channel 899, the available bandwidth for a given channel 899 may be fully saturated with data moves (streaming). As the LUN 810 count increases, it becomes possible to read partial pages 812 (for example, a ½ page or a ¼ page) without any substantial loss in channel utilization (interleaving the additional read commands has minor impact on overall throughput). Reading partial pages 812 efficiently becomes a valuable feature in more advanced data mapping methods (see the discussion of data structures with reference to FIGS. 10 through 14 below).

[00108] Specifically, executing commands against a LUN 810 on a Flash chip will always result in latencies and minimum (but not deterministic) wait times before a previously issued command

is executed and the LUN 810 is available for further exchanges. As noted above, the biggest latencies are related to commands that retrieve a page 812 (around 100µs) and write a page 812. During those wait times, many features of a LUN 810 are generally unavailable. The status of a LUN 810 may be polled by issuing a status request command (for instance, from storage interface component 411n or elements thereof such as routing block 413n).

[00109] As also noted above, in accordance with the disclosed architecture, payload data and command sequences share the same bus or physical channel 899. Accordingly, to maximize throughput on a given channel 899, LUN 810 access may be interleaved across many targets (*i.e.*, pages 812), and an estimate of availability of each LUN 810 may be determined and maintained in order to minimize interrogation overhead. This functionality, and that set forth below, may be provided by storage interface component 411n, in general, and routing block 413n and data store interface 145n (either individually or in combination), in particular.

[00110] In that regard, storage interface component 411n or its components may take all of these requirements into consideration, and may be configured and operative to provide enough processing functionality to reduce or to eliminate the requirement for central processing unit (CPU) intervention (either by compute array 142, processing system 430, or both) by providing hardware resources necessary or sufficient to effectuate some of all of the following (*e.g.*, in routing block 413n): provide or control a dedicated command buffer per LUN 810 of a given physical channel 899; provide or control a dedicated command execution timer per LUN 810 (this timer may be set with a minimum latency of the Flash chip after issuing a command, such that status polling only begins after the minimum delay time has elapsed, and therefore keeps the channel 899 less congested with status requests); provide an autonomous status request command generator, so that status polling does not require CPU intervention; and provide a session identifier buffer for each command that is posted by processing system 430 into the various command FIFOs in routing block 413n (this may allow processing system 430 to "set and (almost) forget" issued commands, as storage interface component 411n may post completed (or failed) transaction updates in message mailboxes 452n along with a corresponding session identifier). In that regard, mailboxes 452n may have the utility set forth above with reference to FIG. 4.

[00111] It is also noted that routing block 413n, memory interface 145n, or a combination of these and (possibly or optionally) other components of storage interface component 411n may also provide hardware, firmware, or software resources necessary or sufficient to effectuate

selective storage of data in a "co-located" manner. In this context, the terms "co-located" or "co-location" generally refer to the operational characteristic (or its practical effect) of maintaining data records that are related to each other in or at memory addresses such as are reflected by or implemented in blocks 811, pages 812, or both that are relatively proximate to each other (*e.g.*, in physical space, logical space, or both) as compared to other blocks 811 or pages 812 in data store 143. Header information or metadata associated with particular data may be used to designate or otherwise to identify certain data as related to or relevant with respect to other data for this purpose. As set forth above, data may be supplied to system 100 with information (that may be recorded in index 813, for example) that may include a data type index or identifier, various statistics, and other information that may be used, among other things, to co-locate particular data with respect to other related data in a particular page 812; in the foregoing manner, related data may be stored in adjacent or contiguous pages 812 or blocks 811, facilitating retrieval of related data using reduced or minimal read operations.

[00112] One particular implementation that may benefit from co-located data may be configured and operative to organize large datasets acquired, by way of example, from a STARE (Surveillance Technology Advancement & Replacement for Ebsicons) data acquisition system. A typical STARE system employs a variety of sensors (*e.g.*, nine or more independent sensors, typically deployed in a satellite or high-altitude surveillance aircraft) that acquire optical or other electromagnetic data across a large geographic area and at successive points in time, such as for satellite reconnaissance, weather observations, time-lapse photographic applications, and the like. While the following discussion addresses such a STARE system, the disclosed subject matter is equally applicable to other types of large datasets, as will generally be appreciated by those of skill in the art. For example, on a micro-scale, the types of data storage and analytics techniques employed in a typical STARE application may be equally applicable to computer tomography (CT) scans, computer aided detection (CAD) mammography analyses, and other medical technologies involving optical or image data, as well as to design verification and quality assurance methodologies (such as machine vision inspection techniques, for instance) in the semiconductor, micro-electrical-mechanical (MEM) machines, and other small scale electrical and mechanical industries.

[00113] In accordance with one implementation, a respective data stream from each of a plurality of data sources (for instance, each respective sensor in a STARE array or each respective channel in a medical imaging device) may be ingested as a dedicated dataset for a

given point in time (*e.g.*, a temporal STARE index). In this example, the respective output of each respective sensor or channel may be considered an independent data source, while the data from each respective independent data source may be considered (and identified, designated, or otherwise characterized as) "related" to all the other data acquired from others of the independent data sources. For instance, data acquired by a specific sensor operating in a specific spectral band on a given satellite may be considered a specific, independent data source; if that satellite employs multiple sensors, each operating in a respective (though, perhaps, overlapping) spectral band such that the satellite, as a whole, acquires and provides data across multiple spectral bands, each such independent data stream may be ingested by system 100 (and, in particular, by compute node 140n) as different, independent, but related datasets. As noted above, the relationship between such independent datasets may be encoded in metadata or header information, for example, such that related datasets may be co-located when stored or written to data store 143 or other memory component.

[00114] In the foregoing manner, and in the context of a STARE system, each independent dataset may represent a collection of data points from a specific sensor observing or dedicated to a specific geographic region at one point in time. In the case of an arthroscopic, mammographic, CT, or other procedure involving medical imagery or other data collection application, for instance, each independent dataset may represent a time-specific or temporally varying collection of data points from a specific type of sensor covering a specific anatomical region or that is dedicated to collecting a particular type of data. Applications include automotive diagnostics, time-dependent stresses in aerospace structures, inventory and logistics tracking, machine vision and machine learning implementations, and any other application in which large datasets and analytics techniques are relevant.

[00115] Returning to the STARE example, each record in a particular independent dataset may include fields to designate, for instance, a STARE index and sensor data sufficient to identify a particular data source (*i.e.*, the sensor from which the data are acquired), a position of each sample in the data stream (*e.g.*, a time stamp or other unique indicium), and any related data streams to facilitate co-location of data during storage. It is noted that some or all data acquired by some or all of the plurality of data sources may be multi-dimensional (*e.g.*, intensity, frequency, directionality or orientation, altitude or distance from sensor to imaged object, and the like, depending upon the application and the type of data collected). Some or all of these factors may be encoded into, included in, or referenced by header information or other metadata associated

with each data packet provided by the independent data source or otherwise derived by other components as a function of or dependent upon the independent data source and *a prior* or predetermined knowledge of its operational characteristics or deployment.

[00116] In some implementations, a user or administrator of system 100 may choose to divide data associated with independent data sources which cover large geographic areas (or other areas or regions of interest) with high resolution into multiple smaller datasets, each of which may include or be specifically directed to a fraction of the overall geographic area or region of interest that is covered by the data source as a whole. If this is done with a suitable amount of spatial overlap, boundary conditions that may arise from distributing the smaller datasets to multiple compute nodes 140n may be mitigated.

[00117] Those of skill in the art will appreciate that the foregoing functionality may be readily implemented, for example, via suitable programming in an application programming interface (API) and/or in logic at compute array 142, routing block 143n, or both, such that related data may be recognized as such and ingested and reproduced as indexed and related datasets in connection with STARE or other applications. In some implementations, permanent, non-intrusive detailed instrumentation may be integrated into programmable logic block 410, for instance, and cooperate with an API to effectuate appropriate hardware configurations, analysis interfaces, and setup methods to accommodate this embodiment supporting co-location of related data.

[00118] Some of the same factors that enable co-location of data also facilitate implementations in which algorithms may operate on multiple datasets. In that regard, it will be appreciated that typical algorithms employ only a single dataset for input; this is both conventional and sensible, since a typical algorithm seeks to apply the same set of rules or instruction sets to data that are related in some way, and since the relationship between particular data points is conventionally defined by the fact that the data points reside in the same dataset. Where, as set forth above, different, independent data streams may be identified as related (*e.g.*, by header information or metadata) across independent datasets to support co-location, however, such relatedness may be leveraged to enable a particular algorithm to execute on multiple datasets in a single instantiation of the algorithm.

[00119] By way of example, a typical API call to execute an algorithm only allows the use of fields that belong to a single (open or active) dataset; multiple dataset computations are difficult

to implement and require extensive programming and computational (and data record access) overhead. However, using the above techniques that contemplate importing and storing data streams as dedicated datasets from independent data sources, and since those multiple sources may be considered for thresholding calculations and other house-keeping and initiation procedures, an API or programmable logic block 410, or both, may readily be modified to accommodate use of multiple datasets by a single instantiation of an algorithm. In that regard, programmable logic block 410 may be implemented such that it allows for each field which provides data for computations to be a member of any dataset that is available (*e.g.*, stored in data store 143n or other applicable memory structure). In a STARE implementation, such algorithms may be embodied in or comprise Connected Component Labelling (CCL) algorithms or other implementations, though these and other algorithms or heuristic approaches may have equal utility in image processing and other applications as noted above.

[00120] FIG. 10 is a diagram illustrating a conventional data record structure, and FIG. 11 is a diagram illustrating one implementation of a data record structure having utility in connection with a compute node operative in accordance with the present disclosure. Further, FIGS. 12 through 14G are diagrams illustrating additional implementations of data record structures having utility in connection with a compute node operative in accordance with the present disclosure.

[00121] In most Big Data and high performance computing applications, many computations may be performed on columns of data, or vectors, rather than on a single data point or value. This typical mode of operation may indicate or suggest that one useful way to store data in Flash is to map columns to dedicated Flash pages for efficient retrieval. On the contrary, however, some applications may benefit from organizing data in records organized in such a manner that a Flash page holds multiple fields (columns) of a common record. Prior data management and data processing systems address these competing requirements by storing the data twice: once, organized in columns; and then again, organized in records.

[00122] In accordance with the disclosed architecture of system 100, since each compute node 140 may have sixteen or more physical channels 899 that may operate concurrently, a sophisticated data mapping method, examples of which are set forth below, may allow for certain features that are typically unavailable in the context of single channel storage solutions.

[00123] The following examples, and the illustrations in FIGS. 10 through 14G, are based upon a hypothetical dataset of eight records, each record consisting of eight fields or columns. In the

drawing figures, a respective color (represented by a unique hatching pattern) is used to indicate a respective column or field type in a record, and a unique number is used to indicate a row or index for each unique record. In the context of the present disclosure, the terms "record" and "page" may be used interchangeably, and the terms "field" and "column" may be used interchangeably.

[00124] In the prior art data record structure depicted in FIG. 10, for example, each distinct row (1 through 8) represents a distinct data record (numbered 1 through 8); as indicated by the legend "Records 1 – y," it will be appreciated that such a data record structure may include more or fewer records than shown. Each column in FIG. 10 represents a particular field, each of which is distinct from the others as indicated by the respective color hatching; as indicated by the legend "Fields 1 – x," it will be appreciated that each record (1 through 8) may include more or fewer fields than shown.

[00125] In accordance with the disclosed subject matter, however, when staged or reformatted for storage in Flash (such as data store 143), the data may be, in effect, rotated by 90 degrees, such that all original fields in the same original column are stored in a new dedicated page (or pages), *i.e.*, a new dedicated page comprises fields, each comprising a single field type. This is illustrated in FIG. 11. Specifically, each respective row in FIG. 11 represents a new respective record, 1 through x (eight, in this example); *i.e.*, one new record for each discrete field in the original data structure. Each new record comprises a number of fields (eight, in this example) 1 through y, corresponding to the number of records in the original data structure of FIG. 10, and each field is of the same field type mapped from the same column in the multiple records in the original data structure.

[00126] In the foregoing manner, original records may be re-mapped into columns, and original fields may be re-mapped into discrete records or pages 812, each comprising a column from the original data structure; each new record (*i.e.*, representing an original column) may be stored in a dedicated Flash page 812 of any channel 899. In the FIG. 11 example, page 812 retrieval may be executed efficiently, as set forth above, though fetching related data may be further optimized with some additional manipulation.

[00127] For example, if only one channel 899 were available, one additional mechanism to facilitate optimization may be to spread the newly formatted pages 812 across multiple LUNs 810 so that access to the various pages 812 may be executed in an interleaved manner. As noted

above, such an interleaving strategy may have a positive effect on overall throughput with respect to the transfer time associated with writing a page 812 to cache 819 in the Flash array; in particular, this specific latency type of may be masked by use of interleaving, in accordance with which other data types or records from a second LUN 810 may be transferred across routing block 413n to compute array 142 while a first cache 819 associated with a first LUN 810 is still filling. This arrangement is illustrated in FIG. 12, which depicts the data structure of FIG. 11 distributed across multiple LUNs 810a, 810b, 810c, … 810n.

[00128] In some implementations in which each compute node 140 accommodates sixteen (or more) parallel channels 899, it may be useful to distribute data traffic across all available channels 899 as described above with reference to FIGS. 4 through 9. In such an embodiment, it may be useful to distribute the data structure of FIG. 11 across the available channels 899 in the manner illustrated in FIG. 13 (only eight channels 899 are shown, for clarity). As illustrated in FIG. 13, a LUN 810a may comprise newly formatted pages 812 (each having an index 813) as set forth above; in this example, however, the newly formatted pages 812 are distributed across each channel 899 available to handle data transfers to and from the data store 143. It will be readily appreciated from examination of FIGS. 13A through 13H that the manner in which pages 812 are stored may be manipulated in accordance with a particular one of the channels 899 with which those pages 812 are associated. For each successive channel 899, for instance, a location of pages 812 associated with the same field type from the original data structure is incremented by an offset value, and each successive channel includes data from the next successive eight original records.

[00129] In the illustrated implementation, the order in which pages 812 are stored (*i.e.*, from record 1 through record 8) by original field type remains constant, but the first record 812 associated with a particular channel 899 increments by an offset value (for example, 1) for each successive channel 899. For example, a first channel 899 may be assigned or associated with a dataset in LUN 810a that begins with a first record or page 812 that represents a first field from the original data structure shown in FIG. 10, a second channel 899 may be assigned or associated with a dataset in LUN 810a that begins with a second record or page 812 that represents a second field from the original data structure, and so on. This is represented in FIGS. 13A through 13H, which depict a respective dataset for each respective one of eight channels 899 that are useable by LUN 810a (as indicated by the dashed lines in FIG. 13).

[00130] It is worth noting here that the actual (*i.e.*, physical) order in which pages 812 are stored in data store 143 (*e.g.*, a Flash array) need not change from channel 899 to channel 899 in many applications; for example, the same effect may be achieved logically or virtually by associating each series of pages 812 with an index or a look up table value that identifies a first page 812 for each channel 899. As long as a read operation (for example, requested by compute array 142 in cooperation with routing block 413n) is apprised of an index and an offset value for each channel 899, the specific order in which pages 812 are stored for any given channel 899 may be irrelevant. In one implementation, such indices and offset values may be maintained in node memory 450 (*e.g.*, in FP tables 454n) and communicated as necessary to routing block 413n via DMA 146b substantially as described above, though other alternatives may have similar or analogous utility. The depiction in FIG. 13 is provided by way of example only, and for purposes of visualization, but those of skill in the art will appreciate that the illustrated structure may be achieved virtually without undue experimentation. It is also noted that the data record structure shown in FIGS. 13 and 13A through 13H may be altered or modified to accommodate the number of physical channels 899 available to a particular LUN 810n, the bandwidth or processing capabilities of compute array 142, the number of records, fields, or both in the original data structure, or a combination of these and a variety of other factors.

[00131] In particular, the diagonal striping depicted in FIG. 13 is indicative of a unique feature provided by this approach: data may be read massively parallel as records (*i.e.*, from the original data structure in FIG. 10), on the one hand, and as pure columns (*i.e.*, from the original data structure in FIG. 10), on the other hand. The type of read may be selectively adjusted or determined by identifying a page 812 selection for each channel 899. As noted above, such a read operation may be requested by compute array 142, either independently or in cooperation with routing block 413n, and may be supported by data retrieved from node memory 450, in some applications.

[00132] If a read request for all channels 899 points to the same page 812 location (*i.e.*, the number of spaces measured from the top down, or from the bottom up, in FIG. 13), then horizontally across all channels 899, this is represented by the row at reference numeral 1399. Reading all channels 899 across the entirety of row 1399 will result in retrieval of complete records for the entire dataset in LUN 810a in parallel. This method may be useful when an application or algorithm needs to check for many conditions concurrently. As noted above, the same result may be effectuated by using a page index and offset value for each respective channel 899,

irrespective of location of a particular page 812 in a data structure maintained for that particular channel 899.

[00133] Additionally or alternatively, the patterned data record structure of FIG. 13 may also be effective in finding all individual records or pages 812 with a specific condition (or field type), for example, by setting an offset value in each channel 899 differently (*i.e.*, diagonally across) so that all channels 899 only deliver that specific condition or field type in parallel. For this search or read strategy, no reformatting of the data structure is required; both record-wide and column-centric selection may be equally effective in the context of effective storage bandwidth utilization.

[00134] To facilitate reassembly of related fields (from the newly formatted data structure) into complete original records after the different fields have been mapped to different Flash pages 812, each index 813 for a new page 812 may include, in addition to the information and metadata noted above, an indication of the field type (*e.g.*, a column number or other identifier from the original data structure in FIG. 10) and record (again, from the original data structure) with which a given page 812 should be associated, as well as an index number or other identifier of both the first and last record (from the original data structure) from which the fields contained in page 812 originated. In some embodiments, compute nodes 140 may assign a local system index to each page 812, for example, as each page 812 is received or otherwise acknowledged (*e.g.*, during boot or initialization). This index may be globally unique if combined with a node identifier and a card identifier, for instance, which uniquely identify compute node 140n and its card 200 in the context of system 100.

[00135] In one simple implementation employing eight compute nodes 140 each having sixteen memory channels, it is possible to spread the various field types across all eight compute nodes 140 for a total of one hundred twenty eight parallel field types. In the absence of sophisticated mesh connectivity, however, it may be cumbersome to accommodate compound conditions across multiple compute nodes 140 in this context, and even one hundred twenty eight field types or columns may not be sufficient to reflect many real-world applications. Accordingly, it may be appropriate or desirable in some circumstances to utilize Flash pages 812 that hold, maintain, reflect, or represent multiple field types (*i.e.*, columns from the prior art data structure of FIG. 10). This is depicted in simplified form in FIGS. 14A through 14G.

[00136] Specifically, the illustrated approach may be employed in some situations to mitigate or to eliminate shortcomings generally associated with current known limitations of block RAM

(BRAM) technologies implemented by many commercially available FPGA structures, for example, such as those having utility in connection with compute array 142. In applications requiring storage and/or retrieval of a vast number of datatypes (*i.e.*, fields) in parallel to facilitate examination or analysis of many conditions concurrently, conventional BRAM-based systems simply do not (and cannot, as a practical matter) comprise enough local storage to store or otherwise to accommodate many records with possibly thousands of fields at the same time.

[00137] Enabling multiple field types to be populated in a single page 812 of memory, and realizing benefits from reduced bandwidth overhead associated with partial page reads (made possible by deployment of multiple LUNs 810 in a high density constellation as illustrated and described herein), allows a compute node 140 to achieve higher throughput rates as compared to a strategy that requires reading specific entries from freshly retrieved entire pages 812 for each read request.

[00138] In some implementations, only minor adjustments to logic (*e.g.*, at routing block 413n, compute array 142, or both) or a few additional instructions (*e.g.*, in FP tables 454n at node memory 450) may be useful to enable partial page read functionality. For example, with mixed column types in a page 812, a normal page header or index 813 as described above may be modified such that it refers to a secondary page header describing the contents of a page 812 and a boundary between field types; using this mechanism, and by apprising relevant logic or instructions sets that partial page reads are possible, fractional compound pages such as those illustrated in FIGS. 14A through 14G may be maintained without altering the above-mentioned methods for selecting pages, pre-sorting, sorting, and so forth.

[00139] Many different applications and algorithmic processing strategies may benefit from partial page reads, particularly in situations where only data from some of the subpages, or a portion thereof, are relevant for a particular computation to be executed; by using a partial page read strategy, other (irrelevant) field types in a particular page may be ignored (*i.e.*, may not be retrieved) to economize on bandwidth and computational or other processing resources. In some such embodiments, it may be desirable to implement a strategy to rotate between active LUNs 810, and their associated resources, to maximize or otherwise to increase throughput (as in the example above addressing a single field type per page).

[00140] FIG. 15 is a diagram illustrating one strategy for data communications between a host compute system and a data processing and analytics system as depicted in FIG. 1. FIG. 15

illustrates, in simplified form, one example of certain data communications that may support the foregoing functionality in the context of a data processing operation or compute task requested by a host compute system 199. Specifically, communications (calls/receipts) operations for host compute system 199 are illustrated on the left of FIG. 15, communications passing through router module 120 are illustrated in the middle of FIG. 15, and communications operations for compute node 140 are illustrated on the right of FIG. 15.

[00141] Among the other things noted above with specific reference to FIG. 1, host compute system 199 may generally comprise a shared library 1510. In some implementations, shared library 1510 may be configured and operative to implement an appropriate application programming interface (API) to enable bi-directional data communications between host compute system 199 and router module 120 as described above and as depicted in FIG. 15. In particular, an API at shared library 1510 may be embodied in or comprise a procedural interface implemented on one or more of common or popular operating systems typical for host compute system 199; examples of such an operating system include Linux™, Windows™, and Mac™ OS™, though others may be available currently, or may be developed in the future. As is generally known, an API binding typically provides basic functionality to authenticate API requests, to enable or to facilitate text encoding/decoding, and to provide generic support for read and write requests; in some implementations, the API at shared library 1510 may employ C™ and Python™ bindings, though others may also be employed. Further, the API may parse dataset descriptions and perform basic parameter validation, as is also generally known.

[00142] As set forth above, the API may also have utility in storing and retrieving datasets in a co-located manner. For example, shared library 1510 may be configured and operative (such as by the foregoing parsing of dataset descriptions, parameter validations, or other techniques) in cooperation with the API to identify specific datasets and to characterize same as related to other distinct datasets. Such a relatedness determination, which may be binary or effectuated in accordance with a weighted scale (*e.g.*, to provide a relatedness "score" or other value for a dataset relative to one or more selected other datasets) may be used to influence a physical or logical location in shared library 1510 at which data are maintained. For example, it may be beneficial in some circumstances to store related datasets in adjacent, continguous, or alternating memory blocks in shared libary 1510. The same may also be true for memory store 143n, as noted above, and so, additionally or alternatively, components at host compute system 199 may communicate datasets to compute nodes 140 (such as illustrated in FIG. 15) with such

relatedness information encoded in metadata or header information, or even embedded within the datastream itself.

[00143] In some embodiments, shared library 1510 may be implemented as a binary libary (for example, on Windows, it may be in the form of a dynamic link library (DLL)) and may be provided as part of a simple software development kit (SDK) that includes appropriate header files, in a manner that is similar to support provided for any other generic C or Python library. In use, shared library 1510 may generally be apprised of, and may maintain, information related to open datasets for each individual user of host compute system 199 (*i.e.*, it may be desirable that information within shared library 1510 is not shared between individual users, but is rather shared between discrete resources of system 100).

[00144] As indicated in FIG. 15, host compute system 199 may further comprise a driver 1520. In operation, functionality of driver 1520 may be called by shared library 1510. In that regard, driver 1520 may employ standard input/output (I/O) calls, such as read and write, and may be operable in cooperation with Linux, Windows, Mac OS, or a combination of these and other operating systems (depending, for instance, upon the operating system governing functionality of host compute system 199 as a whole). Driver 1520 may translate dataset names to internal dataset identifiers, for example, as may be necessitated by host compute system 199 operating system or other resources, and may also provide full parameter validation, security and sharing management functionality, and administrative tasks as are generally required by operating systems and other components in host compute system 199. Specifically in the context of the disclosed subject matter, driver 1520 may manage activity of compute nodes 140 (for instance, in cooperation with individual processing systems 430) by sending commands and receiving (and, optionally, aggregating) responses as necessary during run-time. In that regard, it is noted that some of the functionality of driver 1520 may be provided in cooperation with one or more of the components of processing systems 148/430, and may influence or benefit from data in node memory 144/450, described above with reference to FIGS. 1 and 4.

[00145] Driver 1520 may also maintain and organize open datasets for all registered or authorized users of host compute system 199, maintain dataset identification data or indicia for same, and enforce administration privileges or other access rights to components or modules that are fundamental to operation of host compute system 199.

[00146] Compute node 140 may generally comprise, in addition to the components described above, a firmware module 1590 or instruction set repository as indicated in FIG. 15. Firmware 1590 may be embodied in a dedicated memory component, for example, or may reside in or be maintained by an element of processing system 430 or node memory 450, in some instances. The disclosed subject matter is not intended to be limited by the manner in which firmware 1590 is stored or deployed in the context of the various hardware and software components of compute node 140. In operation, firmware 1590 may generally implement an appropriate data transfer command/response protocol sufficient to enable the communications and functionality set forth herein; in that regard, firmware 1590 may execute or accommodate an asynchronous command packet / response packet interface to allow the communications depicted in FIG. 15 and otherwise to facilitate data transfer between compute node 140 and router module 120 in accordance with the foregoing methodologies and principles. As set forth above, this feature may also comprise one or more mechanisms supporting col-location of related datasets, either independently or in cooperation with information received from host compute system 199 and associated with the data transfer.

[00147] Specifically, firmware 1590 may, in some embodiments, be configured and operative to manage data transfers between compute array 142 and data store 143 (*e.g.*, a Flash array). For example, firmware 1590 may be responsible for, manage, or otherwise facilitate some or all of the following functionality: handling boot-up initializations for compute node 140; building and managing indices or look up tables indicative of data locations for all dataset records managed by compute node 140; creating and managing datasets and any associated dataset descriptors (including co-location indications or relatedness scores across datasets); optimizing data storage by field type per dataset; managing page 812 or block 811 write optimization protocols or strategies (including, but not limited to, co-location or dataset density approaches); managing page 812 replacement and page 812 aging strategies; loading and execution of algorithms in compute array 142 (*e.g.*, such as may be embodied in an FPGA fabric); loading relevant parameters needed or desired for algorithm execution (including whether a particular algorithm may execute on multiple, related datasets); collating and, optionally, passing data (*e.g.*, read data and/or algorithm results) between and amongst components in compute node 140; and generating dataset identifications or other indicia for dataset creation and co-location or relatedness scores, and coordinating same across all compute nodes 140 in a particular distributed processing system 100.

[00148] In the context of all datasets, firmware may maintain and organize all names, access strings, identifiers, descriptors, indices, relevant relatedness scores or co-location preferences, if applicable, and the like. For a particular dataset, firmware 1590 may also maintain a record of data pages 812 that contain dataset data and are located or that reside on or in association with a particular compute node 140. Additionally, firmware may also maintain data associated with a list of outstanding read or write requests (or both) for data that are currently being transferred. As noted above with respect to driver 1520, it will be appreciated that some of the functionality of firmware 1590 may be provided in cooperation with one or more of the components of processing system 148/430, and may influence or benefit from data stored in node memory 144/450.

[00149] Turning now specifically to the data flow illustrated in FIG. 15, a user's application program may make an API call at host compute system 199 to initiate a data processing operation. In some implementations, the API may include different routines for different functions, such as CREATE, OPEN, CLOSE, WRITE, READ, EXECUTE ALGORITHM, DELETE RECORDS, DELETE DATASET, and the like. Typically, the API expects that a dataset to be accessed will or must be specified by name and access string (password) or in accordance with some other unique identifier or identifying indicium.

[00150] In this example, shared library 1510 may field the API call and check, authenticate, verify, or otherwise confirm basic parameters. For example, if the request targets a particular dataset, then shared library 1510, either individually or in cooperation with other components of system 100, may check to ensure that host compute system 199 has that dataset open or is in the process of opening it (*e.g.*, such that the target dataset, or a relevant portion thereof, is accessible in, or via, RAM or via another short term storage medium). As another example, if the request were to create a dataset, shared library 1510, either individually or in cooperation with other components of system 100, may parse and verify a relevant dataset descriptor or other information related to creation of such a dataset. If the request were to access dataset records, on the other hand, shared library 1510 may verify valid record numbers and ranges, again, either autonomously or in cooperation with other system 100 components.

[00151] In some implementations, shared library 1510 may be responsible for building the basis for the requested or required command, for example: the dataset may be specified by name; relevant command codes and other parameters may be set; a "write" command may be formatted to specify a number, say from 1 to n, of data records to be written; a "read," "delete record," or other command may be formatted to specify a starting record number (or an index and offset as

set forth above, for instance) and a number of records to be accessed or affected; and so forth. The types of commands recognized and constructed by shared library 1510 are myriad, and may be application-specific or dynamically created, for example, as a function of the processing operation sought to be executed.

[00152] In that regard, shared library 1510 may build an "algorithm load" or "algorithm execute" command, or both, as necessary or as instructed by host compute system 199, and may retrieve, or request retrieval of, an algorithm bit stream or other pertinent instruction sets from an algorithm library or other memory storage facility maintained, for example, at host compute system 199. Upon specification of a suitable bit stream or other data and information associated with a particular algorithm to be executed, as well as selection of required or desired filters, boundary conditions, relevant variables, or a combination of these and other command parameters, shared library 1510, or a proxy, may then make a call to driver 1520, which call may include a dataset name and other relevant command parameters sufficient to initialize driver 1520 with appropriate data to begin data transfer and processing operations.

[00153] Such "I/O" calls and "Read" and "Write" data transfers are illustrated in FIG. 15 at the vertical line representing a boundary between user operations and kernel (or low level, operating system) operations in host compute system 199.

[00154] Driver 1520 may field calls from the user domain (*i.e.*, across the "I/O" call boundary illustrated in FIG. 15), such as may be embodied in or comprise a request or command string from shared library 1510. Responsive to such a command or request, driver 1520 may execute, or may ensure that some other component executes, a full validation or verification of each and every parameter in a particular command or request. For example, many commands use a dataset identifier or unique indicium or indicia for efficiency; in such instances, driver 1520 may translate a dataset name to an associated identifier for downstream use at compute node 140. As another example involving security checks and administrative privileges, driver 1520 may maintain a record of authorized users, and thus may be apprised of which users are administrative users that have attendant administrative rights or privileges. In situations in which a particular command or request is one that requires administrative rights, driver 1520 may enforce administrative access rules as dictated or suggested by other elements of host compute system 199; alternatively, driver 1520 may ignore such rules involving regulated access to system resources in instances where a particular command or request does not implicate administrative access rules or policies.

[00155] Similarly, driver 1520 may be configured and operative to maintain data indicative of which datasets, system- or enterprise-wide, that are open and currently accessible; further, driver 1520 may also maintain data or other information related to open datasets with respect to which sharing is or may be allowed, and under what circumstances (*e.g.*, sharing may be permitted if so declared when the dataset was opened or if a dataset is related to an open dataset, for instance, or when the dataset was created), as well as which records are valid within a particular dataset (*e.g.*, including a total number of valid records and a percentage of valid records as compared to all records, valid and invalid or out of date).

[00156] If driver 1520 determines that all examined criteria and parameters are within acceptable ranges (*i.e.*, a command or request is properly constructed and formatted), then a cyclic redundancy check (CRC) may be calculated for the command packet (e.g., creating a validated or authenticated command packet), and the newly validated command packet may be transmitted to compute node 140 via host interface 126 (which may be, for example, a PCIe bus in some implementations, as noted above). This data transfer is illustrated in FIG. 15 at the vertical line (labeled "PCIe Bus") representing a boundary between kernel operations in host compute system 199, on the one hand, and router module 120, on the other hand. Router module 120 may then forward the verified command packet to one or more compute nodes 140 (*e.g.*, via communications links 143) substantially as described above.

[00157] At a compute node 140, firmware 1590 may receive a verified command packet and may execute authentication procedures to verify the CRC for the packet; as described above with respect to driver 1520, firmware 1590 may also execute a full parameter check in the compute node 140 domain. Given valid parameters and successful CRC check, firmware 1590 may then execute the command specified by the verified command packet.

[00158] It is noted that a particular command may include zero, one, or a plurality of page read and/or write operations, and instructions for these may be parsed by firmware 1590 and transmitted to elements of compute array 142, processing system 430, or both for execution.

[00159] For algorithm load and algorithm execute commands, the algorithm bit stream provided by shared library 1510 may be loaded into compute array 142, processing system 430, or both, and executed substantially as set forth above with reference to FIGS. 1 and 4. For example, selected data records may be streamed to and from compute array 142, generally in cooperation with routing block 413n, and multi-channel algorithm results may be combined into a

single result (for example, by compute array 142, routing block 413n, or both, either independently or in cooperation with processing system 430).

[00160] A response packet (illustrated at the top center of FIG. 15) generated at compute node 140 may generally describe what happened during data processing operations, and may provide, include, or reference a result (or an analysis of results) generated during computational execution of a relevant algorithm. Data payload in a response packet may include some or all of the following: a status of an ongoing processing operation (*e.g.*, percent complete, estimated time to completion, estimated compute cycles, estimated node resources that must or may be brought online, *etc.*); how many and which particular dataset record(s) are involved or may be implicated or affected by an ongoing processing operation; and what (interim) results have been obtained so far, along with an accounting or other report of such results. Most command packets generally expect a single response, but that is not true for all command packets; in some implementations in which command packets may expect or allow for multiple responses, those multiple responses may be aggregated, either at compute node 140 or at driver 1520, for example, as a function of system design choice or simply to decrease or otherwise to optimize data transfer efficiency or bandwidth utilization.

[00161] In any event, a CRC may be calculated for the whole response packet as generated by elements of compute node 140. Firmware 1590, or a surrogate or delegated component, may then transmit response packets back to host compute system 199 (*e.g.*, via router module 120, across host interface 126 bus, and to driver 1520). Upon receipt of a response packet (*e.g.*, from host interface 126), driver 1520 may then verify the CRC associated with or included in the response packet, ensure that the response matches or is coordinated with outstanding or pending command packets, and manage these and other operations in the context of a multi-access environment in which multiple compute nodes 140 are communicating with a single host compute system 199. In that regard, driver 1520 may aggregate responses, for instances, from a single compute node 140 or from multiple compute nodes 140, and may return such responses, as well as any appropriate or attendant status information to shared library 1510. At this point in some implementations, shared library 1510 may then take control of program execution, update its data structures based upon information in received responses, and send new or updated data to a particular user's buffer or other temporary, user-specific data storage area. Program control may then be returned to a user's program or other software application that requested the data processing operations in the first place.

[00162] FIG. 16 is a functional flow diagram illustrating aspects of one implementation of a method of reformatting a data structure. As indicated at block 1601, a method of reformatting a data structure may begin by receiving a block of data, the block of data comprising a number, y, of original records wherein each original record comprises a number, x, of original fields. See, *e.g.*, FIG. 10 for an illustration of one example of such an original data record. The receiving operation depicted at block 1601 may be executed, for instance, by compute array 142 as illustrated in FIGS. 1 and 4; additionally or alternatively, this operation may be executed by, or supported by, one or more components of processing system 148/430 as described above with reference to FIGS. 1 and 4.

[00163] The method may proceed to an iterative loop as indicated at block 1602 and decision block 1620. In accordance with this iterative procedure, a number, x, of new data records may be defined, each respective new record comprising y new fields, wherein each respective new field corresponds to a respective original field from a respective original record. If a determination is made (*e.g.*, at block 1620) that additional iterations are required, the method may loop back to block 1602; otherwise, the method may proceed to block 1603. At block 1603, the block of data may be reformatted, responsive to the defining operation (at block 1602) such that a reformatted block of data comprises the new data records, each comprising a single field type. See FIG. 11 for an illustration of one example of a data structure that may result from the operations depicted at blocks 1602, 1620, and 1603. In particular, the operations depicted at these blocks are set forth in detail above with specific reference to FIGS. 10 through 14.

[00164] In this example, the method may continue by associating each of the new data records with an index operative to identify a particular new data record in the reformatted block of data. These indices are illustrated at reference numeral 813 in FIG. 11, for example. The method may then proceed by writing the reformatted block of data to memory (block 1605), whereupon the method may conclude (block 1699).

[00165] To summarize the operations depicted in FIG. 16, particularly in the context of FIGS. 1, 4, and 11, it will be appreciated that a method reformatting a data structure may result in a newly formatted data structure that enables flexible and efficient processing operations and data access in cooperation with conventional memory hardware architectures; for example, vector processing or traditional processing may be selectively implemented with minimal administrative overhead.

[00166] It is noted that the arrangement of the blocks and the order of operations depicted in FIG. 16 are not intended to exclude other alternatives or options. For example, the operations depicted at blocks 1603 and 1604 may be reversed in order, or they may be made to occur substantially simultaneously in some implementations. Further, one or more of these operations may occur substantially simultaneously with the operations depicted at blocks 1602 and 1605 in instances where it may be desirable to do so, *e.g.*, for efficiency, where processing resources are sufficient, when multithreaded or parallel operations are accommodated by the compute array 142 and attendant hardware, and the like. Those of skill in the art will appreciate that the foregoing subject matter is susceptible of various design choices that may influence the order or arrangement of the operations depicted in FIG. 16.

[00167] FIG. 17 is a functional flow diagram illustrating aspects of one implementation of a method of executing data processing operations. As indicated at block 1701, a method of executing data processing operations may begin by providing a compute node communicatively coupled to a host computer and operative to manage data processing operations independent of the host computer. Such a compute node may be embodied in or comprise elements described above with reference to compute node 140; similarly, the communicative coupling may be embodied in or comprise elements of router module 120 such as described in the examples above.

[00168] If a determination is made (*e.g.*, at decision block 1720) that additional nodes are required or desired for a particular data processing operation or application, then the method may loop back to block 1701; otherwise, the method may proceed to block 1702.

[00169] The method may continue by providing a programmable logic component in each compute node as indicated at block 1702; each programmable logic component may be configured and operative to execute data processing operations in cooperation with a first memory component. In some disclosed embodiments, the programmable logic component (e.g., compute array 142 in FIGS. 1 and 4) may be embodied in or comprise a microprocessor, a multicore microprocessor, an FPGA, or other digital data processing component suitable to execute the functionality set forth above. Also as noted above, the first memory component may be, or have access to, a non-volatile data storage device such as EEPROM or Flash memory, though other types of memory are contemplated.

[00170] In some disclosed implementations, the method may proceed by providing a data mover component in the compute node that is configured and operative to facilitate data communications between the programmable logic component and a second memory component (block 1703). The second memory component may include any of various types of solid state storage hardware, such as RAM, SRAM, DRAM, DDR RAM, DDR4 RAM, and the like. The data mover component may utilize hardware and firmware components that are operable substantially as set forth above with reference to FIGS. 1 and 4.

[00171] The method may proceed, as indicated at block 1704, by providing a set of instructions that enable the programmable logic component to reformat a block of data comprising original records, each original record comprising a plurality of fields having different field types, such that a reformatted block of data comprises new records, each new record comprising a plurality of fields of a single field type from the original records. Aspects of an example of such reformatting operations are described above with reference to FIGS. 10 through 14.

[00172] The method may then proceed by utilizing a plurality of communications channels to transfer data associated with the reformatted data block between the programmable logic component and the first memory component (block 1705), whereupon the method may conclude (block 1799).

[00173] As with FIG. 16, the arrangement of the blocks and the order of operations depicted in FIG. 17 are not intended to exclude other alternatives or options. For example, it will be appreciated that in accordance with one embodiment, the operations depicted at blocks 1702, 1703, and 1704 may be executed in any order without materially impacting results of the method or the functionality of any particular hardware implementation utilized to execute the method. In addition to the alternatives set forth in detail above, various design choices that may influence the order or arrangement of the operations depicted in FIG. 17 will be readily apparent to those of skill in the art.

[00174] FIGS. 18 and 19 are functional block diagrams illustrating components of various implementations of a compute node having utility in connection with a data processing and analytics system. It will be appreciated that the architectures illustrated in FIGS. 18 and 19 may be implemented as variations of compute node 140, and that node 1800 may be similar in overall architecture to compute node 140 depicted in, and described in detail with specific reference to, FIGS. 1 and 4. As illustrated in FIG. 19, node 1900 is similar to node 1800, though the former

may omit a data store component (reference numeral 143 in FIGS. 1 and 1800) that is included in or accessible by the latter.

[00175] As set forth above with reference to FIG. 1, a communications link (not illustrated in FIGS. 18 and 19) may be employed to couple compute nodes such as 140, 1800, and 1900 at different successive tiers in series. As noted above, such a communications link (reference numeral 149 in FIGS. 1 and 4) may be based upon and operative in accordance with standard serial interfaces or protocols, though some proprietary serial links may also be used as a design choice. For instance, it may be desirable that a communications link employ or comprise a Xilinx Aurora™ communication link. In any event, such a communications link may be selected, configured, and operative as a function of, or otherwise taking into consideration, the hardware and operation of a compute array 142 (or "algorithm framework"). Those of skill in the art will appreciate that many options exist for both compute array 142 and a communications link, and that specific commercial embodiments have been identified by way of example only, and not by way of limitation.

[00176] The "Flash Mass Storage" functional block illustrated in FIG. 18 may be embodied in or comprise the data store 143 described above with reference to FIG. 1. In that regard, data store 143 may be embodied in or include Flash memory or other EEPROM or non-volatile storage medium. A suitable interface, such as data store interface 145 in FIG. 1 (not shown in FIG. 18), may employ a protocol that is necessary or desirable to allow compute array 142 and algorithm program logic 1890 to communicate with (*e.g.*, manage read and write access to) data store 143 as is generally known in the art.

[00177] The "Firmware Workspace" components illustrated in FIGS. 18 and 19 may be identical, similar, or analogous to node memory 144 described above with specific reference to FIGS. 1 and 4. In that regard, node memory 144 may be implemented as or comprise DDR4 RAM or other volatile or non-volatile memory suitable to support operations of node 1800 or 1900, in general, and processing system 148, in particular. As noted above, node memory 144 may be employed to support algorithm execution in some compute-intensive situations. It may be desirable or useful in certain circumstances that node memory 144 comprise, or at least have access to, dedicated or predetermined addresses or blocks of memory (such as algorithm memory 1841) that may be used, *e.g.*, in an "on demand" or "ad hoc" manner, as processing loads or bandwidth requirements dictate as set forth below.

[00178] In that regard, algorithm memory 1841 may be implemented as pre-allocated addresses, blocks, or sectors of node memory 144, for instance, or it may be dynamically allocated on an "as available" basis as processing requirements vary over time. The latter approach may not reserve or pre-allocate large chunks of node memory 144 when it is not needed, but may also ensure that sufficient algorithm memory 1841 may be available to support complex computational tasks. In the illustrated examples, algorithm memory 1841 may generally comprise resource-specific data storage cells or memory locations (such as the FP tables 454n, algorithm buffers 456n, and FP buffers 458n illustrated in FIG. 4), and may generally be operative to store data and instructions necessary or desirable for processing system 148 to perform write/read/execute operations in support of programmable logic block 410 as is generally known.

[00179] Those of skill in the art will appreciate that the present disclosure is not intended to be limited to any particular memory management techniques that may enable or facilitate allocation of algorithm memory 1841 in the context of using node memory 144 in connection with particular drivers, memory controllers, or address allocation methodologies.

[00180] In the illustrated architectures, processing system 148 (which may be similar or analogous to processing system 148 of FIGS. 1 and 4) may generally handle read/write commands and other memory access with respect to data store 143 (*e.g.*, via a suitable data store interface), receive and act upon interrupt requests (*e.g.*, with respect to operations executing at compute array 142), and regulate, control, or otherwise influence intra-node data transfers across the various components of nodes 1800 and 1900. For instance, in one embodiment described above, processing system 148 may comprise a set of one or more data processing elements or hardware components to manage all interrupt handling as well as AXI or other intra-node data transfers, on the one hand, and a different set of one or more processing components to manage general system administration (or "house-keeping") and management tasks for a node 1800, 1900 as well as algorithm execution support for compute array 142, on the other hand. While aspects of this implementation are described above with reference to FIG. 4, FIGS. 18 and 19 illustrate that different component arrangements and task distributions are within the scope and contemplation of the disclosed subject matter.

[00181] As depicted in FIGS. 18 and 19, processing system 148 may comprise processors 439 (or processing "cores") such as described above as well as additional processing cores 1871 that may be used for algorithm processing or execution support as set forth in more detail below. In that regard, processors 1871 may be analogous to (or implemented in addition to) processors

439c and 439d in FIG. 4, and may generally be embodied in or comprise digital processing hardware and firmware that is similar to or identical to that employed in processors 439, though it may be desirable in some instances to use different hardware having different processing capabilities or operational characteristics for processors 1871 as compared to processors 439, particularly in the case where processors 1871 are dedicated to algorithm execution support (as opposed to a situation where a generic core may be used for algorithm execution only on a part-time or as needed basis). It is noted that bus structures, power sources, controllers, internal memory, and other attendant and supporting hardware and communications paths have been omitted from FIGS. 18 and 19 for clarity. Processors 1871 may be implemented as or generally comprise a microprocessor, a multi-core microprocessor, a microcontroller, an FPGA, a PLC, a PLU, or some similar or analogous hardware component suitable for data processing as is generally known in the art and described above with reference to processors 439.

[00182] In operation, processors 439 and 1871 may be responsible for a variety of tasks which, under some circumstances, may be requested or required by processors 432a and 432b (see FIG. 4), or which may be initiated autonomously or otherwise without direction from external components. In various implementations, processors 439 and 1871 may perform some or all of the following functions, or a subset thereof, as requirements of system 100, in general, or node 1800, 1900, dictate: data transformation and storage (*e.g.*, with respect to node memory 144, memory store 143, or both); data dispatch for write/read operations (again, with respect to node memory 144, memory store 143, or both); command interpretation and execution (*e.g.*, with respect to data processing and algorithm compilation and execution); algorithm load and compute; and memory exception handling. Those of skill in the art will appreciate that various other functions may be assigned to or attributable to processing system 148 as a function of desired operability of node 1800, 1900, and that any of numerous hardware arrangements, computing architectures, and distributed processing resources may be employed at processing system 148 to enable its functionality in the context of supporting desired capabilities of node 1800, 1900. Specifically, processors 1871 may be dedicated (either on a full-time or as needed basis) to support the algorithm execution functionality described in more detail below.

[00183] Further, and to support algorithm execution functionality as necessary or desired, processing system 148 may additionally comprise or have access to a memory component 1870. Memory component 1870 may be dedicated to facilitate processing system 148 functionality, and may be embodied in or comprise an Embedded Multimedia Card (EMMC) firmware Flash device

or other EEPROM or non-volatile storage medium. By redirecting read/write operations from node memory 144 to a more local and dedicated memory component 1870 for computationally intensive algorithm execution operations, processing system 148 may realize performance advantages in high-demand applications.

[00184] As with the FIG. 4 approach, programmable logic block 410 is illustrated in FIGS. 18 and 19 as generally comprising a compute array 142 or algorithm framework, which may in some instances be communicatively coupled to data store 143 by a suitable hardware interface (such as storage interface component 411n in FIG. 4) that is omitted from FIG. 18 for clarity. Aspects of the interoperability of programmable logic block 410 and data store 143 are described above with reference to FIG. 4, and it is noted that the FIG. 18 arrangement is not intended to be limited to any particular memory controller or communications protocol.

[00185] In the implementations of FIGS. 18 and 19, compute array 142 is depicted as further comprising or embodying an algorithm program logic functional block (or "APL," reference numeral 1890). As noted above, compute array 142 may employ or comprise any of various commercial embodiments of an FPGA that may be selected in accordance with any of a number of factors or boundaries conditions that are typical in electronics design. In this case, compute array 142 may be selected to accommodate operation of APL 1890 as described below, as well as to support other required or desired functionality of compute node 1800, 1900.

[00186] In operation, processing system 148 may cooperate with programmable logic block 410 for data transfer and read/write operations with respect to data store 143 (FIG. 18) or, more generally, to support algorithm execution at node 1800, 1900.

[00187] FIG. 20 is a functional block diagram illustrating components of an implementation of a management node having utility in connection with a data processing and analytics system. A management node 2000 such as depicted in FIG. 20 may be one implementation of router module 120 such as described above with reference to FIG. 1. In that regard, management node 2000 may generally comprise a router 122 portion, a router memory 124 portion (which, as noted above, may include one or more router memory components 124a, 124b, 124c, ... 124n), and one or more node interfaces or communications links (not illustrated, but described above with reference to FIG. 1). It is noted that attendant power supplies, bus structures, and memory controller elements are omitted from FIG. 20 for clarity, but are well-known in the art.

[00188] Management node 2000, in general, may also include a host interface 126 to communicate with a host compute system 199. As set forth above, host interface 126 may be operative in accordance with PCI or PCIe hardware and associated communications protocols such as Ethernet™, SATA, or other suitable communications technologies. The specific hardware and communication strategy embodied by or incorporated into host interface 126 may be selected in accordance with bandwidth requirements, cooperating hardware at host compute system 199, functionality and operational characteristics of router 122 portion of management node 2000, or a combination of these and a variety of other factors.

[00189] As set forth above with reference to router module 120, one or more node interfaces or communications links may be implemented as a network interface card, a wireless or inductive hardware component, or a communications port, jack, or plug, for instance, and may enable router 122 portion of management node 2000 to communicate with a respective compute node 140, 1800, and 1900 as is generally known in the art. In use, router 122 portion may be operable to transmit data packets between host compute system 199 (via host interface 126) and a respective compute node 140, 1800, 1900 (via an appropriate node interface) in cooperation with routing information (destination address, packet header data, flood domains, *etc.*) and other short term data (look up tables, resource availability, performance or load balancing data, *etc.*) stored in or accessible by router memory 124 portion. Router memory 124 portion may be embodied in or comprise, by way of example, volatile memory such as RAM, SRAM, DRAM, DDR RAM, and the like. Router memory 124 portion, in general, or individual memory components such as 124a – 124n, in particular, may employ dedicated or dynamically allocated memory storage areas or buffers (such as result buffers 2091) to support algorithm processing in high-demand applications. As noted above, the disclosed subject matter is not to be limited to any particular memory management or resource allocation strategy, as many techniques and hardware architectures may be employed to facilitate the functionality described herein.

[00190] In one implementation, router 122 portion may execute all routing functionality (such as described above with reference to FIG. 1) using, or in cooperation with, algorithm program logic (APL) 2022 embedded in an algorithm hardware and/or firmware framework 2021. As with APL 1890 described above, APL 2022 may employ or comprise any of various commercial embodiments of an FPGA that may be selected in accordance with any of a number of factors or design choices. In this case, APL 2022 may be selected to accommodate operation of node 2000 in connection with algorithm processing as described below, in addition to support other required

or desired functionality of router 122 portion set forth above. As many types of hardware implementations are generally available, the structure and operation of router 122 portion and router memory 124 portion of management node 2000 are not described further.

[00191] FIG. 21 is a functional block diagram illustrating one implementation of an algorithm processing system using a management node. FIG. 21 depicts one arrangement utilizing a matrix (of size n X t) of distributed processing resources, wherein the matrix comprises an arbitrary number of processing pipelines (1, 2, ... n), each of which includes an arbitrary number of processing tiers (1, 2, ... t). Other arrangements, for instance, in which the pipelines do not have the same number of tiers, may also be employed in some circumstances as a design choice or as a function of computational requirements or other system parameters. In some implementations, the number of pipelines may be limited to four, eight, or sixteen, for example, depending upon the nature and operational characteristics of the nodes 140, 1800, 1900, and 2000, for instance, or in accordance with processing capabilities, power consumption, size requirements, economic, or other constraints.

[00192] As illustrated in FIG. 21, each processing pipeline (1 through n) may employ a Flash/Compute node 1800 at tier 1 (in some of the description that follows, such a Flash/Compute node 1800 may be referred to as a "memory-supported" compute node, owing to the fact that it has access to, or may incorporate, memory store 143). In some implementations, the tier 1 node 1800 in each pipeline is assigned a particular algorithm processing operation, and may use an associated memory store 143 to support such an assigned operation. To facilitate such processing, however, in addition to making use of available local resources and processing capabilities, node 1800 may employ (*e.g.*, via delegation of tasks, passing of interim results, or using alternative resources downstream in its particular pipeline) additional resources in downstream tiers (tiers 2 through t, for example). In the example illustrated in FIG. 21, each pipeline employs Compute nodes 1900 in tiers 2 through t; as noted above, each node 1900 may omit on-board or accessible memory stores 143, while still having computational capabilities sufficient to support activities delegated or requested by a respective upstream node 1800 (or node 1900, as the case may be) in its particular pipeline. It will be appreciated that a node 1900 at a particular tier in a particular pipeline may be replaced with or supplemented by a node 1800 having access to memory store 143 where desirable or appropriate for a given application or system configuration. Further, it is noted that FIG. 21 has been simplified to show only a single set of slaved, serialized Compute nodes 1900 per Flash/Compute node 1800 in a particular

pipeline; this may not always be the case, as a single Flash/Compute node 1800 may support multiple, parallel pipelines of Compute nodes 1900 in some circumstances, depending upon, for example, the nature, operational characteristics, and bandwidth of the communications links 149 employed, the sophistication of compute arrays or algorithm frameworks 142, 2021 and their respective encodings, or a combination of these and a variety of other system parameters.

[00193] At the terminus of each processing pipeline, a terminal node (*i.e.*, at tier t in FIG. 21) may provide results to a management node 2000 as illustrated in FIG. 21. Though the illustrated arrangement depicts each respective pipeline feeding results to a single management node 2000, other possibilities are contemplated. Similarly, the management node 2000 at the terminus of each pipeline may be the same management node 2000 that originally assigned the processing tasks to the Flash/Compute nodes 1800 at tier 1 of the pipelines, or it may also be a different management node 2000 in other applications or hardware architectures.

[00194] FIG. 22 is a functional flow diagram illustrating aspects of one implementation of a method of executing data processing operations. As indicated in FIG. 22, one method of executing an algorithm in a distributed processing compute environment may begin by providing an initiating compute node comprising a memory store and a first node memory (see block 2201). In one such implementation, an initiating node may be embodied in or comprise a Flash/Compute node 1800 at tier 1 of a pipeline as described above with reference to FIGS. 18 and 21. This initiating compute node may engage in bi-directional data communication with a host compute system 199 substantially as set forth above.

[00195] A method may continue, as indicated a block 2202, by serially coupling one or more additional compute nodes to the initiating compute node to establish an execution pipeline comprising the initiating compute node and the one or more additional compute nodes, each respective one of the one or more additional compute nodes comprising a respective second node memory. In some implementations, the one or more additional compute nodes may not include or have access to a memory store; in such circumstances, these additional compute nodes may be embodied in or comprise Compute nodes 1900 such as those described above with reference to FIGS. 19 and 21 or similar hardware constructs.

[00196] The execution pipeline may be initiated by loading a bit stream comprising an instruction set related to execution of the algorithm into a respective programmable logic space of each respective compute node in the execution pipeline (block 2203). As noted above, loading

a bit stream into each compute node responsible for execution of the algorithm may be desirable in many instances, providing efficiencies and increasing data flow through a particular pipelined architecture. In that regard, it is noted that information in the bit stream may be used in several operations depicted in FIG. 22, as well as in other methods operative in accordance with the disclosed subject matter.

[00197] For algorithm processing, a method may continue with executing a first operation associated with the algorithm at the initiating node using information in the bit stream and data retrieved from the memory store, and passing results of the first operation to a first additional compute node in the execution pipeline (block 2204). As noted above, some implementations may execute this functionality without storing the results of the first operation in the first node memory.

[00198] As described above with specific reference to FIG. 21, the first additional compute node and each subsequent additional compute node in the execution pipeline may execute a respective additional operation associated with the algorithm using information in the bit stream and results of a respective preceding operation, and may then pass respective additional results of the respective additional operation to a next additional compute node in the execution pipeline (block 2205). As with block 2204, in some circumstances, it may be desirable that the operations depicted at block 2205 occur without storing the additional results in the respective second node memory of each respective additional compute node.

[00199] In accordance with the disclosed subject matter, a method may conclude by terminating execution of the algorithm at a management node that receives the additional results from a terminal additional compute node in the execution pipeline, the terminating comprising executing a final operation to produce a final result of the algorithm (block 2206). The management node may be embodied in or comprise a Management node 2000 substantially as described above with reference to FIGS. 20 and 21, and may execute the functionality described above.

[00200] As with FIGS. 16 and 17, the arrangement of the blocks and the order of operations depicted in FIG. 22 are not intended to exclude other alternatives or options. For example, it will be appreciated that in accordance with one embodiment, the operations depicted at blocks 2201 and 2202 may be executed substantially simultaneously, or may be integrated into a single operation. As another example, the operation depicted at block 2203 may be executed

concomitantly with the operations at block 2201, 2202, or both, where compute nodes are selected in real-time or near real-time as a function of execution requirements, for example, which may be determined by the sophistication of the bit stream. Further, those of skill in the art will appreciate that the operations depicted at blocks 2204 and 2205 may occur substantially simultaneously, as a subsequent or downstream compute node is executing a next step while one or more upstream nodes are executing preceding steps in an algorithm processing operation. These and other such alternatives may readily be effectuated without materially impacting results of the method or the functionality of any particular hardware implementation utilized to execute the method. In addition to the alternatives set forth in detail above, various design choices that may influence the order or arrangement of the operations depicted in FIG. 22 will be readily apparent to those of skill in the art.

[00201] Those of skill in the art will appreciate that the foregoing distributed processing system may be particularly suited to complex algorithm processing operations, and that the platform architecture may provide performance and efficiency advantages in many applications. In particular, the disclosed subject matter may have utility in connection with a pipeline architecture machine in accordance with which algorithm execution parameters may be programmed as processing pipelines using various facilities implemented in hardware and firmware resident at or accessible by the nodes 140, 1800, 1900, and 2000 described herein. In some implementations, for example, data may be streamed from a memory component (such as memory store 143) into an execution pipeline (such as depicted in FIG. 21) generally at the maximum clock rate of the algorithm framework 142, 2021 processing components that form or support the nodes in the pipeline.

[00202] As set forth above, an aspect of the disclosed subject matter that facilitates implementing an algorithm to execute in a particular processing node is the compute array or algorithm framework integrated into the programmable logic 410 space (see FIGS. 4 and 18-20) of the Flash/Compute nodes 140, 1800, or, when a module or card (see FIGS. 3B and 3C) is so configured, its Compute nodes 1900 or management node 2000. As noted above, nodes 1900 and 2000 may simply be implemented as Flash/Compute nodes 140, 1800 with no memory store 143 component, and the former may serve in such an arrangement as additional compute accelerators.

[00203] In operation of an algorithm processing application, most single pass algorithms may be distributed as binary bit streams to be loaded and executed in the programmable logic 410

space of one or more nodes 1800 (see FIGS. 4 and 18).More complex algorithms may also utilize other facilities or resources of a particular node or a combination of nodes (140, 1800, 1900, or 2000) such as is illustrated in the pipelines of FIG. 21. These other facilities or resources may include: programmable logic 2022 space of a management node 2000; node memory 144 at nodes 1800, 1900, router memory 124 at node 2000, or both; one of more processors 439, 1871 at a node 1800, 1900; or a combination of the foregoing or other resources.

[00204] Thus, to execute a given algorithm, a bit stream loaded into each node 1800, 1900 may also be accompanied by a bit stream for a management node 2000 that may be tasked with overseeing or managing the relevant or related pipeline (see FIG. 21) and/or executable code to be processed by a local processing system 148, 430. Whether implemented by a single node based bit stream or by a more complex combination of node 1800, 1900 and management node 2000 facilities, all algorithms may be managed by firmware and a proprietary algorithm framework (PAF) code that is part of every bit stream and complements an algorithm's specific bit stream in the programmable logic 410, 2022 space in a given node. For example, FIG. 20 illustrates some functional blocks of a management node 2000 and its facilities, including APL 2022 integrated with or embedded in algorithm framework 2021. These facilities may be operative in accordance with the present disclosure to support the functionality described herein, as will be appreciated by those of skill in the art.

[00205] From an application programmer point of view, the complexities of this approach to algorithm processing may be completely ignored. In that regard, application programmers may simply specify the algorithm to be executed, provide parameters for the algorithm (including the name and selection criteria for the dataset upon which the algorithm is to be executed), and request execution. It is noted that an application programmer may also identify datasets that are related to the named dataset, or that the disclosed algorithm execution methodologies may identify such related datasets without input, depending upon the manner in which the datasets are identified as related in the memory store paradigm. In this manner, a single instantiation of an algorithm execution process may operate on multiple datasets efficiently and without undue programming or experimentation on the part of the application programmer. The facilities or system resources used or required by that algorithm, how nodes 140, 1800, 1900, and 2000 may interact to access data and to execute the algorithm, how much memory is required or where the appropriate memory hardware resides, and other functional and operational details may be entirely transparent to application programmers.

[00206] Some use cases are provided below.

[00207] In the context of the present disclosure, the term "Level 1" algorithm is intended to mean an algorithm that may be implemented in accordance with the present system and method using just the programmable logic 410 space of one or more Flash/Compute nodes 140, 1800 (*i.e.*, those nodes incorporating or having access to a memory store 143). These Level 1 algorithms may make use of a single processing pipeline (say, pipeline n-1 in FIG. 21), which may be replicated and executed in parallel across a plurality of nodes 140, 1800 in the pipeline. A Level 1 algorithm may be complex, large (in terms of number of execution steps), or both; a Level 1 algorithm may even be recursive, but it may generally be characterized by the fact that it has little or no need for intermediate data storage. With the foregoing as context, it will be appreciated that a Level 1 algorithm may interface with a memory store 143 and return results via the PAF (*e.g.*, compute array 142 in FIGS. 4 and 18) under the control of firmware resident in the programmable logic 410 space.

[00208] Examples of such Level 1 algorithms are: basic mathematics and statistics functions such as min, max, average, *etc.*; complex data searches and selections; and more complex mathematical computations such as Fast Fourier Transforms and the like.

[00209] In the context of the present disclosure, the term "Level 2" algorithm is intended to mean an algorithm that generally requires one or more blocks of significant amounts of data storage for intermediate results between the various steps in an associated processing pipeline. It is noted that "significant amounts" of data storage may vary as a function of overall processing requirements, hardware capabilities, throughput or bandwidth thresholds, and other system parameters. In that regard, a typical Level 2 algorithm may be characterized as such because, in order to satisfy or to exceed some performance threshold or processing bandwidth requirement, which may be pre-determined or dynamically calculated or adjusted, the algorithm must or beneficially should: pass intermediate results to slaved nodes 1900; store those intermediate results in a temporary data store for use in connection with subsequent steps within the overall algorithm execution; or both. Specifically, a Level 2 algorithm in this context may be characterized as a series of execution layers in which each execution layer essentially operates on a Level 1 algorithm, the output (*e.g.*, a significant quantity of data) of which is necessary for operations to occur at a subsequent execution layer. In accordance with the disclosed subject matter, such Level 2 algorithms may be executed in a variety of ways.

[00210] In one implementation, for example, a Level 2 algorithm may be approached as a series of simple (*i.e.*, Level 1) algorithms stored within the programmable logic 410 space of a Flash/Compute node 140, 1800, each of which may be configured and operative to feed its output into node memory 144 (of the associated node 140, 1800); the input for each such simple algorithm at each tier in a particular pipeline may be stored results read from either the associated memory store 143 or the node memory 144 at a previous layer (*i.e.*, from the node 140, 1800 in a preceding tier in the pipeline).

[00211] As another example of implementation, a Level 2 algorithm may be approached as a series of simple algorithms stored with the programmable logic 410 space of a Flash/Compute node 140, 1800, and also stored in the programmable logic 410 space of one or more Compute nodes 1900; in this embodiment (illustrated in FIG. 21), each node 1800, 1900 may feed intermediate results or data to the next to form a pipeline. Each node 1900 in a particular pipeline may be slaved to the node 140, 1800 at tier 1, and each particular pipeline may ultimately terminate at the APL 2022 of a management node 2000.The final result of the algorithm execution may be computed (*e.g.*, at management node 2000) and returned to a host compute system 199. In this case, instead of writing intermediate results to short-term storage (*e.g.*, at node memory 144), such results may be transmitted or handed off by the i-1th node (1800, 1900, as the case may be) to the ith node in the slaved pipeline (see FIG. 21). As the cascade of intermediate results propagates through the pipeline, the tier 1 node 1800, for example, may then accept the next data element(s) (and any additional arguments) from data store 143 (or its equivalent) while the tier 2 node 1900 continues to execute concomitantly or substantially simultaneously, ultimately passing its output to the tier 3 node 1900 in the pipeline, whereupon the tier 2 node 1900 may then be ready to accept the next input from the initiating tier 1 node 1800, and so on. The terminal node in a given pipeline (*e.g.*, the tier t node 1900 in FIG. 21) may finally pass or transmit its results to the management node 2000 for any final computation steps (*e.g.*, using APL 2022), after which a final result may be buffered for returned to host compute system 199, and each relevant tier 1 node 1800 may be notified that a result is ready to be returned.

[00212] One example of a Level 2 algorithm as contemplated herein is a common inference algorithm for a neural network executing under TensorFlow™ (an open-source platform for machine learning, artificial intelligence training, and the like). As is generally known in the art, neural networks are typically organized into layers (or tiers): all the neurons (or nodes) in a given tier are tasked with executing processing operations using both input data for that tier (and per

node, as directed by the network) and a coefficient (as supplied by the network) for each node in the tier. The output data or results of the inference processing operations at each node may be quite large; typically, these data are transmitted or fed to the next subsequent tier downstream in the processing pipeline, after which they may be discarded by the node transmitting the results. The final or terminal tier's output (or a computation based upon output from all the nodes in the terminal tier) is the result of the inference, and may be returned to the application that requested the inference that instantiated the algorithm processing in the first place. In essence, then, the basic TensorFlow computational element may be considered a Level 1 algorithm that accepts primarily two inputs or arguments: a coefficient which is computed from results of original training of the particular relevant neural network; and an element of data, which is either taken from (in the case of the first layer or tier of the neural network) the dataset that is the target of the inference process, or from (in the case of a subsequent layer or tier of the neural network) output data from a previous tier in the neural network.

[00213] In the context of the present computing architecture such as illustrated in FIGS. 1 and 4, for instance, the foregoing pipelined processing methodologies may be implemented to run exclusively within a series of Flash/Compute nodes 140, 1800, for example, by storing intermediate results from processing within a given tier in node memory 144 of an applicable node 140, 1800. While this approach may achieve satisfactory performance for most applications, it may not achieve the best possible performance for computationally expensive pipelined processing since it may not, under certain circumstances, fully leverage a particular node's (or tier's) available assets. For example, since the executable code (in compute array 142 and processing system 148, for instance) must store and retrieve intermediate results from node memory 144, the streaming capabilities of the architecture may not be optimized to achieve maximum throughput, as reads from data store 143 must wait during node memory 144 access cycles. As noted above, in many processing scenarios, even in a Big Data context involving high performance computing and massive datasets, the architecture illustrated in FIGS. 1 and 4 may be adequate to provide sufficient performance and data throughput rates. For applications in which performance and data throughput are critical, however, it may be desirable to combine a Flash/Compute node 140, 1800 with a Compute node 1900, for example, such as in the architecture illustrated in FIG. 21.

[00214] As noted above with reference to FIG. 19, some embodiments of a Compute node 1900 may omit data store 143. In addition to economizing on silicon real estate, power

consumption, manufacturing costs, and the like, utilization of node 1900 may enable or facilitate implementation of an overall system architecture that resembles that depicted in FIG. 21. Such a system architecture or hardware arrangement may link a single, particular Flash/Compute node 1800 (which does incorporate or have access to memory store 143) to a pipeline of serially coupled Compute nodes 1900, each of which may be uniquely associated with that particular Flash/Compute node 1800. In the FIG. 21 example, tier 1 comprises node 1800 (*i.e.*, the "initiating node"), and every successive tier (2, 3, … t) comprises a node 1900. The illustrated architecture, where appropriately coded and leveraged to increase or to maximize data throughput, may have particular utility in connection with neural network processing operations.

[00215] In the context of processing the TensorFlow inference algorithm in connection with this hardware architecture, for example, for all nodes in a particular pipeline, the same bit stream implementing the algorithm may be loaded into the programmable logic 410 space of each node 1900, in addition to the programmable logic 410 space of Flash/Compute node 140, 1800. In some implementations, there may be minor differences in the data supplied to node 1800 as compared to those supplied to nodes 1900 in a particular pipeline. For instance, Flash/Compute node 140, 1800 may be programmed not to use its node memory 144 in connection with executing the algorithm, whereas each Compute node 1900 may be instructed or coded to do so before passing or transmitting its output to the next Compute node 1900 in the pipeline. These and any other desired operational parameters may be communicated to nodes 1800, 1900 via the bit stream.

[00216] In operation, the tier 1 initiating Flash/Compute node 1800 may retrieve input for the algorithm processing operation (*e.g.*, from memory store 143), execute required or desired computations for the first layer in the neural network, and transmit computational results or output data to the first Compute node 1900 in the pipeline. Once such computational results or output data are received from the initiating tier 1 node, the first Compute node 1900 in the pipeline (in this case, at tier 2), may begin processing operations at the next layer in the neural network; meanwhile, back at the tier 1 Flash/Compute node 1800, the next element of data to be used as input for the next algorithm processing operation may be retrieved (*e.g.*, from memory store 143) such that Flash/Compute node 1800 may already be executing the next subsequent required or desired computations or processing operations. Data processing responsibilities may propagate downstream through the pipeline, along with output from the previous tier, substantially as illustrated and described above with reference to FIG. 21.

[00217] Each Compute node 1900 in a particular pipeline may handle one or more layers of processing in the given algorithm. Eventually, upon completion of processing at a particular Compute node 1900 in a particular tier of the pipeline, that particular Compute node 1900 may then pass output to the next Compute node 1900 in the next subsequent tier. Those of skill in the art will appreciate that the number of layers for a particular algorithm that may be computed by a given Compute node 1900 in a particular pipeline may depend upon the coding provided in the bit stream, the complexity and structure of the neural network (including the overall architecture, pipeline length, specific implementations of nodes 1800, 1900, bandwidth of the communications links 149, read/write performance of node memory 144 or memory store 143, and the like), sophistication of the algorithm being processed, or a combination of these and other factors. In that regard, the number of Compute nodes 1900 employed in a given pipeline may also depend upon the complexity and structure of the neural network in the context of some or all of the factors listed above. In some instances, it may be useful to balance the ratio of layers (of algorithm processing) per Compute node 1900 with the number of Compute nodes 1900 in a given architecture in such a manner that the associated Flash/Compute node 1800 in a given pipeline may continue to accept data streamed from memory store 143 without needing to stall the read process.

[00218] As noted above, the terminal Compute node 1900 (*e.g.*, at tier t in FIG. 21) in a particular pipeline may transmit its data to a management node 2000; in some implementations, such a management node 2000 may also be supplied with a copy of the algorithm bit stream that instantiated the processing operations in the first place. In operation, management node 2000 generally may execute the final layer (or final several layers) of computation required by the algorithm to produce a result. In some implementations, it may be desirable that a management node 2000 comprises four independent channels to router memory 124; this architecture may enable a four-way parallel computation, for example, such that a given management node 2000 may support up to four pipelines. This is illustrated in FIG. 21, which has been simplified to show only a single set of slaved, serialized Compute nodes 1900 per Flash/Compute node 1800. Other embodiments comprising multiple pipelines per Flash/Compute node 1800 are contemplated as set forth above, and may be implemented as a design choice or as a function of overall system throughput requirements.

[00219] In operation, management node 2000 may buffer computational results or algorithm output for return to host compute system 199 when appropriate or required. When resources at

management node 2000 are taxed or reach a capacity threshold that may be pre-determined or dynamically computed or adjusted (*e.g.*, when result buffers 2091 are full or exceed a threshold or other pre-determined or dynamically computed limit), management node 2000 may signal the algorithm initiating nodes 140, 1800 under its control that computed results are ready for transmission. In some embodiments, one or more applicable Flash/Compute nodes 140, 1800 may initiate the return, *e.g.*, by commanding or instructing management node 2000 to transmit buffered results to host compute system 199; alternatively, management node 2000 may initiate such a data transfer in accordance with coded instructions, *e.g.*, unbidden or otherwise without instructions or requests supplied by any other system component.

[00220] The following description addresses certain hardware and functional details of some of the system components and implementations described above. In that regard, recall that FIGS. 2A through 2C are functional block diagrams illustrating one example of a hardware environment accommodating components of a data processing and analytics system. These hardware environments may be referred to as "cards" in some instances, such as in the description of FIGS. 2A through 2C; additionally, those of skill in the art will appreciate that such cards may also be referred to as "modules" in some industries or for some applications. In the description that follows, the term "module" may be used as a synonym for "card" as that term is used in reference to FIGS. 2A through 2C.

Program Logic 410 Space

[00221] In some implementations, a single implementation of a card or module may generally comprise two types of FPGAs or processing components.

[00222] The first may be embodied in or comprise a straightforward FPGA or other processing component that handles all the functionality of a management node 2000 (FIG. 20) or router module 120 (FIG. 1) set forth above. In the illustrated implementations, there is only one such processing component per module. To accommodate the described functionality, this processing component (for example, algorithm framework 2021 in router 122 portion) may comprise or incorporate a large (in terms of overall silicon real estate or area) programmable logic space (such as APL 2022, for instance) and a large number of I/O lanes. In operation, this FPGA or processing component may generally implement appropriate protocols for messaging to/from the host compute system 199, vector messaging to/from pipelined nodes 140, 1800, and 1900, and other routing and housekeeping functionality substantially as described above.

[00223] The second is an FPGA or other processing component that may generally be implemented as a Multi-Processor System on a Chip (MPSoC). In the illustrated embodiments, every module comprises at least one of these; in some embodiments such as described above, a module or card may comprise eight of such processing components, which may operate in parallel (though other configurations are also possible, and may reflect design choices, cost constraints, desired processing characteristics, and the like). In embodiments, each such processing component includes a reasonably large programmable logic 410 space, four or more (*e.g.*, 64 bit) processors or cores 439, two or more (*e.g.*, 32 bit) processors or cores 432, and a sufficient number of I/O lanes to enable both communications with a management node 2000 via communications links 149 as well as high speed connections to node memory 144 and memory store 143 elements, if applicable. Note that some nodes 1900 may also employ the same or similar processing components, but do not need to implement a connection to a memory store 143 component.

[00224] Within each type of FPGA or processing component, an algorithm's use of the available resources while residing in the programmable logic 410 space may be managed by the algorithm framework (142 in FIGS. 18 and 19 or 2021 in FIG. 20). In accordance with some methodologies, other than restrictions imposed by the algorithm framework 142, 2021, an algorithm may be configured and operative to use any or all available logic within the programmable logic space.

[00225] In operation, local firmware (*e.g.*, executing on one of more of the embedded processors 439, 1871) may initialize the programmable logic 410 space upon system boot or as otherwise requested, and may then load a "golden image" or template containing the appropriate or requested algorithm framework 142, 2021. Once loaded and initialized as is generally known, the local firmware may then load and execute algorithms or attendant or associated instruction sets as dictated by messages received from the host compute system 199.

Algorithm Framework 142, 2021

[00226] The algorithm framework 142, 2021 may generally provide two basic categories of services: system utilities; and a data selection engine.

[00227] In some implementations, system utilities may generally include the following: communications link 149 messaging system protocols and support; control of access (*i.e.*, driver support, controller functionality, *etc.*) to node memory 144 and memory store 143 elements, if

applicable; data movers (described above) that may queue and initiate read/write activities for node memory 144 and memory store 143 elements, if applicable; start/stop/resume control for algorithm execution processes; and management of any unhandled condition (such as interrupts and the like) arising from algorithm processing operations.

[00228] In some implementations, data selection engine services may be requested or utilized in cooperation with the data movers, for instance, to implement a flexible selection capability when an algorithm operates on only a subset of data from a target dataset. For example, selections of targeted portions or subsets of data may be made using any of a variety of fields in a dataset, irrespective of whether those particular data are from the fields on which a particular algorithm will ultimately be operating. In such cases, selection of specific subsets of data may be based upon specific fields in the target dataset, fields in a different dataset that includes indices to the target dataset, both, or a combination of these and other factors or system parameters related to or arising from the applicable algorithm and processing boundary conditions. It is noted that co-location of data or datasets, as well as providing an option for a particular algorithm to execute on multiple datasets, complement these features and may result in increased computational and data access efficiencies in may applications.

[00229] In some embodiments, it may be desirable that the only permitted interaction between algorithms and the algorithm framework facilities of a node 1800, 1900 may be via the algorithm framework 142, 2021 hardware and firmware implementation. In accordance with the disclosed subject matter, this restricted permission paradigm may be a deliberate design choice, as the algorithm framework 142, 2021 may be responsible for preventing algorithms and the constituent hardware and firmware components from corrupting system data and user data maintained in node memory 144 and memory store 143, if applicable. Other alternative embodiments may not be so restricted, however, as will be appreciated by those of skill in the art.

Node Memory 144

[00230] The majority of node memory 144 located on or accessible by a Flash/Compute node 142, 1800 may be dedicated to or otherwise used for managing access to memory store 143, as well as for storing system information required by local firmware that manages operation and overall functionality of the node 142, 1800. Even in light of the foregoing, however, several gigabytes (or more, in some implementations) of node memory 144 may be available for use by Level 2 algorithms, if needed (this is represented by reference numeral 1841 in FIG. 18). Access

to such algorithm memory 1841 may be achieved via the PAF or via interaction with one or more cores 439 or 1871.

[00231] In some embodiments of a Compute node 1900, a minority or relatively small portion of local node memory 144 (typically, only a few gigabytes) may be used for system information and intra-node messaging operations required by local firmware. The remainder of the available capacity (typically, 16 gigabytes or more) of a node memory 144 storage component (*i.e.*, algorithm memory 1841) may thus be made available for Level 2 algorithm execution operations, for example: buffering intermediate results for use within the node 1900 itself; buffering a final intermediate result for transmission to the next node 1900 in the pipeline or to the management node 2000; storing data relevant to a previous or subsequent neural network processing layer; and the like.

[00232] In typical Flash/Compute node 1800 as set forth herein, an algorithm using local node memory 144 generally may slow down the streaming of data from memory store 143 into the programmable logic 410 space as required during algorithm processing operations. Such an intentional governing of data flow may be implemented as a design choice, since the data movers (described above) represent shared resources for both functions (*i.e.*, retrieving data from memory store 143 and managing intra-node data transfers with respect to node memory 144 and other components of node 1800). In some situations, this may affect throughput in a manner that is measureable or significant, which is one of the motivations to implement the architecture depicted in FIG. 21 for high-throughput, computationally expensive applications in which it may be useful to regulate or to limit access to memory store 143.

Processing Cores

[00233] Local firmware running on either a Flash/Compute node 140, 1800, on the one hand, or on Compute node 1900, on the other hand, may generally utilize one or more processors 439, 1871 or cores (see processing system 430 in FIG. 4). As is generally known in the art, such firmware may employ multiple cores, for instance, to increase performance under normal loads, or it may throttle back and dedicate fewer cores to intra-node work in the event that a Level 2 algorithm requires the use of one or more processors 439, 1871 for high load situations.

[00234] In some implementations, it may be desirable to run a host Linux operating system in connection with any processor 432, 439, 1871 that is being used by, or that is managing, an algorithm processing operation or any application-specific code. Access to the various facilities

of an applicable node 142, 1800, 1900, and 2000 may be strictly controlled by such a host operating system, and Linux is one operating system that provides certain protections for user and system data that are located in node memory 144 and memory store 143 that are not available in other commercially available operating systems. Specifically, using Linux, direct access to node memory 144 and memory store 143 elements may be strictly regulated in a manner that is not readily implemented using other operating systems. In some circumstance, the EMMC memory component 1870 integrated into or accessible by a Flash/Compute node 1800 and a Compute node 1900 may be utilized for maintaining data and instruction sets associated with the host operating system, for instance, and to act as a "hard drive" for the local node 1800, 1900 when needed or desired. Such activity may be limited to a specific partition on the memory component 1870 if desired, as is generally known in the art.

Management Node 2000 Facilities

[00235] A management node 2000, as previous described, may have a distinct architecture and specific facilities that are available to accommodate processing of Level 2 algorithms in some situations. In some implementations, for instance, a management node 2000 may include or have access to up to four times (or more) the local memory capacity (*e.g.*, router memory 124) of other nodes (such as nodes 140, 1800, and 1900) and no memory store 143 component; additionally or alternatively, it may also not have any local processors or processing cores.

[00236] In the illustrated embodiments (provided by way of example only) a management node 2000 may generally be controlled by its own local version of the algorithm framework 2021, which may include extensions or appropriate modifications suitable to enable bi-directional data communication with the host compute system 199 (as set forth above with reference to FIG. 1) and any necessary functional blocks to implement necessary PCIe or other communications protocols (or other applicable protocols for interfacing with any other communication fabric that is generally known in the art or developed in accordance with known principles).

[00237] In the context of algorithm processing operations utilizing the programmable logic 2022 space of a management node 2000, for example, a single FPGA or other processing component resident at or integrated into the management node 2000 may be made available via algorithm framework 2021, and may be loaded and made available in the event that a Level 2 algorithm, which may require use of its resources, is selected for execution.

[00238] In the illustrated arrangement, router memory 124 integrated into a management node 2000 may be implemented as four or more separate DRAM modules (reference numerals 124a through 124n in FIGS. 1 and 20), though other configurations may have utility in some situations. In one implementation, each router memory 124 module may be sixteen gigabytes in capacity, and may be coupled to the local algorithm framework 2021 via a dedicated or individual direct memory access (DMA) or other channel. In that regard, a particular management node 2000 may alternatively function effectively as four (or more) virtual Compute nodes 1900 that each have programmable logic 2022 space only. In addition, the memory usage requirements for a given management node 2000 (in terms of overall system operation) may be generally low, so the vast majority of the total memory capacity resident at a management node 2000 may be made available to an algorithm executing in its programmable logic 2022 space.

[00239] A management node 2000 may also be capable of communicating directly with any and all other nodes 140, 1800, and 1900 connected to it via a pipeline or other distributed architecture hardware construct. These communications facilities may generally be used at any point during algorithm execution, though it is most likely that inter-node communications involving a management node 2000 will be related to or associated with a final or terminal operation at the end of execution or when data buffers or other temporary data storage components (such as result buffers 2091) reach some pre-determined capacity, at which point it may be necessary or desirable to initiate a return of data to the host compute system 199.

[00240] Several features and aspects of a system and method have been illustrated and described in detail with reference to particular embodiments by way of example only, and not by way of limitation. Those of skill in the art will appreciate that alternative implementations and various modifications to the disclosed subject matter are within the scope and contemplation of the present disclosure. Therefore, it is intended that the present disclosure be considered as limited only by the scope of the appended claims.

What is claimed is:

CLAIMS

1. A data processing system operative in cooperation with a host compute system, the data processing system comprising:

a router module comprising a host interface and a node interface, wherein the router module is communicatively coupled to the host compute system via the host interface; and

multiple parallel first tier compute nodes coupled to one or more additional compute nodes in series, each compute node comprising:

a communications link, wherein each first tier compute node is communicatively coupled to the node interface at the router module via the communications link, at different successive tiers, each compute node in a series being communicatively coupled, via its communications link to one or more additional compute nodes connected at different successive tiers;

a data store comprising records associated with a data processing operation;

a programmable logic component to execute the data processing operation in cooperation with the data store;

a node memory comprising data and instructions to support operation of the programmable logic component in connection with the data processing operation and to facilitate data communications via the communications link;

a data mover component to facilitate data communications between the programmable logic component and the node memory; and

a storage interface component to facilitate data communications between the programmable logic component and the data store;

wherein the storage interface component utilizes a plurality of communications channels to transfer data between the programmable logic component and the data store.

2. The data processing system of claim 1 wherein the communications link employs a serial communications protocol.

3. The data processing system of claim 1 wherein the data store comprises an Electronically Erasable Programmable Read Only Memory device.
4. The data processing system of claim 3 wherein the storage interface component employs and Open NAND Flash Interface protocol.
5. The data processing system of claim 1 wherein the programmable logic component comprises a field programmable gate array.
6. The data processing system of claim 1 further comprising a processor to manage operation of the data mover component.
7. The data processing system of claim 6 wherein the data mover component comprises a device controller.
8. The data processing system of claim 1 wherein the host compute system executes instructions to influence operation of the compute node.
9. The data processing system of claim 1 wherein the router module and the compute node are integrated into an interface card communicatively coupled to the host compute system.

SUBSTITUTE SPECIFICATION
Clean Version

ABSTRACT

A system and method employing a distributed hardware architecture, either independently or in cooperation with an attendant data structure, in connection with various data processing strategies and data analytics implementations are disclosed. A compute node may be implemented independent of a host compute system to manage and to execute data processing operations. Additionally, an unique algorithm architecture and processing system and method are also disclosed. Different types of nodes may be implemented, either independently or in cooperation with an attendant data structure, in connection with various data processing strategies and data analytics implementations.

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
18/075,576	12/06/2022	2181	830	FERM0003CON200564319	9	1

CONFIRMATION NO. 1074

UPDATED FILING RECEIPT

27799
COZEN O"CONNOR
3WTC, 175 Greenwich Street
55th Floor
NEW YORK, NY 10007

OC000000052400692

Date Mailed: 01/09/2023

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)

Robert Bismuth, Seattle, WA;
Mike Stengle, Seattle, WA;

Applicant(s)

Fermat International, Inc., Paris, FRANCE;

Power of Attorney: None

Domestic Priority data as claimed by applicant

This application is a CON of 17/666,049 02/07/2022 PAT 11,550,512
which is a CON of 16/729,727 12/30/2019 PAT 11,249,690
which claims benefit of 62/882,044 08/02/2019
and claims benefit of 62/853,764 05/29/2019
and claims benefit of 62/801,839 02/06/2019

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: No

Permission to Access Search Results: No

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: 12/19/2022

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 18/075,576**

Projected Publication Date: 04/20/2023

Non-Publication Request: No

Early Publication Request: No
**** SMALL ENTITY ****
Title

Analytics, Algorithm Architecture, and Data Processing System and Method

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: 18/075,576

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	80		N/A	
SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A	N/A	350		N/A	
EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A	N/A	400		N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	9 minus 20 =	* 0	x =	0	OR		
INDEPENDENT CLAIMS (37 CFR 1.16(h))	1 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	830		TOTAL	

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	RATE($)	ADDITIONAL FEE($)	OR	RATE($)	ADDITIONAL FEE($)
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.

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

PTO/AIA/82A (07-13)
Approved for use through 03/31/2021. OMB 0651-0035
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.

TRANSMITTAL FOR POWER OF ATTORNEY TO ONE OR MORE REGISTERED PRACTITIONERS

NOTE: This form is to be submitted with the Power of Attorney by Applicant form (PTO/AIA/82B) to identify the application to which the Power of Attorney is directed, in accordance with 37 CFR 1.5, unless the application number and filing date are identified in the Power of Attorney by Applicant form. If neither form PTO/AIA/82A nor form PTO/AIA82B identifies the application to which the Power of Attorney is directed, the Power of Attorney will not be recognized in the application.

Application Number	18/075,576
Filing Date	12-06-2022
First Named Inventor	Robert Bismuth
Title	Analytics, Algorithm Architecture, and Data Processing System and Method
Art Unit	OPAP
Examiner Name	CENTRAL, DOCKET
Attorney Docket Number	FERM0003CON200564319

SIGNATURE of Applicant or Patent Practitioner

Signature	/Victor Castellucci/	Date (Optional)	02/17/2023
Name	Victor Castellucci	Registration Number	43535
Title (if Applicant is a juristic entity)			
Applicant Name (if Applicant is a juristic entity)			

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

☑ *Total of 2 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.

DocuSign Envelope ID: BBBC912A-2941-497B-A604-A4A91477C721

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: 27799

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):

Fermat International, Inc.

☐ 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		Robert Bismuth	Date (Optional)	7/15/2021
Name	Robert Bismuth			
Title				

Docusigned by: Robert Bismuth E6DB403AF2F349E...

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.

Electronic Acknowledgement Receipt

EFS ID:	47544621
Application Number:	18075576
International Application Number:	
Confirmation Number:	1074
Title of Invention:	Analytics, Algorithm Architecture, and Data Processing System and Method
First Named Inventor/Applicant Name:	Robert Bismuth
Customer Number:	27799
Filer:	Victor Jay Castellucci/Maria Lima
Filer Authorized By:	Victor Jay Castellucci
Attorney Docket Number:	FERM0003CON200564319
Receipt Date:	17-FEB-2023
Filing Date:	06-DEC-2022
Time Stamp:	16:10:56
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Power of Attorney	Poa_Fermat.pdf	272946 03451bc2295e91216c1413c3c5536ae7888e9c04	no	2

Warnings:

Information:	
Total Files Size (in bytes):	272946

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 a 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.

PTO/SB/08a (02-18)
Approved for use through 11/30/2020. 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	18075576
Filing Date	2022-12-06
First Named Inventor	Robert Bismuth
Art Unit	TBD
Examiner Name	TBD
Attorney Docket Number	FERM0003CON200564319

U.S.PATENTS

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.

U.S.PATENT APPLICATION PUBLICATIONS

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	20160196077		2016-07-07	Hitachi LTD	
	2	20040268088		2004-12-30	Lippincott Louis A.	

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

FOREIGN PATENT DOCUMENTS

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

NON-PATENT LITERATURE DOCUMENTS

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	18075576
	Filing Date	2022-12-06
	First Named Inventor	Robert Bismuth
	Art Unit	TBD
	Examiner Name	TBD
	Attorney Docket Number	FERM0003CON200564319

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	EP Extended Search Report	☐

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

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	18075576
	Filing Date	2022-12-06
	First Named Inventor	Robert Bismuth
	Art Unit	TBD
	Examiner Name	TBD
	Attorney Docket Number	FERM0003CON200564319

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 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.

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

☒ A certification statement is not submitted herewith.

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	/Victor Castellucci/	Date (YYYY-MM-DD)	2023-02-17
Name/Print	Victor Castellucci	Registration Number	43535

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.

Electronic Acknowledgement Receipt

EFS ID:	47554182
Application Number:	18075576
International Application Number:	
Confirmation Number:	1074
Title of Invention:	Analytics, Algorithm Architecture, and Data Processing System and Method
First Named Inventor/Applicant Name:	Robert Bismuth
Customer Number:	27799
Filer:	Victor Jay Castellucci/Maria Lima
Filer Authorized By:	Victor Jay Castellucci
Attorney Docket Number:	FERM0003CON200564319
Receipt Date:	17-FEB-2023
Filing Date:	06-DEC-2022
Time Stamp:	18:21:49
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	no

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Transmittal Letter	FERM0003CON2_00564319_IDS_Transmittal_FINAL.pdf	32361 832cac8f6331294ac659a34c5c93249e58459b6b	no	3

Warnings:

Information:					
2	Information Disclosure Statement (IDS) Form (SB08)	FERM0003CON200564319_IDS_FINAL.pdf	63619 d8235c43dc70022da9bb90b1aedc930deab572a3	no	4
Warnings:					
Information:					
This is not an USPTO supplied IDS fillable form					
3	Non Patent Literature	Extended_search_report_Ferm at.pdf	401304 b55d524c593db2bae551f5210a74a524d39e8ec6	no	12
Warnings:					
Information:					
			Total Files Size (in bytes):	497284	

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 a 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.

DOCKET NO.: FERM0003CON2/00564319 **PATENT**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of: Robert Bismuth **Confirmation No.: 1074**

Serial No.: 18/075,576 **Group Art Unit: OPAP**

Filing Date: 12-06-2022 **Docket No.: FERM0003CON2/00564319**

For: Analytics, Algorithm Architecture, and Data Processing System and Method

Customer No.: 27799

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

Dear Commissioner:

INFORMATION DISCLOSURE STATEMENT

Pursuant to 37 C.F.R. §§ 1.56 and in accordance with 37 C.F.R. §§ 1.97 and 1.98, information relating to the above-identified application is hereby disclosed. The Examiner in charge of the above-identified application is requested to consider and make of record the references listed on PTO Forms SB/08A and SB/08B, formerly known as PTO Form 1449, submitted herewith.

Inclusion of the information submitted herewith is not to be construed as an admission that the information is material as that term is defined in 37 C.F.R. § 1.56(b).

In accordance with 37 C.F.R. § 1.97(g), the filing of this Information Disclosure Statement shall not be construed to mean that a search has been made.

This Information Disclosure Statement is being filed:

☐ within three months of the filing date of the patent application.

☐ within three months of the date of entry into the national stage as set forth in 37 C.F.R. § 1.491 of the international application.

☒ **before** the mailing date of a first Office Action on the merits.

☐ **before** the mailing date of a first Office Action on the merits after the filing of a Request for Continued Examination under § 1.114..

☐ **after** the mailing date of a first Office Action on the merits or after the filing of a Request for Continued Examination under § 1.114, but before the mailing date of a Final Office Action under 37 C.F.R. § 1.116 or a Notice of Allowance under 37 C.F.R. § 1.311, and accordingly is accompanied by:

☐ the Statement under 37 C.F.R. § 1.97(e) (see "Statement" below); or

☐ the Fee under 37 C.F.R. § 1.17(p) (see below); or

☒ No fee is owed by the applicant(s).

☐ In accordance with 37 C.F.R. § 1.129(a), this Information Disclosure Statement is being filed in connection with ☐ the first or ☐ second After Final Submission, and accordingly is accompanied by the Statement under 37 C.F.R. § 1.97(e) (see "Statement" below) and the Fee under 37 C.F.R. § 1.17(p) (see below).

☐ **after** the mailing date of a Final Office Action under 37 C.F.R. § 1.116 or a Notice of Allowance under 37 C.F.R. § 1.311, but before, or simultaneously with, the payment of the Issue Fee, and accordingly is accompanied by the Statement under 37 C.F.R. § 1.97(e) (see below) and the Fee under 37 C.F.R. § 1.17(p) (see below).

☒ Copies of the foreign patent documents and non-patent literature references listed on the attached PTO Forms SB/08a, formerly known as PTO Form 1449, are enclosed herewith.

EXCEPT THAT:

☐ In view of the voluminous nature of the references and the likelihood that these references are available to the Examiner, copies of references _____ are not enclosed.

☐ In accordance with 37 C.F.R. § 1.98(d), copies of the following references listed on the attached PTO Form SB/08A and PTO Form SB/08B, formerly known as PTO Form 1449, are not enclosed herewith because they were previously cited by or submitted to the U.S. Patent and Trademark Office in patent application(s) for which a claim for priority under 35 U.S.C. § 120 has been made in the instant application.

☐ Copies of the references listed on the attached PTO Forms SB/08A and SB/08B were previously cited by or submitted to the U.S. Patent and Trademark Office in parent application Serial No. _____, filed _____.

☐ If any of the foregoing publication is not available to the Examiner, Applicant will endeavor to supply a copy at the Examiner's request.

Statement under 37 C.F.R. § 1.97(e)

☐ The undersigned attorney hereby states that each item of information contained in the Information Disclosure Statement was cited in a communication from a foreign patent office in a counterpart application not more than three months prior to the filing of the Information Disclosure Statement.

Statement under 37 C.F.R. § 1.704(d)

☐ The undersigned attorney hereby states that each item of information contained in the Information Disclosure Statement was cited in a communication from a foreign patent office in a counterpart application and the communication was not received by any individual designated in § 1.56(c) more than 30 days prior to the filing of the Information Disclosure Statement.

Fees

☒ No Fee is owed by the applicant(s).

☐ The Information Disclosure Statement Fee of under 37 C.F.R. § 1.17(p) is authorized to be charged to Deposit Account No. 50-3111.

Method of Payment of Fees

☐ Attached is a check in the amount of $_____.

☐ Charge Deposit Account No. 50-3111 in the amount of $90.00.

☐ via EFS Web

☒ Please charge any deficiency or credit any overpayment to Deposit Account 50-3111.

☐ No fee or Statement is required under 37 C.F.R. § 1.97(b).

☐ There are no listed references, which are not in the English language.

Respectfully submitted,

/Victor J. Castellucci /
Victor J. Castellucci
Registration No. **43535**
COZEN O'CONNOR
1200 19th 8th Floor
Washington, D.C. 20036
Tel: (305) 358-5001
Fax: (305) 358-3309

Dated: Feb 17, 2023

Enclosures

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
18/075,576	12/06/2022	Robert Bismuth	FERM0003CON200564319

CONFIRMATION NO. 1074

27799
COZEN O"CONNOR
3WTC, 175 Greenwich Street
55th Floor
NEW YORK, NY 10007

POA ACCEPTANCE LETTER

OC000000054033281

Date Mailed: 02/23/2023

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 02/17/2023.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at **(571) 272-4000** or **(571) 272-4200** or **1-888-786-0101**.

/sstephanos/

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
18/075,576	12/06/2022	Robert Bismuth	FERM0003CON200564319

CONFIRMATION NO. 1074

27799
COZEN O"CONNOR
3WTC, 175 Greenwich Street
55th Floor
NEW YORK, NY 10007

PUBLICATION NOTICE



Title: Analytics, Algorithm Architecture, and Data Processing System and Method

Publication No.US-2023-0107344-A1
Publication Date:04/06/2023

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 accessed through the USPTO's publically available Searchable Databases via the Internet at www.uspto.gov. The direct link to access the publication is currently http://www.uspto.gov/patft/.

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 the Patent Electronic Business Center at www.uspto.gov using the public side of the Patent Application Information and Retrieval (PAIR) system. The direct link to access this status information is currently https://portal.uspto.gov/pair/PublicPair. Prior to publication, such status information is confidential and may only be obtained by applicant using the private side of PAIR.

Further assistance in electronically accessing the publication, or about PAIR, 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

Notice of References Cited	Application/Control No. 18/075,576	Applicant(s)/Patent Under Reexamination Bismuth et al.	
	Examiner CHRISTOPHER B SHIN	Art Unit 2181	Page 1 of 1

U.S. PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	CPC Classification	US Classification
*	A	US-20230107344-A1	04-2023	Bismuth; Robert	G06F9/3867	711/103
	B					
	C					
	D					
	E					
	F					
	G					
	H					
	I					
	J					
	K					
	L					
	M					

FOREIGN PATENT DOCUMENTS

*		Document Number Country Code-Number-Kind Code	Date MM-YYYY	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	
	V	
	W	
	X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	18/075,576	Bismuth et al.
	Examiner	Art Unit
	CHRISTOPHER B SHIN	2181

CPC - Searched*		
Symbol	**Date**	**Examiner**
(G06F3/0661 G06F3/0613 G06F3/067 G06F9/3867).CPC.	04/07/2023	CS

CPC Combination Sets - Searched*		
Symbol	**Date**	**Examiner**

US Classification - Searched*			
Class	**Subclass**	**Date**	**Examiner**
711	103	04/07/2023	CS

* 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**
PDP Searched	04/07/2023	CS
PARENT CASE SEARCHED	04/07/2023	CS
PE2E See the Record	04/07/2023	CS

Interference Search			
US Class/CPC Symbol	**US Subclass/CPC Group**	**Date**	**Examiner**
711 / G06F	103 / (3/0661; 3/0613; 3/067; 9/3867)		

/CHRISTOPHER B SHIN/ Primary Examiner, Art Unit 2181	

Bibliographic Data

Application No: 18/075,576

Foreign Priority claimed: ○ Yes ◉ No

35 USC 119 (a-d) conditions met: ☐ Yes ☑ No ☐ Met After Allowance

Verified and Acknowledged: /CHRISTOPHER B SHIN/ Examiner's Signature | Initials

Title: Analytics, Algorithm Architecture, and Data Processing System and Method

FILING or 371(c) DATE	CLASS	GROUP ART UNIT	ATTORNEY DOCKET NO.
12/06/2022	711	2181	FERM0003CON200564319
RULE			

APPLICANTS

Fermat International, Inc., Paris, FRANCE

INVENTORS

Robert Bismuth, Seattle, WA, UNITED STATES

Mike Stengle, Seattle, WA, UNITED STATES

CONTINUING DATA

This application is a CON of 17666049 02/07/2022 PAT 11550512

17666049 is a CON of 16729727 12/30/2019 PAT 11249690

16729727 has PRO of 62882044 08/02/2019

16729727 has PRO of 62853764 05/29/2019

16729727 has PRO of 62801839 02/06/2019

FOREIGN APPLICATIONS

IF REQUIRED, FOREIGN LICENSE GRANTED**

12/19/2022

** SMALL ENTITY **

STATE OR COUNTRY

UNITED STATES

ADDRESS

COZEN O"CONNOR

3WTC, 175 Greenwich Street

55th Floor

NEW YORK, NY 10007

UNITED STATES

FILING FEE RECEIVED

$830

PE2E SEARCH - Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
L3	10	((("BISMUTH") near3 ("Robert")) OR (("STENGLE") near3 ("Mike"))).INV.	(US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT)	OR	OFF	OFF	2021/06/01 04:23 PM
L4	2	((("FERMAT") near3 ("INTERNATIONAL") near3 ("INC"))).AS,AANM.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/06/01 05:14 PM
L6	235	tier with node with memory	(USPAT)	OR	OFF	OFF	2021/06/01 05:20 PM
L7	10626	compute near6 node	(USPAT)	OR	OFF	OFF	2021/06/01 06:02 PM
L8	235	router with L7	(USPAT)	OR	OFF	OFF	2021/06/01 06:03 PM
L9	16	host with L8	(USPAT)	OR	OFF	OFF	2021/06/01 06:03 PM
L10	56	L7 with tier	(USPAT)	OR	OFF	OFF	2021/06/01 06:11 PM
L11	10626	compute near6 node	(USPAT)	OR	OFF	OFF	2021/06/02 04:41 PM
L12	96	L11 with tier	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/06/02 04:41 PM
L13	56	L11 with tier	(USPAT)	OR	OFF	OFF	2021/06/02 04:41 PM
L14	40	L12 not L13	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/06/02 04:41 PM
L15	6	"20150301900" "20140289458"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/06/02 04:43 PM
L16	21	mover with compute with data with node	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/06/02 04:45 PM
L17	64448	(G06F3/0661 G06F3/0613 G06F3/067 G06F9/3867).CPC.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/06/02 04:56 PM
L18	488	L11 AND L17	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/06/02 04:57 PM
L19	0	mover with compute	(US-PGPUB; USPAT;	OR	ON	ON	2021/06/02

		with PIPELINE with node	USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)				07:27 PM
L20	324	compute with PIPELINE with node	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/06/02 07:28 PM
L21	2012	COMMUNICATION WITH NODE WITH PROGRAM$4 WITH LINK	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/06/02 08:57 PM
L22	935	DATA WITH L21	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/06/02 08:59 PM
L23	64448	(G06F3/0661 G06F3/0613 G06F3/067 G06F9/3867).CPC.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/06/02 08:59 PM
L24	23	L23 AND L22	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/06/02 08:59 PM
L25	3	"20150301900"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/10/29 12:31 PM
L26	11375	compute near6 node	(USPAT)	OR	OFF	OFF	2021/10/29 04:06 PM
L27	264	router with L26	(USPAT)	OR	OFF	OFF	2021/10/29 04:06 PM
L28	257	tier with node with memory	(USPAT)	OR	OFF	OFF	2021/10/29 04:06 PM
L29	264	router with L26	(USPAT)	OR	OFF	OFF	2021/10/29 04:06 PM
L30	17	host with L29	(USPAT)	OR	OFF	OFF	2021/10/29 04:06 PM
L31	2074	COMMUNICATION WITH NODE WITH PROGRAM$4 WITH LINK	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/10/29 04:07 PM
L32	965	DATA WITH L31	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/10/29 04:07 PM
L33	67651	(G06F3/0661 G06F3/0613 G06F3/067 G06F9/3867).CPC.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/10/29 04:07 PM
L34	367	compute with PIPELINE with node	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT;	OR	ON	ON	2021/10/29 04:07 PM

			IBM_TDB)				
L35	10	((("BISMUTH") near3 ("Robert")) OR (("STENGLE") near3 ("Mike"))).INV.	(US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT)	OR	OFF	OFF	2021/11/26 12:51 PM
L36	3	((("FERMAT") near3 ("INTERNATIONAL") near3 ("INC"))).AS,AANM.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/11/26 12:51 PM
L38	67994	(G06F3/0661 G06F3/0613 G06F3/067 G06F9/3867).CPC.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/11/26 12:52 PM
L39	369	compute with PIPELINE with node	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/11/26 12:52 PM
L40	309	(tier WITH node WITH memory)	(USPAT)	OR	OFF	OFF	2022/09/07 04:34 PM
L41	456	(compute WITH PIPELINE WITH node)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/09/07 04:34 PM
L42	955	(tier WITH node WITH memory)	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	OFF	OFF	2022/09/07 04:34 PM
L43	655	compute WITH PIPELINE WITH node	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/09/07 04:34 PM

L44	579	tier WITH node WITH memory	(US-PGPUB; USPAT)	OR	OFF	OFF	2022/09/07 04:35 PM
L45	27	mover WITH compute WITH data WITH node	(US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/09/07 04:35 PM
L46	784	router WITH (compute near6 node)	(US-PGPUB; USPAT)	OR	OFF	OFF	2022/09/07 04:35 PM
L47	784	router WITH (compute near6 node)	(US-PGPUB; USPAT; USOCR)	OR	OFF	OFF	2022/09/07 04:35 PM
L48	115	(compute near6 node) WITH tier	(US-PGPUB; USPAT)	OR	OFF	OFF	2022/09/07 04:36 PM
L50	579	tier WITH node WITH memory	(US-PGPUB; USPAT)	OR	OFF	OFF	2022/09/07 04:40 PM
L51	955	tier WITH node WITH memory	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	OFF	OFF	2022/09/07 04:41 PM
L52	3	"11249690"	(US-PGPUB; USPAT)	OR	ON	ON	2022/09/09 11:41 AM
L53	59217	(G06F3/0661 G06F3/0613 G06F3/067 G06F9/3867).CPC.	(US-PGPUB; USPAT; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO))	OR	ON	ON	2022/10/21 01:31 PM
L54	3	45 AND 46	(US-PGPUB; USPAT)	OR	ON	ON	2022/10/21 01:37 PM
L55	4	("20040268088" OR "20140289458" OR "20150301900" OR "20160196077").pn.	(US-PGPUB; USPAT)	OR	ON	ON	2022/11/03 06:46 PM
L56	6	"11550512" "11249690"	(US-PGPUB; USPAT)	OR	ON	ON	2023/04/07 01:12 PM

L57	1061	tier WITH node WITH memory	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	OFF	OFF	2023/04/07 03:25 PM
L58	642	tier WITH node WITH memory	(US-PGPUB; USPAT)	OR	OFF	OFF	2023/04/07 03:25 PM
L60	128	(compute near6 node) WITH tier	(US-PGPUB; USPAT)	OR	OFF	OFF	2023/04/07 03:25 PM
L61	83420	(G06F3/0661 G06F3/0613 G06F3/067 G06F9/3867).CPC.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/04/07 03:26 PM
L62	181	(compute near6 node) WITH tier	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	OFF	OFF	2023/04/07 03:26 PM
L63	1061	tier WITH node WITH memory	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	OFF	OFF	2023/04/07 03:26 PM

L64	1061	tier WITH node WITH memory	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	OFF	OFF	2023/04/07 03:27 PM
L65	521	(compute WITH PIPELINE WITH node)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/04/07 03:27 PM
L66	1061	(tier WITH node WITH memory)	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	OFF	OFF	2023/04/07 03:27 PM
L67	2	("20040268088" OR "20160196077").pn.	(US-PGPUB; USPAT)	OR	ON	ON	2023/04/08 01:54 PM

PE2E SEARCH - Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
N40	7	((("BISMUTH") near3 ("Robert")) OR (("STENGLE") near3 ("Mike"))).INV.	(US-PGPUB; USPAT)	OR	OFF	OFF	2021/11/26 12:53 PM
N41	1	((("FERMAT") near3 ("INTERNATIONAL") near3 ("INC"))).AS,AANM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/11/26 12:53 PM
N42	31911	(G06F3/0661 G06F3/0613 G06F3/067 G06F9/3867).CPC.	(US-PGPUB; USPAT)	OR	ON	ON	2021/11/26 12:53 PM

N43	353	compute with PIPELINE with node	(US-PGPUB; USPAT)	OR	ON	ON	2021/11/26 12:54 PM
N44	496	tier with node with memory	(US-PGPUB; USPAT)	OR	OFF	OFF	2021/11/26 12:54 PM
N45	34805	(G06F3/0661 G06F3/0613 G06F3/067 G06F9/3867).CPC.	(US-PGPUB; USPAT)	OR	ON	ON	2022/10/21 01:32 PM
N46	441	compute WITH PIPELINE WITH node$3	(US-PGPUB; USPAT)	OR	ON	ON	2022/10/21 01:32 PM
N47	746	tier WITH node$2 WITH memory	(US-PGPUB; USPAT)	OR	OFF	OFF	2022/10/21 01:32 PM
N48	139	45 AND 47	(US-PGPUB; USPAT)	OR	OFF	OFF	2022/10/21 01:37 PM
N49	24	45 AND 46	(US-PGPUB; USPAT)	OR	ON	ON	2022/10/21 01:37 PM

PTO/SB/08a (02-18)
Approved for use through 11/30/2020. 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	18075576
	Filing Date	2022-12-06
	First Named Inventor	Robert Bismuth
	Art Unit	TBD
	Examiner Name	TBD
	Attorney Docket Number	FERM0003CON200564319

U.S.PATENTS						
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.						

U.S.PATENT APPLICATION PUBLICATIONS						
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	20160196077		2016-07-07	Hitachi LTD	
	2	20040268088		2004-12-30	Lippincott Louis A.	
If you wish to add additional U.S. Published Application citation information please click the Add button.						

FOREIGN PATENT DOCUMENTS								
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								

NON-PATENT LITERATURE DOCUMENTS

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.B.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	18075576
	Filing Date	2022-12-06
	First Named Inventor	Robert Bismuth
	Art Unit	TBD
	Examiner Name	TBD
	Attorney Docket Number	FERM0003CON200564319

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	EP Extended Search Report	☐

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

EXAMINER SIGNATURE

Examiner Signature	/CHRISTOPHER B SHIN/	Date Considered	04/08/2023

*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.B.S/

INFORMATION DISCLOSURE STATEMENT BY APPLICANT (Not for submission under 37 CFR 1.99)		
	Application Number	18075576
	Filing Date	2022-12-06
	First Named Inventor	Robert Bismuth
	Art Unit	TBD
	Examiner Name	TBD
	Attorney Docket Number	FERM0003CON200564319

CERTIFICATION STATEMENT

Please see 37 CFR 1.97 and 1.98 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.

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

☒ A certification statement is not submitted herewith.

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	/Victor Castellucci/	Date (YYYY-MM-DD)	2023-02-17
Name/Print	Victor Castellucci	Registration Number	43535

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.

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.
18/075,576	12/06/2022	Robert Bismuth	FERM0003CON200564319	1074

27799 7590 04/13/2023

COZEN O"CONNOR
3WTC, 175 Greenwich Street
55th Floor
NEW YORK, NY 10007

EXAMINER	
SHIN, CHRISTOPHER B	

ART UNIT	PAPER NUMBER
2181	

NOTIFICATION DATE	DELIVERY MODE
04/13/2023	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):

patentdocket@cozen.com
patentsecretary@cozen.com
patentsorter@cozen.com

Office Action Summary	Application No. 18/075,576	Applicant(s) Bismuth et al.	
	Examiner CHRISTOPHER B SHIN	Art Unit 2181	AIA (FITF) Status Yes

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTHS FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☑ Responsive to communication(s) filed on 2/17/2023.

☐ A declaration(s)/affidavit(s) under **37 CFR 1.130(b)** was/were filed on _____.

2a)☐ This action is **FINAL.** 2b) ☑ This action is non-final.

3)☐ 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.

4)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims*

5) ☑ Claim(s) 1-9 is/are pending in the application.

5a) Of the above claim(s) _____ is/are withdrawn from consideration.

6) ☐ Claim(s) _____ is/are allowed.

7) ☑ Claim(s) 1-9 is/are rejected.

8) ☐ Claim(s) _____ is/are objected to.

9) ☐ Claim(s) _____ are subject to restriction and/or election requirement

* If any claims have been determined allowable, 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.**

Application Papers

10)☐ The specification is objected to by the Examiner.

11)☑ The drawing(s) filed on 12/6/2022 is/are: a)☑ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

Priority under 35 U.S.C. § 119

12)☐ 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)).

** See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☑ Notice of References Cited (PTO-892)

2) ☑ Information Disclosure Statement(s) (PTO/SB/08a and/or PTO/SB/08b) Paper No(s)/Mail Date 20230217.

3) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date _____.

4) ☐ Other: _____.

DETAILED ACTION

Notice of Pre-AIA or AIA Status

1. The present application, filed on or after March 16, 2013, is being examined under the first inventor to file provisions of the AIA.

2. Claims 1-9 are pending in the application.

Double Patenting

3. The nonstatutory double patenting rejection is based on a judicially created doctrine grounded in public policy (a policy reflected in the statute) so as to prevent the unjustified or improper timewise extension of the "right to exclude" granted by a patent and to prevent possible harassment by multiple assignees. A nonstatutory double patenting rejection is appropriate where the conflicting claims are not identical, but at least one examined application claim is not patentably distinct from the reference claim(s) because the examined application claim is either anticipated by, or would have been obvious over, the reference claim(s). See, e.g., *In re Berg*, 140 F.3d 1428, 46 USPQ2d 1226 (Fed. Cir. 1998); *In re Goodman*, 11 F.3d 1046, 29 USPQ2d 2010 (Fed. Cir. 1993); *In re Longi*, 759 F.2d 887, 225 USPQ 645 (Fed. Cir. 1985); *In re Van Ornum*, 686 F.2d 937, 214 USPQ 761 (CCPA 1982); *In re Vogel*, 422 F.2d 438, 164 USPQ 619 (CCPA 1970); *In re Thorington*, 418 F.2d 528, 163 USPQ 644 (CCPA 1969).

A timely filed terminal disclaimer in compliance with 37 CFR 1.321(c) or 1.321(d) may be used to overcome an actual or provisional rejection based on nonstatutory double patenting provided the reference application or patent either is shown to be

commonly owned with the examined application, or claims an invention made as a result of activities undertaken within the scope of a joint research agreement. See MPEP § 717.02 for applications subject to examination under the first inventor to file provisions of the AIA as explained in MPEP § 2159. See MPEP § 2146 *et seq.* for applications not subject to examination under the first inventor to file provisions of the AIA. A terminal disclaimer must be signed in compliance with 37 CFR 1.321(b).

The USPTO Internet website contains terminal disclaimer forms which may be used. Please visit www.uspto.gov/patent/patents-forms. The filing date of the application in which the form is filed determines what form (e.g., PTO/SB/25, PTO/SB/26, PTO/AIA/25, or PTO/AIA/26) should be used. A web-based eTerminal Disclaimer may be filled out completely online using web-screens. An eTerminal Disclaimer that meets all requirements is auto-processed and approved immediately upon submission. For more information about eTerminal Disclaimers, refer to www.uspto.gov/patents/process/file/efs/guidance/eTD-info-I.jsp.

4. Claims 1-9 are rejected on the ground of nonstatutory double patenting as being unpatentable over claims 1-9 of U.S. Patent No. 11,249,690. Although the claims at issue are not identical, they are not patentably distinct from each other because the combinations of the (690) patent claims 1-9 claims/teaches functionally equivalent system plus method claims of the present claims 1-9; therefore, the present claims are not patentably distinct form the (690) patent.

5. Claims 1-9 are rejected on the ground of nonstatutory double patenting as being unpatentable over claims 1-12 of U.S. Patent No. 11,550,512. Although the claims at issue are not identical, they are not patentably distinct from each other because the combinations of the (512) patent claims 1-12 claims/teaches functionally equivalent system plus method claims of the present claims 1-9; therefore, the present claims are not patentably distinct form the (512) patent.

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to CHRISTOPHER B SHIN whose telephone number is (571)272-4159. The examiner can normally be reached 8:00-4:00 PM.

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, IDRISS N ALROBAYE can be reached on 571-270-1023. 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.

/CHRISTOPHER B SHIN/
Primary Examiner, Art Unit 2181

PTO/SB/06 (09-11)
Approved for use through 1/31/2014. 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.

PATENT APPLICATION FEE DETERMINATION RECORD Substitute for Form PTO-875	Application or Docket Number 18/075,576	Filing Date 12/06/2022	☐ To be Mailed

ENTITY: ☐ LARGE ☑ SMALL ☐ MICRO

APPLICATION AS FILED - PART I

FOR	(Column 1) NUMBER FILED	(Column 2) NUMBER EXTRA		RATE ($)	FEE ($)
☐ BASIC FEE (37 CFR 1.16(a), (b), or (c))	N/A	N/A		N/A	
☐ SEARCH FEE (37 CFR 1.16(k), (i), or (m))	N/A	N/A		N/A	
☐ EXAMINATION FEE (37 CFR 1.16(o), (p), or (q))	N/A	N/A		N/A	
TOTAL CLAIMS (37 CFR 1.16(i))	minus 20 =	*		x $50 =	
INDEPENDENT CLAIMS (37 CFR 1.16(h))	minus 3 =	*		x $240 =	
☐ 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).				
☐ MULTIPLE DEPENDENT CLAIM PRESENT (37 CFR 1.16(j))					
* If the difference in column 1 is less than zero, enter "0" in column 2.				TOTAL	

APPLICATION AS AMENDED - PART II

		(Column 1)		(Column 2)	(Column 3)			
AMENDMENT	06/02/2023	CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA		RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	* 9	Minus	** 20	= 0		x $40 =	0
	Independent (37 CFR 1.16(h))	* 1	Minus	*** 3	= 0		x $192 =	0
	☐ Application Size Fee (37 CFR 1.16(s))							
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							
							TOTAL ADD'L FEE	0

		(Column 1)		(Column 2)	(Column 3)			
AMENDMENT		CLAIMS REMAINING AFTER AMENDMENT		HIGHEST NUMBER PREVIOUSLY PAID FOR	PRESENT EXTRA		RATE ($)	ADDITIONAL FEE ($)
	Total (37 CFR 1.16(i))	*	Minus	**	=		x $0 =	
	Independent (37 CFR 1.16(h))	*	Minus	***	=		x $0 =	
	☐ Application Size Fee (37 CFR 1.16(s))							
	☐ FIRST PRESENTATION OF MULTIPLE DEPENDENT CLAIM (37 CFR 1.16(j))							
							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".	/DARRYL FORTE/
*** 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 number found in the appropriate box in column 1.	

This collection of information is required by 37 CFR 1.16. 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 12 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.

PTO/SB/26 (08-08)
Approved for use through 09/30/2008. 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 displays a valid OMB control number.

TERMINAL DISCLAIMER TO OBVIATE A DOUBLE PATENTING REJECTION OVER A "PRIOR" PATENT	Docket Number (Optional)

In re Application of: Robert Bismuth et al.

Application No.: 18/075,576

Filed: December 6, 2022

For: ANALYTICS, ALGORITHM ARCHITECTURE, AND DATA PROCESSING SYSTEM AND METHOD

The owner*, Fermat International, Inc., of 100 percent interest 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 **prior patent** No. 11249690; 11550512 as the term of said prior patent is defined in 35 U.S.C. 154 and 173, and 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 as defined in 35 U.S.C. 154 and 173 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.

Check either box 1 or 2 below, if appropriate.

1. ☐ For submissions on behalf of a business/organization (e.g., corporation, partnership, university, government agency, etc.), the undersigned is empowered to act on behalf of the business/organization.

I hereby declare that all statements made herein of my own knowledge are true and that all statements 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.

2. ☑ The undersigned is an attorney or agent of record. Reg. No. 43535

/Victor Castellucci/ — Signature
06/02/2023 — Date

Victor Castellucci — Typed or printed name

Telephone Number

☑ Terminal disclaimer fee under 37 CFR 1.20(d) included.

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

*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.

This collection of information is required by 37 CFR 1.321. 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 12 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.

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.

Electronic Patent Application Fee Transmittal

Application Number:	18075576
Filing Date:	06-Dec-2022
Title of Invention:	Analytics, Algorithm Architecture, and Data Processing System and Method
First Named Inventor/Applicant Name:	Robert Bismuth
Filer:	Victor Jay Castellucci/Maria Lima
Attorney Docket Number:	FERM0003CON200564319

Filed as Small Entity

Filing Fees for Utility under 35 USC 111(a)

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
STATUTORY OR TERMINAL DISCLAIMER	2814	1	170	170

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Extension-of-Time:				
Miscellaneous:				
			Total in USD ($)	**170**

Electronic Acknowledgement Receipt	
EFS ID:	48091999
Application Number:	18075576
International Application Number:	
Confirmation Number:	1074
Title of Invention:	Analytics, Algorithm Architecture, and Data Processing System and Method
First Named Inventor/Applicant Name:	Robert Bismuth
Customer Number:	27799
Filer:	Victor Jay Castellucci/Maria Lima
Filer Authorized By:	Victor Jay Castellucci
Attorney Docket Number:	FERM0003CON200564319
Receipt Date:	02-JUN-2023
Filing Date:	06-DEC-2022
Time Stamp:	10:19:14
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	CARD
Payment was successfully received in RAM	$170
RAM confirmation Number	E202362A19527412
Deposit Account	
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Amendment/Request for Reconsideration-After Non-Final Rejection	FINAL_1FERM0003CON2_564319_Reply.pdf	31369 0ae6801af5e6954a20726f1105222e6f78d98cd1	no	5
Warnings:					
Information:					
2	Terminal Disclaimer Filed	1FINAL_FERM0003CON2_564319_TERMINAL_DISCLAIMER.pdf	162833 ddef47eddf001ab5e80456a6fcdc47a461a6afd8	no	2
Warnings:					
Information:					
3	Fee Worksheet (SB06)	fee-info.pdf	38093 2abcacff5feb4e6d665d6b2682d3e21a367e9dfc	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	232295		

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 a 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.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In Re Application of:	**Robert Bismuth et al.**	Confirmation No.:	**1074**
Serial No.:	**18/075,576**	Group Art Unit:	**2181**
Filing Date:	**December 6, 2022**	Examiner:	**SHIN, Christopher B.**
Docket No.:	**FERM0003CON2.564319**		

For: ANALYTICS, ALGORITHM ARCHITECTURE, AND DATA PROCESSING SYSTEM AND METHOD

FILED VIA EFS-WEB

Date Filed: June 2, 2023

Commissioner for Patent
P.O. Box 1450
Alexandria, VA 22313-1450

Commissioner:

Reply Under 37 C.F.R. §1.111

Responsive to the outstanding Office Action dated April 13, 2023 (the "Office Action"), Applicant requests reconsideration in view of the amendments and/or remarks as indicated below:

A **Listing of Claims** begins on page 2;

Remarks begin on page 4.

The following listing of claims supersedes all other listings of claims in the present application. Prior to examination on the merits, please amend the claims as follows:

Listing of Claims

What is claimed is:

1. (Original) A data processing system operative in cooperation with a host compute system, the data processing system comprising:

    a router module comprising a host interface and a node interface, wherein the router module is communicatively coupled to the host compute system via the host interface; and

    multiple parallel first tier compute nodes coupled to one or more additional compute nodes in series, each compute node comprising:

        a communications link, wherein each first tier compute node is communicatively coupled to the node interface at the router module via the communications link, at different successive tiers, each compute node in a series being communicatively coupled, via its communications link to one or more additional compute nodes connected at different successive tiers;

        a data store comprising records associated with a data processing operation;

        a programmable logic component to execute the data processing operation in cooperation with the data store;

        a node memory comprising data and instructions to support operation of the programmable logic component in connection with the data processing operation and to facilitate data communications via the communications link;

        a data mover component to facilitate data communications between the programmable logic component and the node memory; and

        a storage interface component to facilitate data communications between the programmable logic component and the data store;

        wherein the storage interface component utilizes a plurality of communications channels to transfer data between the programmable logic component and the data store.

2. (Original) The data processing system of claim 1 wherein the communications link employs a serial communications protocol.

3. (Original) The data processing system of claim 1 wherein the data store comprises an Electronically Erasable Programmable Read Only Memory device.
4. (Original) The data processing system of claim 3 wherein the storage interface component employs and Open NAND Flash Interface protocol.
5. (Original) The data processing system of claim 1 wherein the programmable logic component comprises a field programmable gate array.
6. (Original) The data processing system of claim 1 further comprising a processor to manage operation of the data mover component.
7. (Original) The data processing system of claim 6 wherein the data mover component comprises a device controller.
8. (Original) The data processing system of claim 1 wherein the host compute system executes instructions to influence operation of the compute node.
9. (Original) The data processing system of claim 1 wherein the router module and the compute node are integrated into an interface card communicatively coupled to the host compute system.

REMARKS

Status of the Claims

Claims 1-9 are all the claims pending in the application. As indicated in the Listing of Claims above, claims 1-9 are all "Original," and have not been amended; no new matter has been added.

Claims 1-9 stand rejected on the ground of non-statutory double patenting as unpatentable over claims 1-9 in Applicant's own US Patent No. 11,249,690 (see paragraph 4 at page 3 of the Office Action) and on the ground of non-statutory double patenting as unpatentable over claims 1-12 in Applicant's own US Patent No. 11,550,512 (see paragraph 5 at page 4 of the Office Action). As indicated in the paragraph bridging pages 2 and 3 of the Office Action, a timely Terminal Disclaimer may be used to overcome the foregoing rejections.

Applicant requests reconsideration and allowance of all the pending claims in light of the following remarks considered in connection with the accompanying Terminal Disclaimer. The Examiner is invited to contact the undersigned by telephone if, in the Examiner's view, the present application is not in condition for allowance.

The Terminal Disclaimer

Applicant is submitting herewith a Terminal Disclaimer that addresses both US Patent No. 11,249,690 and US Patent No. 11,550,512, which is sufficient to obviate the non-statutory double patenting rejection of claims 1-9. Accordingly, all of the pending claims 1-9 are allowable as presented.

Conclusion

Based upon the remarks set forth above and the documents submitted in connection with this Reply, it is respectfully submitted that the present application is now in condition for allowance. Prompt and favorable action to this effect and early passing of the application to issue are respectfully requested. The Examiner is invited to contact the undersigned to discuss this application if the Examiner believes that such discussion may be productive in advancing the application.

[REMAINDER OF PAGE INTENTIONALLY LEFT BLANK]

No additional fees except for the Terminal Disclaimer fee, which is submitted herewith are believed to be necessary at this time. Should any additonal fees be required in connection with this Reply, the Office is authorized to charge any underpayment or to credit any overpayment to Deposit Account No. 50-3111.

Date: June 2, 2023

Respectfully submitted,

/Victor Castellucci/

Victor Castellucci
Attorney for Applicant
Reg. No. 43,535

COZEN O'CONNOR

1200 19th Street NW
Washington, DC 20036

Application Number * 18/075,576 *	Application/Control No. 18/075,576	Applicant(s)/Patent under Reexamination Bismuth et al.	
	Examiner SHIN, CHRISTOPHER B	Art Unit 2181	

Document Code - DISQ	Internal Document - DO NOT MAIL

TERMINAL DISCLAIMER	☑ APPROVED	☐ DISAPPROVED
Date Filed: 02 June 2023	This patent is subject to a Terminal Disclaimer	

Approved/Disapproved by:
/JEAN PROCTOR/ **Technology Center: OPLC** **Telephone: (571)272-1040** _____

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

27799 7590 07/06/2023

COZEN O''CONNOR
3WTC, 175 Greenwich Street
55th Floor
NEW YORK, NY 10007

EXAMINER
SHIN, CHRISTOPHER B

ART UNIT	PAPER NUMBER
2181	

DATE MAILED: 07/06/2023

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
18/075,576	12/06/2022	Robert Bismuth	FERM0003CON200564319	1074

TITLE OF INVENTION: Analytics, Algorithm Architecture, and Data Processing System and Method

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	SMALL	$480	$0.00	$0.00	$480	10/06/2023

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 EFS-Web.

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)

27799 7590 07/06/2023
COZEN O''CONNOR
3WTC, 175 Greenwich Street
55th Floor
NEW YORK, NY 10007

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 EFS-Web 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.
18/075,576	12/06/2022	Robert Bismuth	FERM0003CON200564319	1074

TITLE OF INVENTION: Analytics, Algorithm Architecture, and Data Processing System and Method

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	SMALL	$480	$0.00	$0.00	$480	10/06/2023

EXAMINER	ART UNIT	CLASS-SUBCLASS
SHIN, CHRISTOPHER B	2181	711-103000

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 Patent Center or EFS-Web ❑ 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 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.
18/075,576	12/06/2022	Robert Bismuth	FERM0003CON200564319	1074

27799 7590 07/06/2023

COZEN O''CONNOR
3WTC, 175 Greenwich Street
55th Floor
NEW YORK, NY 10007

EXAMINER	
SHIN, CHRISTOPHER B	
ART UNIT	**PAPER NUMBER**
2181	

DATE MAILED: 07/06/2023

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. 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.

Notice of Allowability	**Application No.** 18/075,576	**Applicant(s)** Bismuth et al.	
	Examiner CHRISTOPHER B SHIN	**Art Unit** 2181	**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 the Amendment /TD received 6/2/2023.
   ☐ 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 1-9 . 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 _____.

/CHRISTOPHER B SHIN/
Primary Examiner, Art Unit 2181

Search Notes	Application/Control No.	Applicant(s)/Patent Under Reexamination
	18/075,576	Bismuth et al.
	Examiner	Art Unit
	CHRISTOPHER B SHIN	2181

CPC - Searched*		
Symbol	**Date**	**Examiner**
(G06F3/0661 G06F3/0613 G06F3/067 G06F9/3867).CPC.	04/07/2023	CS

CPC Combination Sets - Searched*		
Symbol	**Date**	**Examiner**

US Classification - Searched*			
Class	**Subclass**	**Date**	**Examiner**
711	103	04/07/2023	CS

* 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**
PDP Searched	04/07/2023	CS
PARENT CASE SEARCHED	04/07/2023	CS
PE2E See the Record	04/07/2023	CS

Interference Search			
US Class/CPC Symbol	**US Subclass/CPC Group**	**Date**	**Examiner**
711 / G06F	103 / (3/0661; 3/0613; 3/067; 9/3867)	06/15/2023	CS

/CHRISTOPHER B SHIN/ Primary Examiner, Art Unit 2181	

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	18/075,576	Bismuth et al.
	Examiner	**Art Unit**
	CHRISTOPHER B SHIN	2181

CPC							
Symbol						**Type**	**Version**
G06F	/	3	/	0661		F	2013-01-01
G06F	/	9	/	3867		I	2013-01-01
G06F	/	3	/	0613		I	2013-01-01
G06F	/	3	/	067		I	2013-01-01

CPC Combination Sets								
Symbol						**Type**	**Set**	**Ranking**	**Version**
	/		/						

NONE		Total Claims Allowed:	
(Assistant Examiner)	(Date)	9	
/CHRISTOPHER B SHIN/ Primary Examiner, Art Unit 2181	16 June 2023	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	1	1

U.S. Patent and Trademark Office — Part of Paper No.: 20230615

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	18/075,576	Bismuth et al.
	Examiner	**Art Unit**
	CHRISTOPHER B SHIN	2181

INTERNATIONAL CLASSIFICATION				
CLAIMED				
G06F	/	3	/	06
G06F	/	9	/	38

NON-CLAIMED				
	/		/	

US ORIGINAL CLASSIFICATION	
CLASS	**SUBCLASS**
711	103

CROSS REFERENCES(S)						
CLASS	**SUBCLASS (ONE SUBCLASS PER BLOCK)**					

NONE (Assistant Examiner)	(Date)	**Total Claims Allowed:** 9	
/CHRISTOPHER B SHIN/ Primary Examiner, Art Unit 2181 (Primary Examiner)	16 June 2023 (Date)	O.G. Print Claim(s) 1	O.G. Print Figure 1

U.S. Patent and Trademark Office — Part of Paper No.: 20230615

Issue Classification	Application/Control No.	Applicant(s)/Patent Under Reexamination
	18/075,576	Bismuth et al.
	Examiner	**Art Unit**
	CHRISTOPHER B SHIN	2181

☑ 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

NONE		**Total Claims Allowed:**	
(Assistant Examiner)	(Date)	9	
/CHRISTOPHER B SHIN/ Primary Examiner, Art Unit 2181	16 June 2023	O.G. Print Claim(s)	O.G. Print Figure
(Primary Examiner)	(Date)	1	1

U.S. Patent and Trademark Office — Part of Paper No.: 20230615

PE2E SEARCH - Search History (Prior Art)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
L3	10	((("BISMUTH") near3 ("Robert")) OR (("STENGLE") near3 ("Mike"))).INV.	(US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT)	OR	OFF	OFF	2021/06/01 04:23 PM
L4	2	((("FERMAT") near3 ("INTERNATIONAL") near3 ("INC"))).AS,AANM.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/06/01 05:14 PM
L6	235	tier with node with memory	(USPAT)	OR	OFF	OFF	2021/06/01 05:20 PM
L7	10626	compute near6 node	(USPAT)	OR	OFF	OFF	2021/06/01 06:02 PM
L8	235	router with L7	(USPAT)	OR	OFF	OFF	2021/06/01 06:03 PM
L9	16	host with L8	(USPAT)	OR	OFF	OFF	2021/06/01 06:03 PM
L10	56	L7 with tier	(USPAT)	OR	OFF	OFF	2021/06/01 06:11 PM
L11	10626	compute near6 node	(USPAT)	OR	OFF	OFF	2021/06/02 04:41 PM
L12	96	L11 with tier	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/06/02 04:41 PM
L13	56	L11 with tier	(USPAT)	OR	OFF	OFF	2021/06/02 04:41 PM
L14	40	L12 not L13	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/06/02 04:41 PM
L15	6	"20150301900" "20140289458"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/06/02 04:43 PM
L16	21	mover with compute with data with node	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/06/02 04:45 PM
L17	64448	(G06F3/0661 G06F3/0613 G06F3/067 G06F9/3867).CPC.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/06/02 04:56 PM
L18	488	L11 AND L17	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/06/02 04:57 PM
L19	0	mover with compute	(US-PGPUB; USPAT;	OR	ON	ON	2021/06/02

		with PIPELINE with node	USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)				07:27 PM
L20	324	compute with PIPELINE with node	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/06/02 07:28 PM
L21	2012	COMMUNICATION WITH NODE WITH PROGRAM$4 WITH LINK	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/06/02 08:57 PM
L22	935	DATA WITH L21	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/06/02 08:59 PM
L23	64448	(G06F3/0661 G06F3/0613 G06F3/067 G06F9/3867).CPC.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/06/02 08:59 PM
L24	23	L23 AND L22	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/06/02 08:59 PM
L25	3	"20150301900"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/10/29 12:31 PM
L26	11375	compute near6 node	(USPAT)	OR	OFF	OFF	2021/10/29 04:06 PM
L27	264	router with L26	(USPAT)	OR	OFF	OFF	2021/10/29 04:06 PM
L28	257	tier with node with memory	(USPAT)	OR	OFF	OFF	2021/10/29 04:06 PM
L29	264	router with L26	(USPAT)	OR	OFF	OFF	2021/10/29 04:06 PM
L30	17	host with L29	(USPAT)	OR	OFF	OFF	2021/10/29 04:06 PM
L31	2074	COMMUNICATION WITH NODE WITH PROGRAM$4 WITH LINK	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/10/29 04:07 PM
L32	965	DATA WITH L31	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/10/29 04:07 PM
L33	67651	(G06F3/0661 G06F3/0613 G06F3/067 G06F9/3867).CPC.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/10/29 04:07 PM
L34	367	compute with PIPELINE with node	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT;	OR	ON	ON	2021/10/29 04:07 PM

			IBM_TDB)				
L35	10	((("BISMUTH") near3 ("Robert")) OR (("STENGLE") near3 ("Mike"))).INV.	(US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT)	OR	OFF	OFF	2021/11/26 12:51 PM
L36	3	((("FERMAT") near3 ("INTERNATIONAL") near3 ("INC"))).AS,AANM.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/11/26 12:51 PM
L38	67994	(G06F3/0661 G06F3/0613 G06F3/067 G06F9/3867).CPC.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/11/26 12:52 PM
L39	369	compute with PIPELINE with node	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2021/11/26 12:52 PM
L40	309	(tier WITH node WITH memory)	(USPAT)	OR	OFF	OFF	2022/09/07 04:34 PM
L41	456	(compute WITH PIPELINE WITH node)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/09/07 04:34 PM
L42	955	(tier WITH node WITH memory)	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	OFF	OFF	2022/09/07 04:34 PM
L43	655	compute WITH PIPELINE WITH node	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/09/07 04:34 PM

L44	579	tier WITH node WITH memory	(US-PGPUB; USPAT)	OR	OFF	OFF	2022/09/07 04:35 PM
L45	27	mover WITH compute WITH data WITH node	(US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2022/09/07 04:35 PM
L46	784	router WITH (compute near6 node)	(US-PGPUB; USPAT)	OR	OFF	OFF	2022/09/07 04:35 PM
L47	784	router WITH (compute near6 node)	(US-PGPUB; USPAT; USOCR)	OR	OFF	OFF	2022/09/07 04:35 PM
L48	115	(compute near6 node) WITH tier	(US-PGPUB; USPAT)	OR	OFF	OFF	2022/09/07 04:36 PM
L50	579	tier WITH node WITH memory	(US-PGPUB; USPAT)	OR	OFF	OFF	2022/09/07 04:40 PM
L51	955	tier WITH node WITH memory	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	OFF	OFF	2022/09/07 04:41 PM
L52	3	"11249690"	(US-PGPUB; USPAT)	OR	ON	ON	2022/09/09 11:41 AM
L53	59217	(G06F3/0661 G06F3/0613 G06F3/067 G06F9/3867).CPC.	(US-PGPUB; USPAT; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO))	OR	ON	ON	2022/10/21 01:31 PM
L54	3	45 AND 46	(US-PGPUB; USPAT)	OR	ON	ON	2022/10/21 01:37 PM
L55	4	("20040268088" OR "20140289458" OR "20150301900" OR "20160196077").pn.	(US-PGPUB; USPAT)	OR	ON	ON	2022/11/03 06:46 PM
L56	6	"11550512" "11249690"	(US-PGPUB; USPAT)	OR	ON	ON	2023/04/07 01:12 PM

L57	1061	tier WITH node WITH memory	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	OFF	OFF	2023/04/07 03:25 PM
L58	642	tier WITH node WITH memory	(US-PGPUB; USPAT)	OR	OFF	OFF	2023/04/07 03:25 PM
L60	128	(compute near6 node) WITH tier	(US-PGPUB; USPAT)	OR	OFF	OFF	2023/04/07 03:25 PM
L61	83420	(G06F3/0661 G06F3/0613 G06F3/067 G06F9/3867).CPC.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/04/07 03:26 PM
L62	181	(compute near6 node) WITH tier	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	OFF	OFF	2023/04/07 03:26 PM
L63	1061	tier WITH node WITH memory	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	OFF	OFF	2023/04/07 03:26 PM

L64	1061	tier WITH node WITH memory	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	OFF	OFF	2023/04/07 03:27 PM
L65	521	(compute WITH PIPELINE WITH node)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/04/07 03:27 PM
L66	1061	(tier WITH node WITH memory)	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	OFF	OFF	2023/04/07 03:27 PM
L67	2	("20040268088" OR "20160196077").pn.	(US-PGPUB; USPAT)	OR	ON	ON	2023/04/08 01:54 PM
L68	17	((("BISMUTH") near3 ("Robert")) OR (("STENGLE") near3 ("Mike"))).INV.	(US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT)	OR	OFF	OFF	2023/06/15 08:51 PM
L69	11	((("FERMAT") near3 ("INTERNATIONAL") near3 ("INC"))).AS,AANM.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/06/15 08:51 PM
L70	350	tier with node with memory	(USPAT)	OR	OFF	OFF	2023/06/15 08:51 PM
L71	14242	compute near6 node	(USPAT)	OR	OFF	OFF	2023/06/15 08:51 PM
L72	417	router with L7	(USPAT)	OR	OFF	OFF	2023/06/15 08:51 PM
L73	29	host with L8	(USPAT)	OR	OFF	OFF	2023/06/15 08:51 PM

L74	70	L7 with tier	(USPAT)	OR	OFF	OFF	2023/06/15 08:51 PM
L75	14242	compute near6 node	(USPAT)	OR	OFF	OFF	2023/06/15 08:51 PM
L76	228	L11 with tier	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/06/15 08:51 PM
L77	70	L11 with tier	(USPAT)	OR	OFF	OFF	2023/06/15 08:51 PM
L78	78	L12 not L13	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/06/15 08:51 PM
L79	7	"20150301900" "20140289458"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/06/15 08:51 PM
L80	29	mover with compute with data with node	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/06/15 08:51 PM
L81	84182	(G06F3/0661 G06F3/0613 G06F3/067 G06F9/3867).CPC.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/06/15 08:51 PM
L82	2068	L11 AND L17	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/06/15 08:51 PM
L83	3	mover with compute with PIPELINE with node	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/06/15 08:51 PM
L84	545	compute with PIPELINE with node	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/06/15 08:51 PM
L85	2483	COMMUNICATION WITH NODE WITH PROGRAM$4 WITH LINK	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/06/15 08:51 PM
L86	1212	DATA WITH L21	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/06/15 08:51 PM
L87	84182	(G06F3/0661 G06F3/0613 G06F3/067 G06F9/3867).CPC.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/06/15 08:51 PM
L88	24	L23 AND L22	(US-PGPUB; USPAT; USOCR; FPRS; EPO;	OR	ON	ON	2023/06/15 08:51 PM

			JPO; DERWENT; IBM_TDB)				
L89	4	"20150301900"	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/06/15 08:51 PM
L90	14242	compute near6 node	(USPAT)	OR	OFF	OFF	2023/06/15 08:51 PM
L91	417	router with L26	(USPAT)	OR	OFF	OFF	2023/06/15 08:51 PM
L92	350	tier with node with memory	(USPAT)	OR	OFF	OFF	2023/06/15 08:51 PM
L93	417	router with L26	(USPAT)	OR	OFF	OFF	2023/06/15 08:51 PM
L94	29	host with L29	(USPAT)	OR	OFF	OFF	2023/06/15 08:51 PM
L95	2483	COMMUNICATION WITH NODE WITH PROGRAM$4 WITH LINK	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/06/15 08:51 PM
L96	1212	DATA WITH L31	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/06/15 08:51 PM
L97	84182	(G06F3/0661 G06F3/0613 G06F3/067 G06F9/3867).CPC.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/06/15 08:51 PM
L98	545	compute with PIPELINE with node	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/06/15 08:51 PM
L99	17	((("BISMUTH") near3 ("Robert")) OR (("STENGLE") near3 ("Mike"))).INV.	(US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT)	OR	OFF	OFF	2023/06/15 08:51 PM
L100	11	((("FERMAT") near3 ("INTERNATIONAL") near3 ("INC"))).AS,AANM.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/06/15 08:51 PM
L101	84182	(G06F3/0661 G06F3/0613 G06F3/067 G06F9/3867).CPC.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/06/15 08:51 PM
L102	545	compute with PIPELINE with node	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/06/15 08:51 PM
L103	350	(tier WITH node WITH memory)	(USPAT)	OR	OFF	OFF	2023/06/15 08:51 PM
L104	545	(compute WITH	(US-PGPUB; USPAT;	OR	ON	ON	2023/06/15

		PIPELINE WITH node)	USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)				08:51 PM
L105	1091	(tier WITH node WITH memory)	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	OFF	OFF	2023/06/15 08:51 PM
L106	770	compute WITH PIPELINE WITH node	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/06/15 08:51 PM
L107	656	tier WITH node WITH memory	(US-PGPUB; USPAT)	OR	OFF	OFF	2023/06/15 08:51 PM
L108	29	mover WITH compute WITH data WITH node	(US-PGPUB; USPAT; USOCR; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/06/15 08:51 PM
L109	896	router WITH (compute near6 node)	(US-PGPUB; USPAT)	OR	OFF	OFF	2023/06/15 08:51 PM
L110	896	router WITH (compute near6 node)	(US-PGPUB; USPAT; USOCR)	OR	OFF	OFF	2023/06/15 08:51 PM
L111	131	(compute near6 node) WITH tier	(US-PGPUB; USPAT)	OR	OFF	OFF	2023/06/15 08:51 PM
L112	656	tier WITH node WITH memory	(US-PGPUB; USPAT)	OR	OFF	OFF	2023/06/15 08:51 PM
L113	1091	tier WITH node WITH memory	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB,	OR	OFF	OFF	2023/06/15 08:51 PM

			HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)				
L114	5	"11249690"	(US-PGPUB; USPAT)	OR	ON	ON	2023/06/15 08:51 PM
L115	63382	(G06F3/0661 G06F3/0613 G06F3/067 G06F9/3867).CPC.	(US-PGPUB; USPAT; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO))	OR	ON	ON	2023/06/15 08:51 PM
L116	5	45 AND 46	(US-PGPUB; USPAT)	OR	ON	ON	2023/06/15 08:51 PM
L117	4	("20040268088" OR "20140289458" OR "20150301900" OR "20160196077").pn.	(US-PGPUB; USPAT)	OR	ON	ON	2023/06/15 08:51 PM
L118	6	"11550512" "11249690"	(US-PGPUB; USPAT)	OR	ON	ON	2023/06/15 08:51 PM
L119	1091	tier WITH node WITH memory	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	OFF	OFF	2023/06/15 08:51 PM
L120	656	tier WITH node WITH memory	(US-PGPUB; USPAT)	OR	OFF	OFF	2023/06/15 08:51 PM
L121	131	(compute near6 node) WITH tier	(US-PGPUB; USPAT)	OR	OFF	OFF	2023/06/15 08:51 PM

L122	84182	(G06F3/0661 G06F3/0613 G06F3/067 G06F9/3867).CPC.	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/06/15 08:51 PM
L123	190	(compute near6 node) WITH tier	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	OFF	OFF	2023/06/15 08:51 PM
L124	1091	tier WITH node WITH memory	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	OFF	OFF	2023/06/15 08:51 PM
L125	1091	tier WITH node WITH memory	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	OFF	OFF	2023/06/15 08:51 PM
L126	545	(compute WITH PIPELINE WITH node)	(US-PGPUB; USPAT; USOCR; FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	ON	ON	2023/06/15 08:51 PM

L127	1091	(tier WITH node WITH memory)	(US-PGPUB; USPAT; USOCR; FIT (AP, AT, AU, BE, BG, BR, BY, CA, CH, CN, CS, CU, CZ, DD, DE, DK, EA, EE, EP, ES, FI, FR, GB, HR, HU, ID, IE, IL, IS, IT, JP, KR, LT, LU, LV, MA, MC, MD, MY, NL, NO, NZ, OA, PH, PL, PT, RO, RS, RU, SE, SG, SI, SK, SU, TH, TN, TR, TW, UA, VN, WO); FPRS; EPO; JPO; DERWENT; IBM_TDB)	OR	OFF	OFF	2023/06/15 08:51 PM
L128	2	("20040268088" OR "20160196077").pn.	(US-PGPUB; USPAT)	OR	ON	ON	2023/06/15 08:51 PM

PE2E SEARCH - Search History (Interference)

Ref #	Hits	Search Query	DBs	Default Operator	Plurals	British Equivalents	Time Stamp
N40	7	((("BISMUTH") near3 ("Robert")) OR (("STENGLE") near3 ("Mike"))).INV.	(US-PGPUB; USPAT)	OR	OFF	OFF	2021/11/26 12:53 PM
N41	1	((("FERMAT") near3 ("INTERNATIONAL") near3 ("INC"))).AS,AANM.	(US-PGPUB; USPAT)	OR	ON	ON	2021/11/26 12:53 PM
N42	31911	(G06F3/0661 G06F3/0613 G06F3/067 G06F9/3867).CPC.	(US-PGPUB; USPAT)	OR	ON	ON	2021/11/26 12:53 PM
N43	353	compute with PIPELINE with node	(US-PGPUB; USPAT)	OR	ON	ON	2021/11/26 12:54 PM
N44	496	tier with node with memory	(US-PGPUB; USPAT)	OR	OFF	OFF	2021/11/26 12:54 PM
N45	34805	(G06F3/0661 G06F3/0613 G06F3/067 G06F9/3867).CPC.	(US-PGPUB; USPAT)	OR	ON	ON	2022/10/21 01:32 PM
N46	441	compute WITH PIPELINE WITH node$3	(US-PGPUB; USPAT)	OR	ON	ON	2022/10/21 01:32 PM
N47	746	tier WITH node$2 WITH memory	(US-PGPUB; USPAT)	OR	OFF	OFF	2022/10/21 01:32 PM
N48	139	45 AND 47	(US-PGPUB; USPAT)	OR	OFF	OFF	2022/10/21 01:37 PM
N49	24	45 AND 46	(US-PGPUB; USPAT)	OR	ON	ON	2022/10/21 01:37 PM
N50	14	((("BISMUTH") near3	(US-PGPUB; USPAT)	OR	OFF	OFF	2023/06/15

		("Robert")) OR (("STENGLE") near3 ("Mike"))).INV.					08:52 PM
N51	8	((("FERMAT") near3 ("INTERNATIONAL") near3 ("INC"))).AS,AANM.	(US-PGPUB; USPAT)	OR	ON	ON	2023/06/15 08:52 PM
N52	36812	(G06F3/0661 G06F3/0613 G06F3/067 G06F9/3867).CPC.	(US-PGPUB; USPAT)	OR	ON	ON	2023/06/15 08:52 PM
N53	513	compute with PIPELINE with node	(US-PGPUB; USPAT)	OR	ON	ON	2023/06/15 08:52 PM
N54	656	tier with node with memory	(US-PGPUB; USPAT)	OR	OFF	OFF	2023/06/15 08:52 PM
N55	36812	(G06F3/0661 G06F3/0613 G06F3/067 G06F9/3867).CPC.	(US-PGPUB; USPAT)	OR	ON	ON	2023/06/15 08:52 PM
N56	513	compute WITH PIPELINE WITH node$3	(US-PGPUB; USPAT)	OR	ON	ON	2023/06/15 08:52 PM
N57	820	tier WITH node$2 WITH memory	(US-PGPUB; USPAT)	OR	OFF	OFF	2023/06/15 08:52 PM
N58	156	45 AND 47	(US-PGPUB; USPAT)	OR	OFF	OFF	2023/06/15 08:52 PM
N59	32	45 AND 46	(US-PGPUB; USPAT)	OR	ON	ON	2023/06/15 08:52 PM

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 #
18/075,576	**10/02/2023 04:51:50 PM ET**	**FERM0003CON200564319**

Title of Invention

Analytics, Algorithm Architecture, and Data Processing System and Method

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	-
CONFIRMATION #	1074	FILED BY	Victor Castellucci
PATENT CENTER #	62913459	FILING DATE	12/06/2022
CUSTOMER #	27799	FIRST NAMED INVENTOR	Robert Bismuth
CORRESPONDENCE ADDRESS	-	AUTHORIZED BY	-

Documents

TOTAL DOCUMENTS: 1

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
FERM0809-10-03US_PoA_Documents.pdf	8	Power of Attorney	692 KB

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)
FERM0809-10-03US_PoA_Documents.pdf	CEE7612E28AC7F27D12BD0E40E525A8037FBD0500FB758022D18D5016E0DCA1AD0D6AB2E274C46706AA422A8C2FBECC423CA9A032ED118613DB92A1A4F47E10F

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.

PTO/AIA/80 (07-17)
Approved for use through 09/30/2025. OMB 0651-0035
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no person is required to respond to a collection of information unless it displays a valid OMB control number

POWER OF ATTORNEY TO PROSECUTE APPLICATIONS BEFORE THE USPTO

I hereby revoke all previous powers of attorney given in the application identified in the attached statement under 37 CFR 3.73(c).

I hereby appoint:

[X] Practitioners associated with Customer Number: 56074

OR

[ ] Practitioner(s) named below (if more than ten patent practitioners are to be named, then a customer number must be used):

Name	Registration Number	Name	Registration Number

As attorney(s) or agent(s) to represent the undersigned before the United States Patent and Trademark Office (USPTO) in connection with any and all patent applications assigned only to the undersigned according to the USPTO assignment records or assignment documents attached to this form in accordance with 37 CFR 3.73(c).

Please change the correspondence address for the application identified in the attached statement under 37 CFR 3.73(c) to:

[X] The address associated with Customer Number: 56074

OR

[ ]

Firm or individual name		
Address		
City	State	Zip
Country		
Telephone	Email	

Assignee name and address:

A copy of this form, together with a statement under 37 CFR 3.73(c) (Form PTO/AIA/96 or equivalent) is required to be filed in each application in which this form is used. The statement under 37 CFR 3.73(c) may be completed by one of the practitioners appointed in this form, and must identify the application in which this Power of Attorney is to be filed.

SIGNATURE of Assignee of Record

The individual whose signature and title is supplied below is authorized to act on behalf of the assignee.

Signature	Date 10/2/2023
Name Robert Bismuth	Telephone 206 941 1923
Title President & CEO	

A Federal agency may not conduct or sponsor, and a person is not required to respond to, nor shall a person be subject to a penalty for failure to comply with an information collection subject to the requirements of the Paperwork Reduction Act of 1995, unless the information collection has a currently valid OMB Control Number. The OMB Control Number for this information collection is 0651-0035. Public burden for this form is estimated to average 18 minutes per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the information collection. Send comments regarding this burden estimate or any other aspect of this information collection, including suggestions for reducing this burden to the Chief Administrative Officer, United States Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450 or email InformationCollection@uspto.gov. **DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS.** If filing this completed form by mail, 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.

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 9) 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.

Additional Uses

Additional USPTO uses of the information in this record may include disclosure to: 1) the International Bureau of the World Intellectual Property Organization, if the record is related to an international application filed under the Patent Cooperation Treaty; 2) the public i) after publication of the application pursuant to 35 U.S.C. 122(b), ii) after issuance of a patent pursuant to 35 U.S.C. 151, iii) 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, or iv) without publication of the application or patent under the specific circumstances provided for by 37 CFR 1.14(a)(1)(v)-(vii); and/or 3) the National Archives and Records Administration, for inspection of records.

PTO/AIA/96 (08-12)
Approved for use through 11/30/2020. 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 displays a valid OMB control number.

STATEMENT UNDER 37 CFR 3.73(c)

Applicant/Patent Owner: Fermat International, Inc.

Application No./Patent No.: 18/075,576 Filed/Issue Date: 12/06/2022

Titled: Analytics, Algorith Architecture, and Data Processing System and Method

Fermat International, Inc., a corporation

(Name of Assignee) (Type of Assignee, e.g., corporation, partnership, university, government agency, etc.)

states that, for the patent application/patent identified above, it is (choose **one** of options 1, 2, 3 or 4 below):

1. ☑ The assignee of the entire right, title, and interest.

2. ☐ An assignee of less than the entire right, title, and interest (check applicable box):

   ☐ The extent (by percentage) of its ownership interest is _____%. Additional Statement(s) by the owners holding the balance of the interest must be submitted to account for 100% of the ownership interest.

   ☐ There are unspecified percentages of ownership. The other parties, including inventors, who together own the entire right, title and interest are:

   Additional Statement(s) by the owner(s) holding the balance of the interest must be submitted to account for the entire right, title, and interest.

3. ☐ The assignee of an undivided interest in the entirety (a complete assignment from one of the joint inventors was made). The other parties, including inventors, who together own the entire right, title, and interest are:

   Additional Statement(s) by the owner(s) holding the balance of the interest must be submitted to account for the entire right, title, and interest.

4. ☐ The recipient, via a court proceeding or the like (*e.g.*, bankruptcy, probate), of an undivided interest in the entirety (a complete transfer of ownership interest was made). The certified document(s) showing the transfer is attached.

The interest identified in option 1, 2 or 3 above (not option 4) is evidenced by either (choose **one** of options A or B below):

A. ☑ An assignment from the inventor(s) of the patent application/patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel 063060, Frame 0543, or for which a copy thereof is attached.

B. ☐ A chain of title from the inventor(s), of the patent application/patent identified above, to the current assignee as follows:

   1. From: _____ To: _____

      The document was recorded in the United States Patent and Trademark Office at

      Reel _____, Frame _____, or for which a copy thereof is attached.

   2. From: _____ To: _____

      The document was recorded in the United States Patent and Trademark Office at

      Reel _____, Frame _____, or for which a copy thereof is attached.

This collection of information is required by 37 CFR 3.73(b). 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 12 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.

PTO/AIA/96 (08-12)
Approved for use through 11/30/2020. 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 displays a valid OMB control number.

STATEMENT UNDER 37 CFR 3.73(c)

3. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____, Frame _____, or for which a copy thereof is attached.

4. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____, Frame _____, or for which a copy thereof is attached.

5. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____, Frame _____, or for which a copy thereof is attached.

6. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

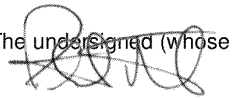
Reel _____, Frame _____, or for which a copy thereof is attached.

☐ Additional documents in the chain of title are listed on a supplemental sheet(s).

☑ As required by 37 CFR 3.73(c)(1)(i), the documentary evidence of the chain of title from the original owner to the assignee was, or concurrently is being, submitted for recordation pursuant to 37 CFR 3.11.

[NOTE: A separate copy (i.e., a true copy of the original assignment document(s)) must be submitted to Assignment Division in accordance with 37 CFR Part 3, to record the assignment in the records of the USPTO. See MPEP 302.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignee.



Signature	Date
Robert Bismuth	President & CEO
Printed or Typed Name	Title or Registration Number

PATENT ASSIGNMENT COVER SHEET

Electronic Version v1.1
Stylesheet Version v1.2

EPAS ID: PAT7859667

SUBMISSION TYPE:	NEW ASSIGNMENT
NATURE OF CONVEYANCE:	ASSIGNMENT

CONVEYING PARTY DATA

Name	Execution Date
ROBERT BISMUTH	12/31/2019
MIKE STENGLE	12/31/2019

RECEIVING PARTY DATA

Name:	FERMAT INTERNATIONAL, INC
Street Address:	2625 MIDDLEFIELD ROAD
Internal Address:	#800
City:	PALO ALTO
State/Country:	CALIFORNIA
Postal Code:	94306

PROPERTY NUMBERS Total: 1

Property Type	Number
Application Number:	18075576

CORRESPONDENCE DATA

Fax Number:

Correspondence will be sent to the e-mail address first; if that is unsuccessful, it will be sent using a fax number, if provided; if that is unsuccessful, it will be sent via US Mail.

Phone: 3053970845
Email: mlima@cozen.com
Correspondent Name: MARIA LIMA/COZEN O'CONNOR
Address Line 1: SOUTHEAST FINANCIAL CENTER
Address Line 2: 200 SOUTH BISCAYNE BLVD, SUITE 3000
Address Line 4: MIAMI, FLORIDA 33131

NAME OF SUBMITTER:	VICTOR CASTELLUCCI
SIGNATURE:	/Victor Castellucci/
DATE SIGNED:	03/22/2023

Total Attachments: 2
source=assignment-Fermat#page1.tif
source=assignment-Fermat#page2.tif

DocuSign Envelope ID: B0DF7688-0849-493D-8160-8CAF53E7E43B

DOCKET NO.: FERM0003U.464609

ASSIGNMENT

WHEREAS, we, Robert Bismuth, and Mike Stengle, maintaining an address at 3213 West Wheeler St. Unit 180 Seattle, WA 98199 and Ackerstrasse 30, 4057, Basel, Switzerland, hereinafter referred to as the assignors, are the inventors of certain inventions or improvements for which we have filed an application in the United States Patent and Trademark Office, identified as U.S. Application No. 16729727, entitled ANALYTICS, ALGORITHM ARCHITECTURE, AND DATA PROCESSING SYSTEM AND METHOD and

WHEREAS, Fermat International, Inc., hereinafter referred to as the assignee, of 2625 MIDDLEFIELD ROAD #800, PALO ALTO, CA 94306, a Delaware corporation, is desirous of acquiring the entire right, title and interest in and to the said inventions or improvements and in and to the said application, and in, to and under any and all Letters Patent which may be granted on or as a result thereof in any and all countries:

NOW, THEREFORE, for good and valuable consideration, the receipt and sufficiency of which are hereby acknowledged, and intending to be legally bound hereby, we, the said assignors, have sold, assigned, transferred and set over, and by these presents agree to and hereby irrevocably do sell, assign, transfer and set over, to said assignee, the entire right, title and interest in and to said inventions as well as any and all improvements thereto and said application and any resulting or related applications, including, but not limited to, any and all continuations, continuations-in-part, divisions, and renewals of and substitutes for said application, and in, to and under any and all Letters Patent which may be granted on or as a result thereof in the United States and any and all other countries, and any re-issue or re-issues or extension or extensions of said Letters Patent, including the full right to sue for and collect damages for past violations of provisional rights having arisen from any publication of said application or any continuations, continuations-in-part, divisions, or renewals of or substitutes for said application. We further assign to and authorize said assignee to file in our names, applications for Letters Patent in all countries, the same to be held and enjoyed by said assignee, its successors, assigns, nominees or legal representatives, to the full end of the term or terms for which said Letters Patent respectively may be granted, reissued or extended, as fully and entirely as the same would have been held and enjoyed by us had this assignment, sale and transfer not been made.

AND we hereby covenant that we have full right to convey the entire interest herein assigned, and that we have not executed and will not execute any agreement in conflict herewith, and we further covenant and agree that we will, each time request is made and without undue delay, execute and deliver all such papers as may be necessary or desirable to perfect the title to said inventions or improvements, said application and said Letters Patent to said assignee, its successors, assigns, nominees or legal representatives, and we agree to communicate to said assignee, or to its nominee all known facts respecting said inventions or improvements, said application and said Letters Patent, to testify in any legal proceedings, to sign all lawful papers, to execute all disclaimers and divisional, continuing (including continuations-in-part), reissue and foreign applications, to make all rightful oaths, and generally to do everything possible to aid said assignee, its successors, assigns, nominees and legal representatives to obtain and enforce for its or their own benefit proper patent protection for said inventions or improvements in any and all countries.

AND we hereby authorize and request the Commissioner of Patents and Trademarks of the United States and any official of any country or countries foreign to the United States whose duty it is to issue patents of applications as aforesaid, to issue to said assignee, as assignee of the entire right, title and interest, any and all Letters Patent for said inventions or improvements, including any and all Letters Patent of the United States which may be issued and granted on or as a result of the application aforesaid, in accordance with the terms of this assignment.

Each undersigned hereby grants the firm of **COZEN O'CONNOR** the power to insert on this assignment any further identification that may be necessary or desirable in order to comply with the rules of the United States Patent and Trademark Office for recordation of this document.

Assignors DocuSigned by: Robert Bismuth E6DB403AF2F349E...

Name: Robert Bismuth

Date: 12/31/2019

DocuSigned by: Mike Stengle 6E5154CD...

Name: Mike Stengle

Date: 12/31/2019

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.

PART B - FEE(S) TRANSMITTAL

Complete and send this form, together with applicable fee(s), by mail or fax, or via EFS-Web.

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)

27799 7590 07/06/2023
COZEN O''CONNOR
3WTC, 175 Greenwich Street
55th Floor
NEW YORK, NY 10007

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 EFS-Web 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.
18/075,576	12/06/2022	Robert Bismuth	FERM0003CON200564319	1074

TITLE OF INVENTION: Analytics, Algorithm Architecture, and Data Processing System and Method

APPLN. TYPE	ENTITY STATUS	ISSUE FEE DUE	PUBLICATION FEE DUE	PREV. PAID ISSUE FEE	TOTAL FEE(S) DUE	DATE DUE
nonprovisional	SMALL	$480	$0.00	$0.00	$480	10/06/2023

EXAMINER	ART UNIT	CLASS-SUBCLASS
SHIN, CHRISTOPHER B	2181	711-103000

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 Patent Center or EFS-Web ❑ 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 /Victor Castellucci/ Date 10/05/2023

Typed or printed name Victor Castellucci Registration No. 43535

Electronic Patent Application Fee Transmittal

Application Number:	18075576
Filing Date:	06-Dec-2022
Title of Invention:	Analytics, Algorithm Architecture, and Data Processing System and Method
First Named Inventor/Applicant Name:	Robert Bismuth
Filer:	Victor Jay Castellucci
Attorney Docket Number:	FERM0003CON200564319

Filed as Small Entity

Filing Fees for Utility under 35 USC 111(a)

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Basic Filing:				
Pages:				
Claims:				
Miscellaneous-Filing:				
Petition:				
Patent-Appeals-and-Interference:				
Post-Allowance-and-Post-Issuance:				
UTILITY APPL ISSUE FEE	2501	1	480	480

Description	Fee Code	Quantity	Amount	Sub-Total in USD($)
Extension-of-Time:				
Miscellaneous:				
			Total in USD ($)	**480**

Electronic Acknowledgement Receipt	
EFS ID:	48684727
Application Number:	18075576
International Application Number:	
Confirmation Number:	1074
Title of Invention:	Analytics, Algorithm Architecture, and Data Processing System and Method
First Named Inventor/Applicant Name:	Robert Bismuth
Customer Number:	27799
Filer:	Victor Jay Castellucci
Filer Authorized By:	
Attorney Docket Number:	FERM0003CON200564319
Receipt Date:	05-OCT-2023
Filing Date:	06-DEC-2022
Time Stamp:	14:02:15
Application Type:	Utility under 35 USC 111(a)

Payment information:

Submitted with Payment	yes
Payment Type	EFT
Payment was successfully received in RAM	$480
RAM confirmation Number	E202305E03109222
Deposit Account	
Authorized User	
The Director of the USPTO is hereby authorized to charge indicated fees and credit any overpayment as follows:	

File Listing:

Document Number	Document Description	File Name	File Size(Bytes)/ Message Digest	Multi Part /.zip	Pages (if appl.)
1	Issue Fee Payment (PTO-85B)	Issue_Fee_PART_B.pdf	258647 94610ac7744f95169e716076d9fac1260b30b952	no	1
Warnings:					
Information:					
2	Fee Worksheet (SB06)	fee-info.pdf	38209 a8657d30c473b6a01f652e8292748e277196f700	no	2
Warnings:					
Information:					
		Total Files Size (in bytes):	296856		

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 a 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 NUMBER	FILING OR 371(C) DATE	FIRST NAMED APPLICANT	ATTY. DOCKET NO./TITLE
18/075,576	12/06/2022	Robert Bismuth	FERM0003CON200564319

CONFIRMATION NO. 1074

IMPROPER CPOA LETTER

27799
COZEN O"CONNOR
3WTC, 175 Greenwich Street
55th Floor
NEW YORK, NY 10007

OC000000063146103

Date Mailed: 10/12/2023

NOTICE REGARDING POWER OF ATTORNEY

This is in response to the power of attorney filed 10/02/2023. The power of attorney in this application is not accepted for the reason(s) listed below:

- The revocation is not signed by the applicant or the assignee of the entire interest.
- The power of attorney has not been accepted because the party who is giving power has not been identified. Power of attorney may only be signed by the applicant for patent (37 CFR 1.42) or the patent owner. A party who is not the applicant must become the applicant in accordance with 37 CFR 1.46(c) and appoint any power of attorney in compliance with 37 CFR 3.71 and 3.73. For a reissue application, reexamination proceeding, or supplemental examination proceeding, a patent owner who was not the applicant under 37 CFR 1.46 must appoint any power of attorney in compliance with 37 CFR 3.71 and 3.73. See 37 CFR 1.32(b)(4).

/sleutchit/

Office of Data Management, Application Assistance Unit (571) 272-4000, or (571) 272-4200, or 1-888-786-0101

56074
BELLINNOVO IP
1510 Wisconsin Ave. NW
Unit 213
Washington, DC 20007
UNITED STATES

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.
18/075,576	12/06/2022	Robert Bismuth	FERM0003CON200564319	1074

27799 7590 11/30/2023
COZEN O"CONNOR
3WTC, 175 Greenwich Street
55th Floor
NEW YORK, NY 10007

EXAMINER	
SHIN, CHRISTOPHER B	

ART UNIT	PAPER NUMBER
2181	

NOTIFICATION DATE	DELIVERY MODE
11/30/2023	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):

patentdocket@cozen.com
patentsecretary@cozen.com
patentsorter@cozen.com

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

Application No. : 18075576
Applicant : Bismuth
Filing Date : 12/06/2022
Date Mailed : 11/30/2023

NOTICE TO FILE CORRECTED APPLICATION PAPERS

Notice of Allowance Mailed

This application has been accorded an Allowance Date and is being prepared for issuance. The application, however, is incomplete for the reasons below.

Applicant is given two (2) months from the mail date of this Notice within which to respond. This time period for reply is extendable under 37 CFR 1.136(a) for only TWO additional MONTHS.

The informalities requiring correction are indicated in the attachment(s). If the informality pertains to the abstract, specification (including claims) or drawings, the informality must be corrected with an amendment in compliance with 37 CFR 1.121 (or, if the application is a reissue application, 37 CFR 1.173). Such an amendment may be filed after payment of the issue fee if limited to correction of informalities noted herein. See Waiver of 37 CFR 1.312 for Documents Required by the Office of Patent Publication, 1280 Off. Gaz. Patent Office 918 (March 23, 2004). In addition, if the informality is not corrected until after payment of the issue fee, for purposes of 35 U.S.C. 154(b)(1)(iv), "all outstanding requirements" will be considered to have been satisfied when the informality has been corrected. A failure to respond within the above-identified time period will result in the application being ABANDONED.

See attachment(s).

A copy of this notice MUST be returned with the reply. Please address response to "Mail Stop Issue Fee, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450".

/Marsha Palmer-Lee/
Publication Branch
Office of Data Management
(571) 272-4200

Application No. 18075576

IDENTIFICATION OF APPLICATION DEFICIENCIES IN APPLICATION FILED ON OR AFTER SEPTEMBER 16, 2012

☐ Applicant must provide legible text for the following item(s).

- ☐ Specification filed , page(s) .
- ☐ Claims filed , claim(s) .
- ☐ Other: .

☐ Applicant must provide missing information on the following page(s) of the specification by amending the specification to add the missing text. No new matter may be added.
Page/line no(s).

☐ The specification refers to one or more applications by attorney docket number and does not show the U.S. application number(s). Applicant must supply the U.S. application number in place of each attorney docket number.
Page/line no(s).

☐ Applicant must provide an Abstract of the Disclosure.

☒ The Application Data Sheet (ADS dated 01/04/2023) does not supply the inventor's city and/or does not supply the inventor's U.S. state and/or does not supply the inventor's country. Applicant must submit a signed, in accordance with 37 CFR 1.76(e) and 1.33(b), application data sheet that corrects this deficiency. To be in compliance with 37 CFR 1.76, the corrected application data sheet must identify the information being changed by using underlining for additions and strikethroughs or brackets for deletions.

☒ Other: Please verify the 2nd inventors residence

The response to this notice should contain only amended papers, as the waiver of 37 CFR 1.312 is limited to corrections of documents specifically requested in the notice.

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 #
18/075,576	**12/14/2023 04:31:18 PM Z ET**	**FERM0003CON200564319**

Title of Invention

Analytics, Algorithm Architecture, and Data Processing System and Method

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	-
CONFIRMATION #	1074	FILED BY	Victor Castellucci
PATENT CENTER #	63620796	FILING DATE	12/06/2022
CUSTOMER #	27799	FIRST NAMED INVENTOR	Robert Bismuth
CORRESPONDENCE ADDRESS	-	AUTHORIZED BY	-

Documents

TOTAL DOCUMENTS: 1

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
Power_of_Attorney_Executed.pdf	8	Power of Attorney	4868 KB

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)
Power_of_Attorney_Executed.pdf	C2B104982D6686478F298C8A272E75AD2037F492A174639BE01F1974DE39C5A733889B0BE27624970A9F064E0292A3DD97A6DFD7E7AD0D9B3CF76EBEC2BAC9EC

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.

PTO/AIA/80 (07-17)
Approved for use through 09/30/2025. OMB 0651-0035
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no person is required to respond to a collection of information unless it displays a valid OMB control number

POWER OF ATTORNEY TO PROSECUTE APPLICATIONS BEFORE THE USPTO

I hereby revoke all previous powers of attorney given in the application identified in the attached statement under 37 CFR 3.73(c).

I hereby appoint:

[X] Practitioners associated with Customer Number: 195041

OR

[ ] Practitioner(s) named below (if more than ten patent practitioners are to be named, then a customer number must be used):

Name	Registration Number	Name	Registration Number

As attorney(s) or agent(s) to represent the undersigned before the United States Patent and Trademark Office (USPTO) in connection with any and all patent applications assigned only to the undersigned according to the USPTO assignment records or assignment documents attached to this form in accordance with 37 CFR 3.73(c).

Please change the correspondence address for the application identified in the attached statement under 37 CFR 3.73(c) to:

[X] The address associated with Customer Number: 195041

OR

[ ]

Firm or individual name		
Address		
City	State	Zip
Country		
Telephone	Email	

Assignee name and address: Fermat International, Inc.

A copy of this form, together with a statement under 37 CFR 3.73(c) (Form PTO/AIA/96 or equivalent) is required to be filed in each application in which this form is used. The statement under 37 CFR 3.73(c) may be completed by one of the practitioners appointed in this form, and must identify the application in which this Power of Attorney is to be filed.

SIGNATURE of Assignee of Record

The individual whose signature and title is supplied below is authorized to act on behalf of the assignee.

Signature	Date 12/12/2023
Name Robert Bismuth	Telephone
Title President & CEO	

A Federal agency may not conduct or sponsor, and a person is not required to respond to, nor shall a person be subject to a penalty for failure to comply with an information collection subject to the requirements of the Paperwork Reduction Act of 1995, unless the information collection has a currently valid OMB Control Number. The OMB Control Number for this information collection is 0651-0035. Public burden for this form is estimated to average 18 minutes per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the information collection. Send comments regarding this burden estimate or any other aspect of this information collection, including suggestions for reducing this burden to the Chief Administrative Officer, United States Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450 or email InformationCollection@uspto.gov. **DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS.** If filing this completed form by mail, 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.

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 9) 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.

Additional Uses

Additional USPTO uses of the information in this record may include disclosure to: 1) the International Bureau of the World Intellectual Property Organization, if the record is related to an international application filed under the Patent Cooperation Treaty; 2) the public i) after publication of the application pursuant to 35 U.S.C. 122(b), ii) after issuance of a patent pursuant to 35 U.S.C. 151, iii) 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, or iv) without publication of the application or patent under the specific circumstances provided for by 37 CFR 1.14(a)(1)(v)-(vii); and/or 3) the National Archives and Records Administration, for inspection of records.

PTO/AIA/96 (08-12)
Approved for use through 11/30/2020. 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 displays a valid OMB control number.

STATEMENT UNDER 37 CFR 3.73(c)

Applicant/Patent Owner: Fermat International, Inc.

Application No./Patent No.: 18/075,576 Filed/Issue Date: 12/06/2022

Titled: Analytics, Algorith Architecture, and Data Processing System and Method

Fermat International, Inc., a corporation

(Name of Assignee) (Type of Assignee, e.g., corporation, partnership, university, government agency, etc.)

states that, for the patent application/patent identified above, it is (choose **one** of options 1, 2, 3 or 4 below):

1. ☑ The assignee of the entire right, title, and interest.

2. ☐ An assignee of less than the entire right, title, and interest (check applicable box):

   ☐ The extent (by percentage) of its ownership interest is _____%. Additional Statement(s) by the owners holding the balance of the interest must be submitted to account for 100% of the ownership interest.

   ☐ There are unspecified percentages of ownership. The other parties, including inventors, who together own the entire right, title and interest are:

   Additional Statement(s) by the owner(s) holding the balance of the interest must be submitted to account for the entire right, title, and interest.

3. ☐ The assignee of an undivided interest in the entirety (a complete assignment from one of the joint inventors was made). The other parties, including inventors, who together own the entire right, title, and interest are:

   Additional Statement(s) by the owner(s) holding the balance of the interest must be submitted to account for the entire right, title, and interest.

4. ☐ The recipient, via a court proceeding or the like (*e.g.*, bankruptcy, probate), of an undivided interest in the entirety (a complete transfer of ownership interest was made). The certified document(s) showing the transfer is attached.

The interest identified in option 1, 2 or 3 above (not option 4) is evidenced by either (choose **one** of options A or B below):

A. ☑ An assignment from the inventor(s) of the patent application/patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel 063060, Frame 0543, or for which a copy thereof is attached.

B. ☐ A chain of title from the inventor(s), of the patent application/patent identified above, to the current assignee as follows:

   1. From: _____ To: _____

      The document was recorded in the United States Patent and Trademark Office at

      Reel _____, Frame _____, or for which a copy thereof is attached.

   2. From: _____ To: _____

      The document was recorded in the United States Patent and Trademark Office at

      Reel _____, Frame _____, or for which a copy thereof is attached.

This collection of information is required by 37 CFR 3.73(b). 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 12 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.

PTO/AIA/96 (08-12)
Approved for use through 11/30/2020. 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 displays a valid OMB control number.

STATEMENT UNDER 37 CFR 3.73(c)

3. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____, Frame _____, or for which a copy thereof is attached.

4. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____, Frame _____, or for which a copy thereof is attached.

5. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____, Frame _____, or for which a copy thereof is attached.

6. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____, Frame _____, or for which a copy thereof is attached.

☐ Additional documents in the chain of title are listed on a supplemental sheet(s).

☑ As required by 37 CFR 3.73(c)(1)(i), the documentary evidence of the chain of title from the original owner to the assignee was, or concurrently is being, submitted for recordation pursuant to 37 CFR 3.11.

[NOTE: A separate copy (i.e., a true copy of the original assignment document(s)) must be submitted to Assignment Division in accordance with 37 CFR Part 3, to record the assignment in the records of the USPTO. See MPEP 302.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignee.

Signature	Date
Robert Bismuth	President & CEO
Printed or Typed Name	Title or Registration Number

12/12/2023

[Page 2 of 2]

PATENT ASSIGNMENT COVER SHEET

Electronic Version v1.1
Stylesheet Version v1.2

EPAS ID: PAT7859667

SUBMISSION TYPE:	NEW ASSIGNMENT
NATURE OF CONVEYANCE:	ASSIGNMENT

CONVEYING PARTY DATA

Name	Execution Date
ROBERT BISMUTH	12/31/2019
MIKE STENGLE	12/31/2019

RECEIVING PARTY DATA

Name:	FERMAT INTERNATIONAL, INC
Street Address:	2625 MIDDLEFIELD ROAD
Internal Address:	#800
City:	PALO ALTO
State/Country:	CALIFORNIA
Postal Code:	94306

PROPERTY NUMBERS Total: 1

Property Type	Number
Application Number:	18075576

CORRESPONDENCE DATA

Fax Number:

Correspondence will be sent to the e-mail address first; if that is unsuccessful, it will be sent using a fax number, if provided; if that is unsuccessful, it will be sent via US Mail.

Phone: 3053970845
Email: mlima@cozen.com
Correspondent Name: MARIA LIMA/COZEN O'CONNOR
Address Line 1: SOUTHEAST FINANCIAL CENTER
Address Line 2: 200 SOUTH BISCAYNE BLVD, SUITE 3000
Address Line 4: MIAMI, FLORIDA 33131

NAME OF SUBMITTER:	VICTOR CASTELLUCCI
SIGNATURE:	/Victor Castellucci/
DATE SIGNED:	03/22/2023

Total Attachments: 2
source=assignment-Fermat#page1.tif
source=assignment-Fermat#page2.tif

DocuSign Envelope ID: B0DF7688-0849-493D-8160-8CAF53E7E43B

ASSIGNMENT

WHEREAS, we, Robert Bismuth, and Mike Stengle, maintaining an address at 3213 West Wheeler St. Unit 180 Seattle, WA 98199 and Ackerstrasse 30, 4057, Basel, Switzerland, hereinafter referred to as the assignors, are the inventors of certain inventions or improvements for which we have filed an application in the United States Patent and Trademark Office, identified as U.S. Application No. 16729727, entitled ANALYTICS, ALGORITHM ARCHITECTURE, AND DATA PROCESSING SYSTEM AND METHOD and

WHEREAS, Fermat International, Inc., hereinafter referred to as the assignee, of 2625 MIDDLEFIELD ROAD #800, PALO ALTO, CA 94306, a Delaware corporation, is desirous of acquiring the entire right, title and interest in and to the said inventions or improvements and in and to the said application, and in, to and under any and all Letters Patent which may be granted on or as a result thereof in any and all countries:

NOW, THEREFORE, for good and valuable consideration, the receipt and sufficiency of which are hereby acknowledged, and intending to be legally bound hereby, we, the said assignors, have sold, assigned, transferred and set over, and by these presents agree to and hereby irrevocably do sell, assign, transfer and set over, to said assignee, the entire right, title and interest in and to said inventions as well as any and all improvements thereto and said application and any resulting or related applications, including, but not limited to, any and all continuations, continuations-in-part, divisions, and renewals of and substitutes for said application, and in, to and under any and all Letters Patent which may be granted on or as a result thereof in the United States and any and all other countries, and any re-issue or re-issues or extension or extensions of said Letters Patent, including the full right to sue for and collect damages for past violations of provisional rights having arisen from any publication of said application or any continuations, continuations-in-part, divisions, or renewals of or substitutes for said application. We further assign to and authorize said assignee to file in our names, applications for Letters Patent in all countries, the same to be held and enjoyed by said assignee, its successors, assigns, nominees or legal representatives, to the full end of the term or terms for which said Letters Patent respectively may be granted, reissued or extended, as fully and entirely as the same would have been held and enjoyed by us had this assignment, sale and transfer not been made.

AND we hereby covenant that we have full right to convey the entire interest herein assigned, and that we have not executed and will not execute any agreement in conflict herewith, and we further covenant and agree that we will, each time request is made and without undue delay, execute and deliver all such papers as may be necessary or desirable to perfect the title to said inventions or improvements, said application and said Letters Patent to said assignee, its successors, assigns, nominees or legal representatives, and we agree to communicate to said assignee, or to its nominee all known facts respecting said inventions or improvements, said application and said Letters Patent, to testify in any legal proceedings, to sign all lawful papers, to execute all disclaimers and divisional, continuing (including continuations-in-part), reissue and foreign applications, to make all rightful oaths, and generally to do everything possible to aid said assignee, its successors, assigns, nominees and legal representatives to obtain and enforce for its or their own benefit proper patent protection for said inventions or improvements in any and all countries.

AND we hereby authorize and request the Commissioner of Patents and Trademarks of the United States and any official of any country or countries foreign to the United States whose duty it is to issue patents of applications as aforesaid, to issue to said assignee, as assignee of the entire right, title and interest, any and all Letters Patent for said inventions or improvements, including any and all Letters Patent of the United States which may be issued and granted on or as a result of the application aforesaid, in accordance with the terms of this assignment.

Each undersigned hereby grants the firm of **COZEN O'CONNOR** the power to insert on this assignment any further identification that may be necessary or desirable in order to comply with the rules of the United States Patent and Trademark Office for recordation of this document.

Assignors DocuSigned by: Robert Bismuth E6DB403AF2F349E...

Name: Robert Bismuth

Date: 12/31/2019

DocuSigned by: Mike Stengle 6E5154CD...

Name: Mike Stengle

Date: 12/31/2019

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.

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
18/075,576	12/06/2022	Robert Bismuth	FERM0003CON200564319

CONFIRMATION NO. 1074

195041
BELLINNOVO IP
1510 Wisconsin Ave. NW
Unit 213
Washington, DC 20007

POA ACCEPTANCE LETTER

OC000000066066546

Date Mailed: 12/19/2023

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 12/14/2023.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at **(571) 272-4000** or **(571) 272-4200** or **1-888-786-0101**.

/agizaw/

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
18/075,576	12/06/2022	Robert Bismuth	FERM0003CON200564319

CONFIRMATION NO. 1074

27799
COZEN O"CONNOR
3WTC, 175 Greenwich Street
55th Floor
NEW YORK, NY 10007

POWER OF ATTORNEY NOTICE

OC000000066066503

Date Mailed: 12/19/2023

NOTICE REGARDING CHANGE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 12/14/2023.

- The Power of Attorney to you in this application has been revoked by the applicant. Future correspondence will be mailed to the new address of record(37 CFR 1.33).

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at **(571) 272-4000** or **(571) 272-4200** or **1-888-786-0101**.

/agizaw/

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	FERM0809-10-03US
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

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
	Robert		Bismuth	

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

City	Seattle	State/Province	WA	Country of Residence	US

Mailing Address of Inventor:

Address 1	2113 West Wheeler St.		
Address 2			
City	Seattle	State/Province	WA
Postal Code	98199	Country	US

Inventor 2 — Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Mike		Stengle	

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

City	Basel ~~Seattle~~	Country of Residence	CH ~~US~~

Mailing Address of Inventor:

Address 1	Ackerstrasse 30		
Address 2	~~4057~~		
City	Basel	State/Province	
Postal Code	4057	Country	CH

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

Correspondence Information:

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	FERM0809-10-03US
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

Enter either Customer Number or complete the Correspondence Information section below. For further information see 37 CFR 1.33(a).			
☐ An Address is being provided for the correspondence Information of this application.			
Customer Number	195041 ~~27749~~		
Email Address	victor@bellinnovo.com ~~AIVIPDocketing@cozen.com~~	Add Email	Remove Email

Application Information:

Title of the Invention	Analytics, Algorithm Architecture, and Data Processing System and Method		
Attorney Docket Number	FERM0809-10-03US	Small Entity Status Claimed	☒
Application Type	Nonprovisional		
Subject Matter	Utility		
Total Number of Drawing Sheets (if any)	29	Suggested Figure for Publication (if any)	

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	195041		

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	FERM0809-10-03US
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

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	Patented			Remove	
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
18075576	Continuation of	17666049	2022-02-07	11550512	2022-12-21
Prior Application Status	Patented			Remove	
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
17666049	Continuation of	16729727	2019-12-30	11249690	2022-02-15
Prior Application Status	Expired		Remove		
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)		
16729727	Claims benefit of provisional	62882044	2019-08-02		
Prior Application Status	Expired		Remove		
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)		
16729727	Claims benefit of provisional	62853764	2019-05-29		
Prior Application Status	Expired		Remove		
Application Number	Continuity Type	Prior Application Number	Filing or 371(c) Date (YYYY-MM-DD)		
16729727	Claims benefit of provisional	62801839	2019-02-06		

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

Foreign Priority Information:

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)[1] 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).

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	FERM0809-10-03US
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

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

Remove

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

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.

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	FERM0809-10-03US
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

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.

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	FERM0809-10-03US
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

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

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	Fermat International, Inc.		
Mailing Address Information For Applicant:			
Address 1	27/29 rue Raffet		
Address 2			
City	Paris	**State/Province**	
Country	FR	Postal Code	75016
Phone Number		Fax Number	
Email Address			

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

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.

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	FERM0809-10-03US
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

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.

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[i]		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.

Signature:

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	/Victor Castellucci/			Date (YYYY-MM-DD)	2023-12-19
First Name	Victor	Last Name	Castellucci	Registration Number	43535

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

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	FERM0809-10-03US
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

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.

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 #
18/075,576	**12/19/2023 11:57:32 AM Z ET**	**FERM0003CON200564319**

Title of Invention

Analytics, Algorithm Architecture, and Data Processing System and Method

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	-
CONFIRMATION #	1074	FILED BY	Victor Castellucci
PATENT CENTER #	63667207	FILING DATE	12/06/2022
CUSTOMER #	195041	FIRST NAMED INVENTOR	Robert Bismuth
CORRESPONDENCE ADDRESS	-	AUTHORIZED BY	-

Documents

TOTAL DOCUMENTS: 1

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
FERM0809_10_03US_Corrected_ADS.pdf	9	Application Data Sheet	12237 KB

Warning: This is not a USPTO supplied ADS fillable form. Data in the form cannot be automatically loaded to other USPTO systems.

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)
FERM0809_10_03US_Corrected_ADS.pdf	61BE83B4A5C3A4BEF78F5EEE4D07556C67E17F93ED2CB1750C12EBCB1D3AB2B13091B1CE4232D51B0166438E265C5CC73

B12FDF394C42277437D1D87D0BA4FC2

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.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
18/075,576	12/06/2022	Robert Bismuth	FERM0003CON200564319	1074

27799 7590 11/30/2023
COZEN O"CONNOR
3WTC, 175 Greenwich Street
55th Floor
NEW YORK, NY 10007

EXAMINER	
SHIN, CHRISTOPHER B	

ART UNIT	PAPER NUMBER
2181	

NOTIFICATION DATE	DELIVERY MODE
11/30/2023	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):

patentdocket@cozen.com
patentsecretary@cozen.com
patentsorter@cozen.com

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

Application No. : 18075576
Applicant : Bismuth
Filing Date : 12/06/2022
Date Mailed : 11/30/2023

NOTICE TO FILE CORRECTED APPLICATION PAPERS

Notice of Allowance Mailed

This application has been accorded an Allowance Date and is being prepared for issuance. The application, however, is incomplete for the reasons below.

Applicant is given two (2) months from the mail date of this Notice within which to respond. This time period for reply is extendable under 37 CFR 1.136(a) for only TWO additional MONTHS.

The informalities requiring correction are indicated in the attachment(s). If the informality pertains to the abstract, specification (including claims) or drawings, the informality must be corrected with an amendment in compliance with 37 CFR 1.121 (or, if the application is a reissue application, 37 CFR 1.173). Such an amendment may be filed after payment of the issue fee if limited to correction of informalities noted herein. See Waiver of 37 CFR 1.312 for Documents Required by the Office of Patent Publication, 1280 Off. Gaz. Patent Office 918 (March 23, 2004). In addition, if the informality is not corrected until after payment of the issue fee, for purposes of 35 U.S.C. 154(b)(1)(iv), "all outstanding requirements" will be considered to have been satisfied when the informality has been corrected. A failure to respond within the above-identified time period will result in the application being ABANDONED.

See attachment(s).

A copy of this notice MUST be returned with the reply. Please address response to "Mail Stop Issue Fee, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450".

/Marsha Palmer-Lee/
Publication Branch
Office of Data Management
(571) 272-4200

Application No. 18075576

IDENTIFICATION OF APPLICATION DEFICIENCIES IN APPLICATION FILED ON OR AFTER SEPTEMBER 16, 2012

☐ Applicant must provide legible text for the following item(s).

- ☐ Specification filed , page(s) .
- ☐ Claims filed , claim(s) .
- ☐ Other: .

☐ Applicant must provide missing information on the following page(s) of the specification by amending the specification to add the missing text. No new matter may be added.
Page/line no(s).

☐ The specification refers to one or more applications by attorney docket number and does not show the U.S. application number(s). Applicant must supply the U.S. application number in place of each attorney docket number.
Page/line no(s).

☐ Applicant must provide an Abstract of the Disclosure.

☒ The Application Data Sheet (ADS dated 01/04/2023) does not supply the inventor's city and/or does not supply the inventor's U.S. state and/or does not supply the inventor's country. Applicant must submit a signed, in accordance with 37 CFR 1.76(e) and 1.33(b), application data sheet that corrects this deficiency. To be in compliance with 37 CFR 1.76, the corrected application data sheet must identify the information being changed by using underlining for additions and strikethroughs or brackets for deletions.

☒ Other: Please verify the 2nd inventors residence

The response to this notice should contain only amended papers, as the waiver of 37 CFR 1.312 is limited to corrections of documents specifically requested in the notice.

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	FERM0809-10-03US
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

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
	Robert		Bismuth	

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

City	Seattle	State/Province	WA	Country of Residence	US

Mailing Address of Inventor:

Address 1	2113 West Wheeler St.		
Address 2			
City	Seattle	State/Province	WA
Postal Code	98199	Country	US

Inventor 2 — Remove

Legal Name

Prefix	Given Name	Middle Name	Family Name	Suffix
	Mike		Stengle	

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

City	Basel ~~Seattle~~	Country of Residence	CH ~~US~~

Mailing Address of Inventor:

Address 1	Ackerstrasse 30		
Address 2	~~4057~~		
City	Basel	State/Province	
Postal Code	4057	Country	CH

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

Correspondence Information:

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	FERM0809-10-03US
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

Enter either Customer Number or complete the Correspondence Information section below. For further information see 37 CFR 1.33(a).			
☐ An Address is being provided for the correspondence Information of this application.			
Customer Number	195041 ~~27749~~		
Email Address	victor@bellinnovo.com ~~AIVIPDocketing@cozen.com~~	Add Email	Remove Email

Application Information:

Title of the Invention	Analytics, Algorithm Architecture, and Data Processing System and Method		
Attorney Docket Number	FERM0809-10-03US	Small Entity Status Claimed	☒
Application Type	Nonprovisional		
Subject Matter	Utility		
Total Number of Drawing Sheets (if any)	29	Suggested Figure for Publication (if any)	

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	195041		

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	FERM0809-10-03US
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

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	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
18075576	Continuation of	17666049	2022-02-07	11550512	2022-12-21
Prior Application Status	Patented				Remove
Application Number	Continuity Type	Prior Application Number	Filing Date (YYYY-MM-DD)	Patent Number	Issue Date (YYYY-MM-DD)
17666049	Continuation of	16729727	2019-12-30	11249690	2022-02-15
Prior Application Status	Expired				Remove
Application Number	Continuity Type	Prior Application Number			Filing or 371(c) Date (YYYY-MM-DD)
16729727	Claims benefit of provisional	62882044			2019-08-02
Prior Application Status	Expired				Remove
Application Number	Continuity Type	Prior Application Number			Filing or 371(c) Date (YYYY-MM-DD)
16729727	Claims benefit of provisional	62853764			2019-05-29
Prior Application Status	Expired				Remove
Application Number	Continuity Type	Prior Application Number			Filing or 371(c) Date (YYYY-MM-DD)
16729727	Claims benefit of provisional	62801839			2019-02-06

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

Foreign Priority Information:

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)[1] 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).

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	FERM0809-10-03US
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

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

Remove

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

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.

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	FERM0809-10-03US
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

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.

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	FERM0809-10-03US
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

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

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	Fermat International, Inc.		
Mailing Address Information For Applicant:			
Address 1	27/29 rue Raffet		
Address 2			
City	Paris	**State/Province**	
Country	FR	Postal Code	75016
Phone Number		Fax Number	
Email Address			

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

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.

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	FERM0809-10-03US
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

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.

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 i		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.

Signature:

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	/Victor Castellucci/			Date (YYYY-MM-DD)	2023-12-19
First Name	Victor	Last Name	Castellucci	Registration Number	43535

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

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	FERM0809-10-03US
	Application Number	
Title of Invention	Analytics, Algorithm Architecture, and Data Processing System and Method	

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.

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 #
18/075,576	**12/19/2023 12:12:36 PM Z ET**	**FERM0003CON200564319**

Title of Invention

Analytics, Algorithm Architecture, and Data Processing System and Method

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	-
CONFIRMATION #	1074	FILED BY	Victor Castellucci
PATENT CENTER #	63667572	FILING DATE	12/06/2022
CUSTOMER #	195041	FIRST NAMED INVENTOR	Robert Bismuth
CORRESPONDENCE ADDRESS	-	AUTHORIZED BY	-

Documents

TOTAL DOCUMENTS: 2

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
Notice_for_Corrected_ADS.pdf	3	Miscellaneous Incoming Letter	101 KB
FERM0809_10_03US_Corrected_ADS.pdf	9	Application Data Sheet	12237 KB

Warning: This is not a USPTO supplied ADS fillable form. Data in the form cannot be automatically loaded to other USPTO systems.

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)

Notice_for_Corrected_ADS.pdf	5446ADB4A0D6E0AEEA294E2F43C91304E01838EFA2326F0FD9F1BD2C8C19833B3763019B0B9D71757449CE083E153D8049191D450B83D30B98F7BF6E0F235BC0
FERM0809_10_03US_Corrected_ADS.pdf	61BE83B4A5C3A4BEF78F5EEE4D07556C67E17F93ED2CB1750C12EBCB1D3AB2B13091B1CE4232D51B0166438E265C5CC73B12FDF394C42277437D1D87D0BA4FC2

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 NUMBER	FILING or 371(c) DATE	GRP ART UNIT	FIL FEE REC'D	ATTY.DOCKET.NO	TOT CLAIMS	IND CLAIMS
18/075,576	12/06/2022	2181	830	FERM0003CON200564319	9	1

CONFIRMATION NO. 1074

REPLACEMENT FILING RECEIPT

195041
BELLINNOVO IP
1510 Wisconsin Ave. NW
Unit 213
Washington, DC 20007



Date Mailed: 12/20/2023

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)

Robert Bismuth, Seattle, WA;
Mike Stengle, BASEL, SWITZERLAND;

Applicant(s)

Fermat International, Inc., Paris, FRANCE;

Power of Attorney: The patent practitioners associated with Customer Number 195041

Domestic Priority data as claimed by applicant

This application is a CON of 17/666,049 02/07/2022 PAT 11,550,512
which is a CON of 16/729,727 12/30/2019 PAT 11,249,690
which claims benefit of 62/882,044 08/02/2019
and claims benefit of 62/853,764 05/29/2019
and claims benefit of 62/801,839 02/06/2019

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: No

Permission to Access Search Results: No

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: 12/19/2022

The country code and number of your priority application, to be used for filing abroad under the Paris Convention, is **US 18/075,576**

Projected Publication Date: 04/06/2023

Non-Publication Request: No

Early Publication Request: No
**** SMALL ENTITY ****
Title

Analytics, Algorithm Architecture, and Data Processing System and Method

Preliminary Class

711

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.

PTO/SB/08a (02-18)
Approved for use through 11/30/2020. 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	18075576
Filing Date	2022-12-06
First Named Inventor	Robert Bismuth
Art Unit	TBD
Examiner Name	TBD
Attorney Docket Number	FERM0003CON200564319

U.S.PATENTS

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.

U.S.PATENT APPLICATION PUBLICATIONS

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
Change(s) applied to document /J.G./ 12/20/2023	1	20160196077		2016-07-07	Nakamura et al. ~~Hitachi LTD~~	
	2	20040268088		2004-12-30	Lippincott Louis A.	

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

FOREIGN PATENT DOCUMENTS

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

NON-PATENT LITERATURE DOCUMENTS

ALL REFERENCES CONSIDERED EXCEPT WHERE LINED THROUGH. /C.B.S/

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.
18/075,576	01/30/2024	11886750	FERM0003CON200564319	1074

195041 7590 01/10/2024

BELLINNOVO IP
1510 Wisconsin Ave. NW
Unit 213
Washington, DC 20007

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):

Robert Bismuth, Seattle, WA;
Mike Stengle, BASEL, SWITZERLAND;

APPLICANT(s) (Please see PATENT CENTER site https://patentcenter.uspto.gov for additional applicants):

Fermat International, Inc., Paris, FRANCE;

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.
18/075,576	12/06/2022	Robert Bismuth	FERM0003CON200564319	1074

195041 7590 01/30/2024

BELLINNOVO IP
1510 Wisconsin Ave. NW
Unit 213
Washington, DC 20007

EXAMINER
SHIN, CHRISTOPHER B

ART UNIT	PAPER NUMBER
2181	

NOTIFICATION DATE	DELIVERY MODE
01/30/2024	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):

victor@bellinnovo.com

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.
18/075,576	30-Jan-2024	11886750

BELLINNOVO IP
1510 Wisconsin Ave. NW
Washington, DC 20007

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

P.O. Box 1450
Alexandria, VA 22313 - 1450
www.uspto.gov

ELECTRONIC ACKNOWLEDGEMENT RECEIPT

APPLICATION #	RECEIPT DATE / TIME	ATTORNEY DOCKET #
18/075,576	**11/26/2025 06:16:33 AM Z ET**	**FERM0003CON200564319**

Title of Invention

Analytics, Algorithm Architecture, and Data Processing System and Method

Application Information

APPLICATION TYPE	Utility - Nonprovisional Application under 35 USC 111(a)	PATENT #	11886750
CONFIRMATION #	1074	FILED BY	Jonathan Hall
PATENT CENTER #	73345938	FILING DATE	12/06/2022
CUSTOMER #	195041	FIRST NAMED INVENTOR	Robert Bismuth
CORRESPONDENCE ADDRESS	-	AUTHORIZED BY	-

Documents

TOTAL DOCUMENTS: 2

DOCUMENT	PAGES	DESCRIPTION	SIZE (KB)
11886750_373Statement.pdf	3	Assignee showing of ownership per 37 CFR 3.73	200 KB
signed_POA.pdf	2	Power of Attorney	553 KB

Warning: The PDF file has been signed with a digital signature and the legal effect of the document will be based on the contents of the file not the digital signature.

Digest

DOCUMENT	MESSAGE DIGEST(SHA-512)
11886750_373Statement.pdf	230FD80934ABC9AFCB0BF5D23962E4656A2A56024D4C2B7A BA597B217D43E99534B17F2A5641BAF0F72C578573D75B78F 9A9939307153D2DB34B2B321D256FF4

signed_POA.pdf	E23F7A15C4B1760B58E5B94A1C4C25FFCA9470065DF0E8018 56AB6EA5427C5BC9AD3C5E6C8972F28B9B1AAB73EE898CE0 E7CA0FD54DEE7A97B11B6CC22C7782F

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.

PTO/AIA/96 (11-23)
Approved for use through 11/30/2027. 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 displays a valid OMB control number.

STATEMENT UNDER 37 CFR 3.73(c)

First Named Inventor/Patent Owner: Fermat International, Inc.

Application No./Patent No.: 11,886,750 Filed/Issue Date: 2024-01-30

Titled: ANALYTICS, ALGORITHM ARCHITECTURE, AND DATA PROCESSING SYSTEM AND METHOD

Fermat International, Inc., a corporation

(Name of Assignee) (Type of Assignee, e.g., corporation, partnership, university, government agency, etc.)

states that, for the patent application/patent identified above, it is (choose **one** of options 1, 2, 3 or 4 below):

1. ☒ The assignee of the entire right, title, and interest.

2. ☐ An assignee of less than the entire right, title, and interest (check applicable box):

   ☐ The extent (by percentage) of its ownership interest is _____%. Additional Statement(s) by the owners holding the balance of the interest must be submitted to account for 100% of the ownership interest.

   ☐ There are unspecified percentages of ownership. The other parties, including inventors, who together own the entire right, title and interest are:

   Additional Statement(s) by the owner(s) holding the balance of the interest must be submitted to account for the entire right, title, and interest.

3. ☐ The assignee of an undivided interest in the entirety (a complete assignment from one of the joint inventors was made). The other parties, including inventors, who together own the entire right, title, and interest are:

   Additional Statement(s) by the owner(s) holding the balance of the interest must be submitted to account for the entire right, title, and interest.

4. ☐ The recipient, via a court proceeding or the like (*e.g.*, bankruptcy, probate), of an undivided interest in the entirety (a complete transfer of ownership interest was made). The certified document(s) showing the transfer is attached.

The interest identified in option 1, 2 or 3 above (not option 4) is evidenced by either (choose **one** of options A or B below):

A. ☒ An assignment from the inventor(s) of the patent application/patent identified above. The assignment was recorded in the United States Patent and Trademark Office at Reel 63060, Frame 0543, or for which a copy thereof is attached*.

B. ☐ A chain of title from the inventor(s), of the patent application/patent identified above, to the current assignee as follows:

   1. From: _____ To: _____

      The document was recorded in the United States Patent and Trademark Office at

      Reel _____, Frame _____, or for which a copy thereof is attached*.

   2. From: _____ To: _____

      The document was recorded in the United States Patent and Trademark Office at

      Reel _____, Frame _____, or for which a copy thereof is attached*.

A Federal agency may not conduct or sponsor, and a person is not required to respond to, nor shall a person be subject to a penalty for failure to comply with an information collection subject to the requirements of the Paperwork Reduction Act of 1995, unless the information collection has a currently valid OMB Control Number. The OMB Control Number for this information collection is 0651-0031. Public burden for this form is estimated to average 15 minutes per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the information collection. Send comments regarding this burden estimate or any other aspect of this information collection, including suggestions for reducing this burden to the Chief Administrative Officer, United States Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450 or email InformationCollection@uspto.gov. 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/AIA/96 (11-23)
Approved for use through 11/30/2027. 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 displays a valid OMB control number.

STATEMENT UNDER 37 CFR 3.73(c)

3. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____, Frame _____, or for which a copy thereof is attached*.

4. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____, Frame _____, or for which a copy thereof is attached*.

5. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____, Frame _____, or for which a copy thereof is attached*.

6. From: _____ To: _____

The document was recorded in the United States Patent and Trademark Office at

Reel _____, Frame _____, or for which a copy thereof is attached*.

☐ Additional documents in the chain of title are listed on a supplemental sheet(s).

☑ *As required by 37 CFR 3.73(c)(1)(i), the documentary evidence of the chain of title from the original owner to the assignee was, or concurrently is being, submitted for recordation pursuant to 37 CFR 3.11.

[NOTE: A separate copy (i.e., a true copy of the original assignment document(s)) must be submitted to Assignment Division in accordance with 37 CFR Part 3, to record the assignment in the records of the USPTO. See MPEP 302.08]

The undersigned (whose title is supplied below) is authorized to act on behalf of the assignee.

/Jonathan D Hall/	2025-11-25
Signature	Date
Jonathan D. Hall	67130
Printed or Typed Name	Title or Registration 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 9) 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.

Docusign Envelope ID: F116EDAE-5D70-4B8A-8E67-ACCB73576839

PTO/AIA/80 (07-17)
Approved for use through 09/30/2025. OMB 0651-0035
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no person is required to respond to a collection of information unless it displays a valid OMB control number

POWER OF ATTORNEY TO PROSECUTE APPLICATIONS BEFORE THE USPTO

I hereby revoke all previous powers of attorney given in the application identified in the attached statement under 37 CFR 3.73(c).

I hereby appoint:

[X] Practitioners associated with Customer Number: 181231

OR

[ ] Practitioner(s) named below (if more than ten patent practitioners are to be named, then a customer number must be used):

Name	Registration Number	Name	Registration Number

As attorney(s) or agent(s) to represent the undersigned before the United States Patent and Trademark Office (USPTO) in connection with any and all patent applications assigned only to the undersigned according to the USPTO assignment records or assignment documents attached to this form in accordance with 37 CFR 3.73(c).

Please change the correspondence address for the application identified in the attached statement under 37 CFR 3.73(c) to:

[X] The address associated with Customer Number: 181231

OR

[ ]

Firm or individual name		
Address		
City	State	Zip
Country		
Telephone	Email	

Assignee name and address: Fermat International, Inc.
2625 Middlefield Road, #800
Palo Alto, CA 94306
United States of America

A copy of this form, together with a statement under 37 CFR 3.73(c) (Form PTO/AIA/96 or equivalent) is required to be filed in each application in which this form is used. The statement under 37 CFR 3.73(c) may be completed by one of the practitioners appointed in this form, and must identify the application in which this Power of Attorney is to be filed.

SIGNATURE of Assignee of Record

The individual whose signature and title is supplied below is authorized to act on behalf of the assignee.

Signature: Signed by: Robert Bismuth	Date 11/7/2025
Name Robert Bismuth	Telephone
Title CEO	

A Federal agency may not conduct or sponsor, and a person is not required to respond to, nor shall a person be subject to a penalty for failure to comply with an information collection subject to the requirements of the Paperwork Reduction Act of 1995, unless the information collection has a currently valid OMB Control Number. The OMB Control Number for this information collection is 0651-0035. Public burden for this form is estimated to average 30 minutes per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the information collection. Send comments regarding this burden estimate or any other aspect of this information collection, including suggestions for reducing this burden to the Chief Administrative Officer, United States Patent and Trademark Office, P.O. Box 1450, Alexandria, VA 22313-1450 or email InformationCollection@uspto.gov. **DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS.** If filing this completed form by mail, 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.

Docusign Envelope ID: F116EDAE-5D70-4B8A-8E67-ACCB73576839

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 9) 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.

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
18/075,576	12/06/2022	Robert Bismuth	FERM0003CON200564319

CONFIRMATION NO. 1074

181231
Ethos Legal PC
30 Gruver Road
Pipersville, PA 18947

POA ACCEPTANCE LETTER

OC000000100052123

Date Mailed: 01/14/2026

NOTICE OF ACCEPTANCE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 11/26/2025.

The Power of Attorney in this application is accepted. Correspondence in this application will be mailed to the above address as provided by 37 CFR 1.33.

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at **(571) 272-4000** or **(571) 272-4200** or **1-888-786-0101**.

/atesfai/

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
18/075,576	12/06/2022	Robert Bismuth	FERM0003CON200564319

CONFIRMATION NO. 1074

195041
BELLINNOVO IP
712 H Street NE
Unit 591
Washington, DC 20002

POWER OF ATTORNEY NOTICE

OC000000100052086

Date Mailed: 01/14/2026

NOTICE REGARDING CHANGE OF POWER OF ATTORNEY

This is in response to the Power of Attorney filed 11/26/2025.

- The Power of Attorney to you in this application has been revoked by the applicant. Future correspondence will be mailed to the new address of record(37 CFR 1.33).

Questions about the contents of this notice and the requirements it sets forth should be directed to the Office of Data Management, Application Assistance Unit, at **(571) 272-4000** or **(571) 272-4200** or **1-888-786-0101**.

/atesfai/