

Microsoft Corporation
Petitioner

v.

ParTec AG
Patent Owner

IPR2025-00318

Patent 11,537,442

ParTec AG's Trial Presentation

April 14, 2026

Outline



The '442 Patent



Asserted Prior Art Overview



Petition Ground Deficiencies

The '442 Patent

(12) United States Patent Lippert et al.

(10) Patent No.: US 11,537,442 B2
(45) Date of Patent: Dec. 27, 2022

(54) APPLICATION RUNTIME DETERMINED DYNAMICAL ALLOCATION OF HETEROGENEOUS COMPUTE RESOURCES

(71) Applicant: **ParTec Cluster Competence Center
GmbH**, Munich (DE)

(72) Inventors: **Thomas Lippert**, Aschaffenburg (DE);
Bernhard Frohwitter, Munich (DE)

(73) Assignee: **ParTec Cluster Competence Center
GmbH**, Munich (DE)

(*) Notice: Subject to any disclaimer, the term of this
patent is extended or adjusted under 35
U.S.C. 154(b) by 35 days.

(21) Appl. No.: **16/963,749**

(22) PCT Filed: **Jan. 23, 2019**

(86) PCT No.: **PCT/EP2019/051615**

§ 371 (c)(1),
(2) Date: **Jul. 21, 2020**

(87) PCT Pub. No.: **WO2019/145354**

PCT Pub. Date: **Aug. 1, 2019**

(65) Prior Publication Data

US 2021/0096923 A1 Apr. 1, 2021

(30) Foreign Application Priority Data

Jan. 23, 2018 (EP) 18152903

(51) Int. CL. **G06F 9/50** (2006.01)

(52) U.S. CL. **G06F 9/5061** (2013.01); **G06F 2209/509**
(2013.01)

(58) Field of Classification Search
CPC: G06F 9/5061; G06F 2209/509
See application file for complete search history.

(56) References Cited

U.S. PATENT DOCUMENTS

2012/0317168 A1* 12/2012 Driesen G06Q 30/06
709/202
2013/0282787 A1* 10/2013 Lippert H04L 29/08081
709/201

FOREIGN PATENT DOCUMENTS

EP 2442228 A1 4/2012
WO WO-2018065530 A1 4/2018

OTHER PUBLICATIONS

International Search Report for International Application No. PCT/
EP2019/0518615, dated Apr. 29, 2019.
(Continued)

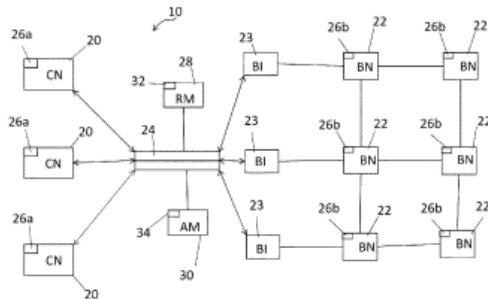
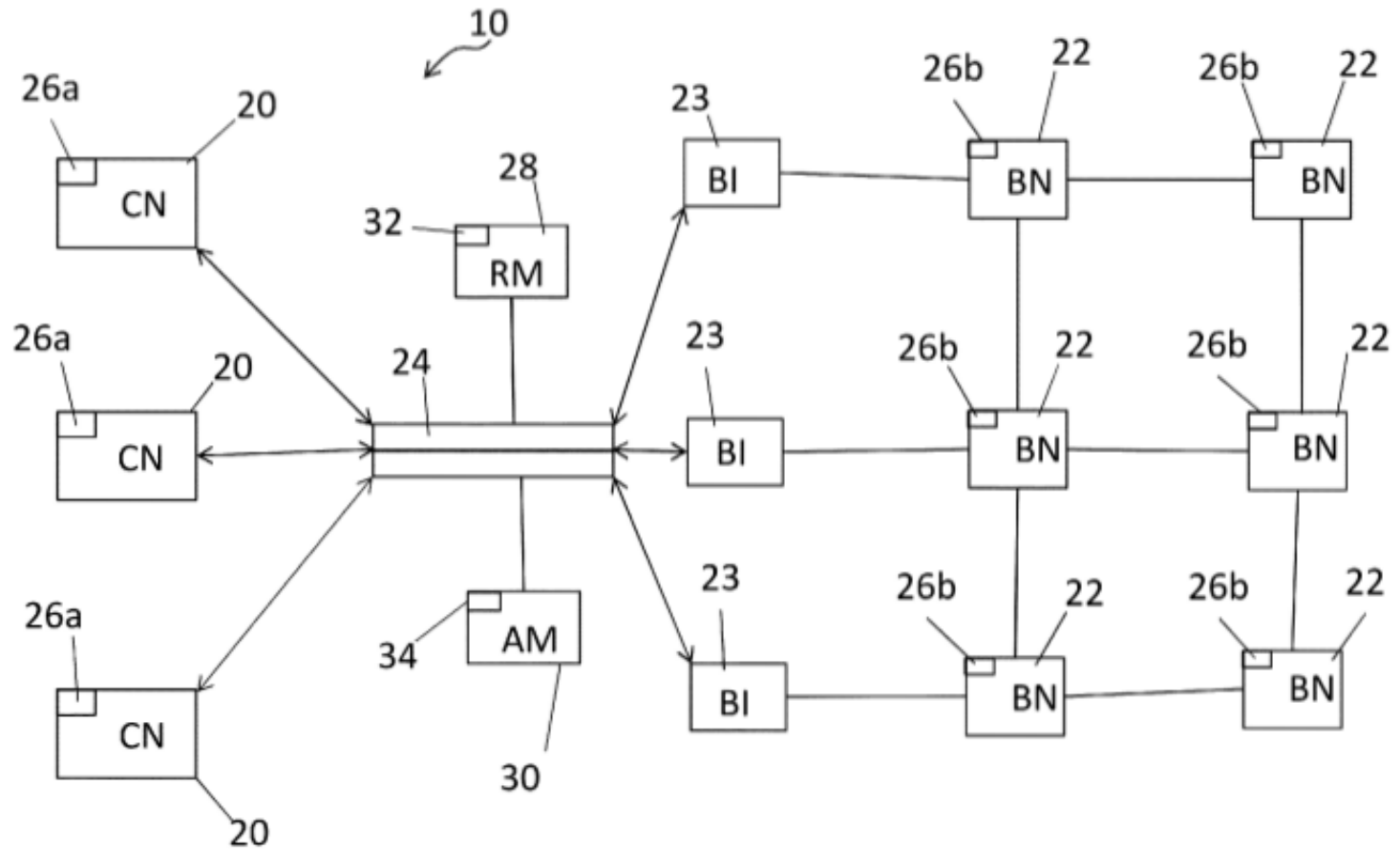
Primary Examiner — Sisley N Kim

(74) Attorney, Agent, or Firm — Marshall, Gerstein &
Borun LLP

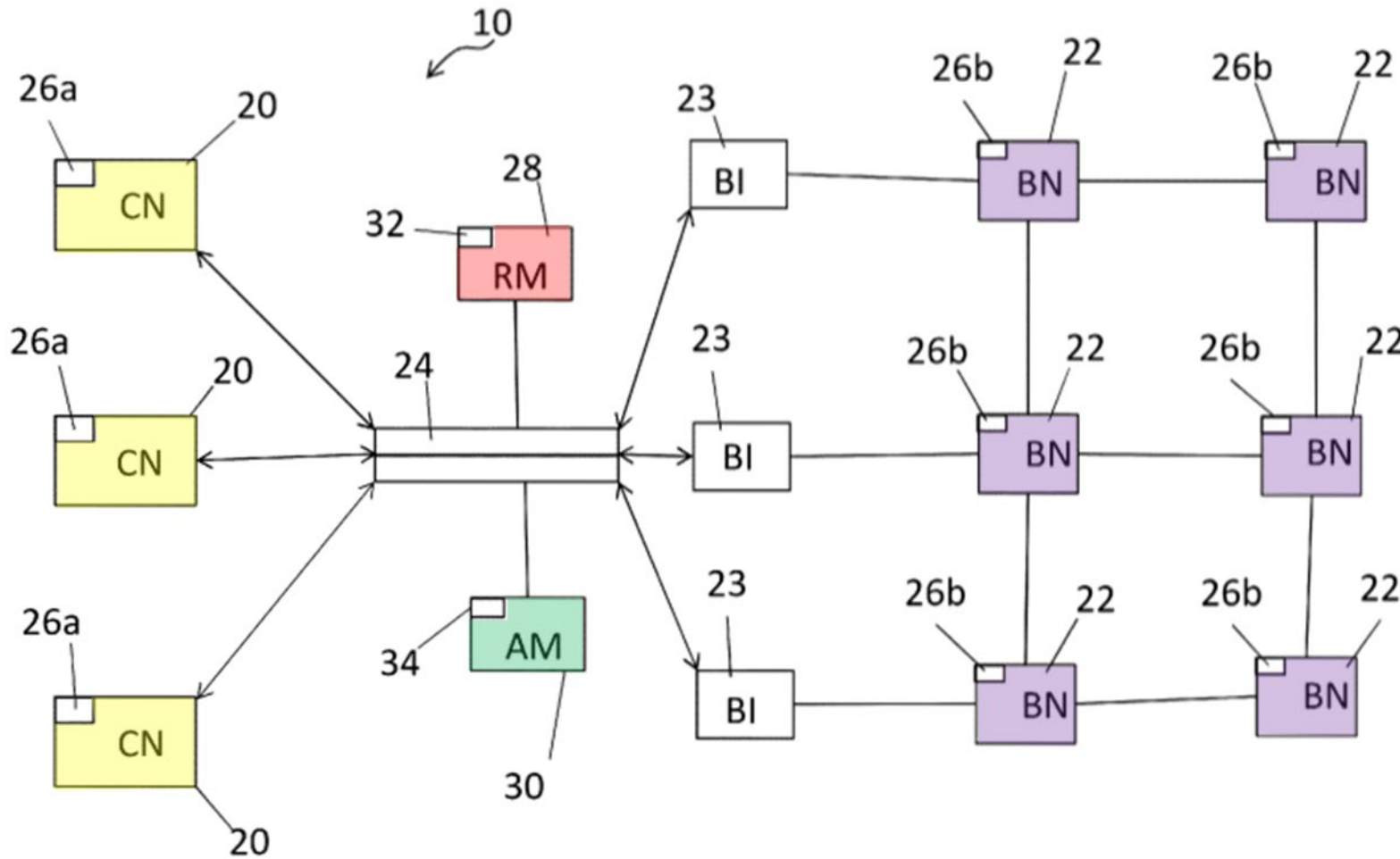
(57) ABSTRACT

The present invention provides a method of operating a heterogeneous computing system comprising a plurality of computation nodes and a plurality of booster nodes, at least one of the plurality of computation nodes and plurality of booster nodes being arranged to compute a computation task, the computation task comprising a plurality of sub-tasks, wherein in a first computing iteration, the plurality of sub-tasks are assigned to and processed by ones of the plurality of computation nodes and booster nodes in a first distribution; and information relating to the processing of the plurality of sub-tasks by the plurality of computation nodes and booster nodes is used to generate a further distribution of the sub-tasks between the computation nodes and booster node for processing thereby in a further computing iteration.

10 Claims, 1 Drawing Sheet



The '442 Patent



- Computation nodes (“CN”) 20 with CN daemons 26a
- Booster nodes (“BN”) 22 with BN daemons 26b
- Resource manager (“RM”) 28
- Application manager (“AM”) 30

*All highlighting and emphasis added unless otherwise noted.

'442 Patent: Tasks and Sub-Tasks

(12) United States Patent Lippert et al.

(10) Patent No.: US 11,537,442 B2
(45) Date of Patent: Dec. 27, 2022

(54) APPLICATION RUNTIME DETERMINED DYNAMICAL ALLOCATION OF HETEROGENEOUS COMPUTE RESOURCES

(71) Applicant: **ParTec Cluster Competence Center
GmbH**, Munich (DE)

(72) Inventors: **Thomas Lippert**, Aschaffenburg (DE);
Bernhard Frohwitter, Munich (DE)

(73) Assignee: **ParTec Cluster Competence Center
GmbH**, Munich (DE)

(*) Notice: Subject to any disclaimer, the term of this
patent is extended or adjusted under 35
U.S.C. 154(b) by 35 days.

(21) Appl. No.: **16/963,749**

(22) PCT Filed: **Jan. 23, 2019**

(86) PCT No.: **PCT/EP2019/051615**

§ 371 (c)(1),
(2) Date: **Jul. 21, 2020**

(87) PCT Pub. No.: **WO2019/145354**

PCT Pub. Date: **Aug. 1, 2019**

(65) Prior Publication Data

US 2021/0096923 A1 Apr. 1, 2021

(30) Foreign Application Priority Data

Jan. 23, 2018 (EP) 18152903

(51) Int. CL **G06F 9/50**
(2006.01)

(52) U.S. CL **G06F 9/5061** (2013.01); **G06F 2209/509**
(2013.01)

(58) Field of Classification Search
CPC G06F 9/5061; G06F 2209/509
See application file for complete search history.

(56) References Cited

U.S. PATENT DOCUMENTS

2012/0317168 A1* 12/2012 Driesen G06Q 30/06

709/202

2013/0282787 A1* 10/2013 Lippert H04L 29/08081

709/201

FOREIGN PATENT DOCUMENTS

EP 2442228 A1 4/2012

WO WO-2018065530 A1 4/2018

OTHER PUBLICATIONS

International Search Report for International Application No. PCT/
EP2019/0518615, dated Apr. 29, 2019.

(Continued)

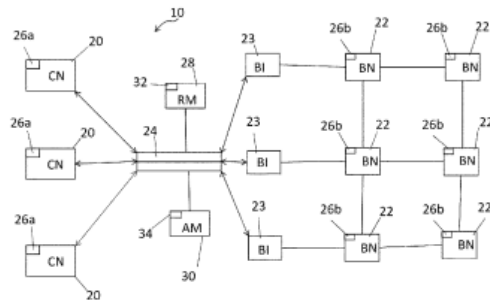
Primary Examiner — Sisley N Kim

(74) Attorney, Agent, or Firm — Marshall, Gerstein &
Borun LLP

(57) ABSTRACT

The present invention provides a method of operating a heterogeneous computing system comprising a plurality of computation nodes and a plurality of booster nodes, at least one of the plurality of computation nodes and plurality of booster nodes being arranged to compute a computation task, the computation task comprising a plurality of sub-tasks, wherein in a first computing iteration, the plurality of sub-tasks are assigned to and processed by ones of the plurality of computation nodes and booster nodes in a first distribution; and information relating to the processing of the plurality of sub-tasks by the plurality of computation nodes and booster nodes is used to generate a further distribution of the sub-tasks between the computation nodes and booster node for processing thereby in a further computing iteration.

10 Claims, 1 Drawing Sheet



A job to be computed by the system may comprise a number of **tasks** some of which or all may be repeated a number of times during the execution of the job. For

EX1001 ('442 Patent) at 3:44-46

The **tasks** may comprise a number of **sub-tasks** or kernels. Each of these sub-tasks may be more or less **suitable for processing** by one or more of the **computation nodes** or one or more of the **boosters**. In particular, the scalability of the

EX1001 ('442 Patent) at 3:50-53

'442 Patent: Optimal Further Division

(12) United States Patent Lippert et al.

(10) Patent No.: US 11,537,442 B2
(45) Date of Patent: Dec. 27, 2022

(54) APPLICATION RUNTIME DETERMINED
DYNAMICAL ALLOCATION OF
HETEROGENEOUS COMPUTE RESOURCES

(71) Applicant: ParTec Cluster Competence Center
GmbH, Munich (DE)

(72) Inventors: Thomas Lippert, Aschaffenburg (DE);
Bernhard Frohwitter, Munich (DE)

(73) Assignee: ParTec Cluster Competence Center
GmbH, Munich (DE)

(*) Notice: Subject to any disclaimer, the term of this
patent is extended or adjusted under 35
U.S.C. 154(b) by 35 days.

(21) Appl. No.: 16/963,749

(22) PCT Filed: Jan. 23, 2019

(86) PCT No.: PCT/EP2019/051615

§ 371 (c)(1),
(2) Date: Jul. 21, 2020

(87) PCT Pub. No.: WO2019/145354

PCT Pub. Date: Aug. 1, 2019

(65) Prior Publication Data

US 2021/0096923 A1 Apr. 1, 2021

(30) Foreign Application Priority Data

Jan. 23, 2018 (EP) 18152903

(51) Int. CL. G06F 9/50 (2006.01)

(52) U.S. CL. CPC G06F 9/5061 (2013.01); G06F 2209/509
(2013.01)

(58) Field of Classification Search
CPC G06F 9/5061; G06F 2209/509
See application file for complete search history.

(56) References Cited

U.S. PATENT DOCUMENTS

2012/0317168 A1* 12/2012 Driesen G06Q 30/06

709/202

2013/0282787 A1* 10/2013 Lippert H04L 29/08081

709/201

FOREIGN PATENT DOCUMENTS

EP 2442228 A1 4/2012

WO WO-2018065530 A1 4/2018

OTHER PUBLICATIONS

International Search Report for International Application No. PCT/

EP2019/0518615, dated Apr. 29, 2019.

(Continued)

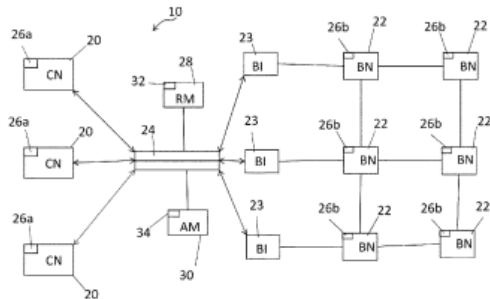
Primary Examiner — Sisley N Kim

(74) Attorney, Agent, or Firm — Marshall, Gerstein &
Borun LLP

(57) ABSTRACT

The present invention provides a method of operating a heterogeneous computing system comprising a plurality of computation nodes and a plurality of booster nodes, at least one of the plurality of computation nodes and plurality of booster nodes being arranged to compute a computation task, the computation task comprising a plurality of sub-tasks, wherein in a first computing iteration, the plurality of sub-tasks are assigned to and processed by ones of the plurality of computation nodes and booster nodes in a first distribution; and information relating to the processing of the plurality of sub-tasks by the plurality of computation nodes and booster nodes is used to generate a further distribution of the sub-tasks between the computation nodes and booster node for processing thereby in a further computing iteration.

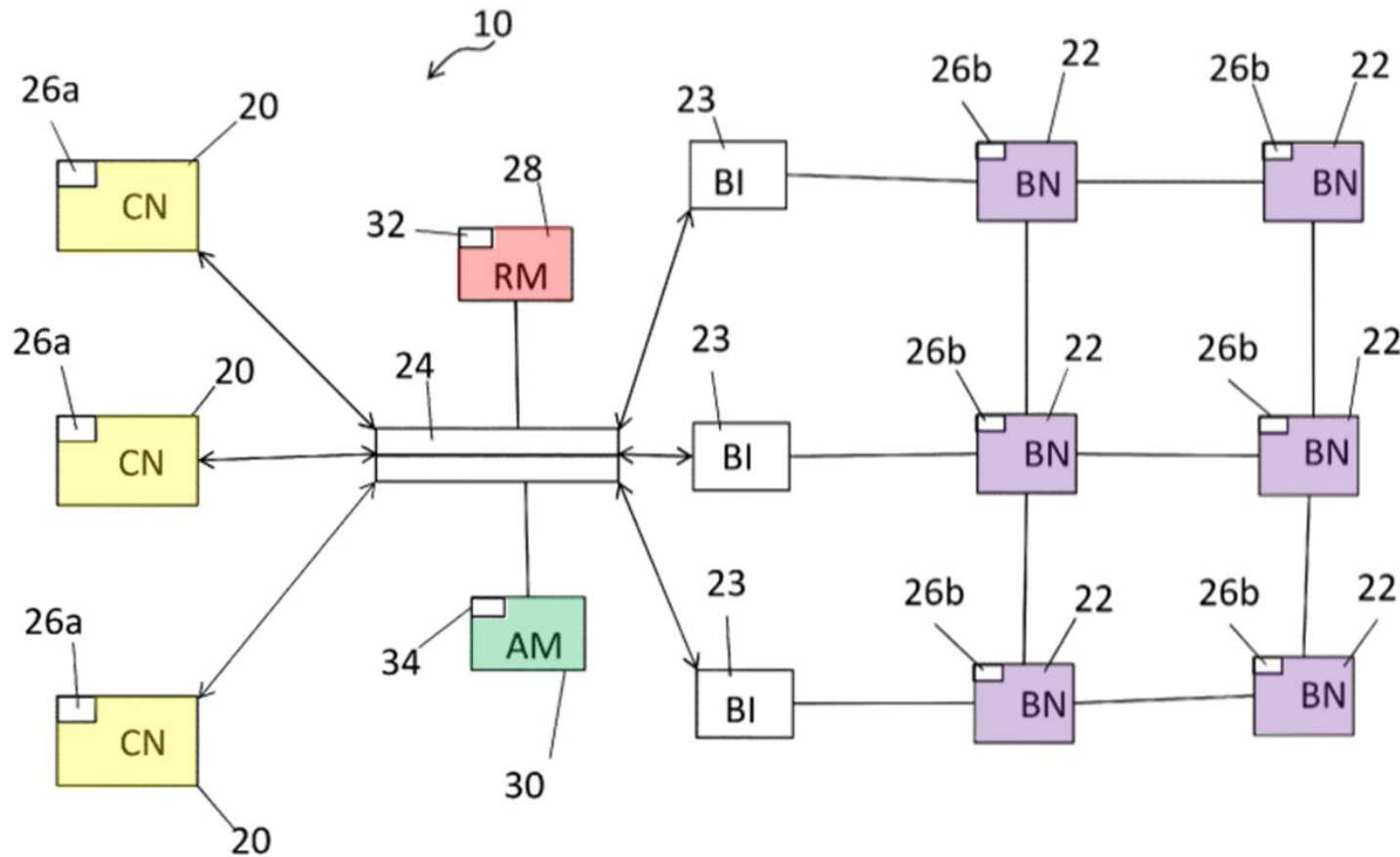
10 Claims, 1 Drawing Sheet



If a task is computed with a first division of sub-tasks between a computation node and a booster, it is possible that such a division is not an optimal division for the computation of the task. Certain sub-tasks which were assigned to a booster on the first iteration might in fact not be suitable for processing by the booster such that a processing of the sub-tasks by a computation node rather than a booster might optimize the computation of the task as a whole. Accordingly, for a second and possible further iterations, if needed, of the task with an altered second and/or further sub-task distribution might improve the efficiency of the computation of the task.

EX1001 ('442 Patent) at 3:59-4:3

'442 Patent: Further Sub-Task Adjustments



Accordingly, the system 10 includes a mechanism whereby each of the computation nodes and the cluster nodes are arranged such that daemons 26a, 26b and 32 feed back information to the daemon 34 relating to the processing of sub-tasks and a current state of the respective processing entity. The daemon 34 uses the information provided by the daemons 26a, 26b and 32 to determine whether the distribution of sub-tasks to computation nodes and boosters could be adjusted to optimize or adapt the computation of the task for a subsequent iteration. The resource manager can also reassign computation nodes and boosters to each other, in addition to the adjusted distribution of tasks.

EX1001 ('442 Patent) at 4:4-15

For each iteration, the daemons operating in the computation nodes and the boosters report status information to the application manager and the resource manager enabling the calculation of subsequent iterations to be optimized by a further adjustments to the allocation of sub-tasks to computation nodes and boosters.

EX1001 ('442 Patent) at 4:32-38

Dynamically Rearrange Sub-Task Assignments During Runtime

(12) **United States Patent**
Lippert et al.

(10) **Patent No.:** US 11,537,442 B2
(45) **Date of Patent:** Dec. 27, 2022

(54) **APPLICATION RUNTIME DETERMINED DYNAMIC ALLOCATION OF HETEROGENEOUS COMPUTE RESOURCES**

(71) Applicant: **ParTec Cluster Competence Center GmbH**, Munich (DE)

(72) Inventors: **Thomas Lippert**, Aschaffenburg (DE); **Bernhard Frohwitter**, Munich (DE)

(73) Assignee: **ParTec Cluster Competence Center GmbH**, Munich (DE)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 35 days.

(21) Appl. No.: **16/963,749**

(22) PCT Filed: **Jan. 23, 2019**

(86) PCT No.: **PCT/EP2019/051615**
§ 371 (c)(1),
(2) Date: **Jul. 21, 2020**

(87) PCT Pub. No.: **WO2019/145354**
PCT Pub. Date: **Aug. 1, 2019**

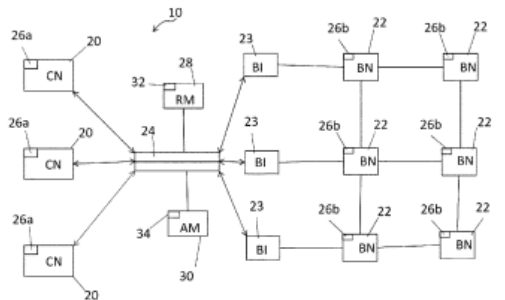
(65) **Prior Publication Data**
US 2021/0096923 A1 Apr. 1, 2021

(30) **Foreign Application Priority Data**
Jan. 23, 2018 (EP) 18152903

(51) **Int. CL**
G06F 9/50 (2006.01)

(52) **U.S. CL**
CPC **G06F 9/5061** (2013.01); **G06F 2209/509** (2013.01)

(58) **Field of Classification Search**
CPC **G06F 9/5061**; **G06F 2209/509**
See application file for complete search history.



An important aspect of the invention comes with the recognition that a mapping, tailoring of computation tasks, and sub-tasks onto a computer hierarchy may not provide sufficient use of the inherent flexibility and adaptability of the computer architecture, as shown as an example according to WO 2012/049247 A1. Therefore, the present invention integrates tailoring the application task most sufficiently, as an example according to WO 2017/0262319 A1, as well as configuring dynamically the computation nodes and booster nodes to each other, and eventually dynamically rearranging the mapping of the computation tasks during runtime, based on the information provided by daemons about the efficiency of the execution of the computation task, and dynamically reassigning the computation nodes and computation boosters to each other.

EX1001 ('442 Patent) at 5:10-24

'442 Patent Independent Claims

1. A method of operating a heterogeneous computing system comprising a plurality of computation nodes and a plurality of booster nodes, at least one of the plurality of computation nodes and a plurality of booster nodes being arranged to compute a computation task, the computation task comprising a plurality of sub-tasks, the method comprising:

in a first computing iteration, assigning and processing the plurality of sub-tasks by at least a portion of the plurality of computation nodes and at least a portion of the plurality of booster nodes in a first distribution; and generating, using information relating to the processing of the plurality of sub-tasks by at least the portion of the plurality of computation nodes and at least the portion of the plurality of booster nodes, a further distribution of the plurality of sub-tasks between the plurality of computation nodes and the plurality of booster nodes for processing thereby in a further computing iteration.

9. A heterogeneous computing system comprising:
a plurality of computation nodes and a plurality of booster nodes for computing one or more tasks comprising multiple sub-tasks;
a communication interface connecting the plurality of computation nodes with each other and the plurality of booster nodes;
a resource manager for assigning at least a portion of the plurality of booster nodes and at least a portion of the plurality of computation nodes to each other for the computing of the one or more tasks in a first computing iteration; and
an application manager configured to receive information from daemons operating in at least the portion of the plurality of computation nodes and at least the portion of the plurality of booster nodes to update a distribution of the multiple sub-tasks between the plurality of computation nodes and the plurality of booster nodes in a further computing iteration.

Outline



The '442 Patent



Asserted Prior Art Overview



Petition Ground Deficiencies

Asserted Prior Art

	Reference(s)	Basis	Claims
Ground 1	Lippert (EX1004) and Budenske (EX1005)	35 U.S.C. § 103	1-10
Ground 2	Lippert, Budenske, and Kambatla (EX1006)	35 U.S.C. § 103	2-5, 8-10

Pet. at 2

Lippert Published Application

(19) **United States**

(12) **Patent Application Publication**
Lippert

(10) Pub. No.: **US 2013/0282787 A1**

(43) Pub. Date: **Oct. 24, 2013**

(54) **COMPUTER CLUSTER ARRANGEMENT
FOR PROCESSING A COMPUTATION TASK
AND METHOD FOR OPERATION THEREOF**

Publication Classification

(51) Int. Cl.
H04L 29/08 (2006.01)
(52) U.S. Cl.
CPC *H04L 29/08081* (2013.01)
USPC 709/201

(71) Applicant: **ParTec Cluster Competence Center
GmbH, Munchen (DE)**

(72) Inventor: **Thomas Lippert, Aschaffenburg (DE)**

(21) Appl. No.: **13/861,429**

(22) Filed: **Apr. 12, 2013**

Related U.S. Application Data

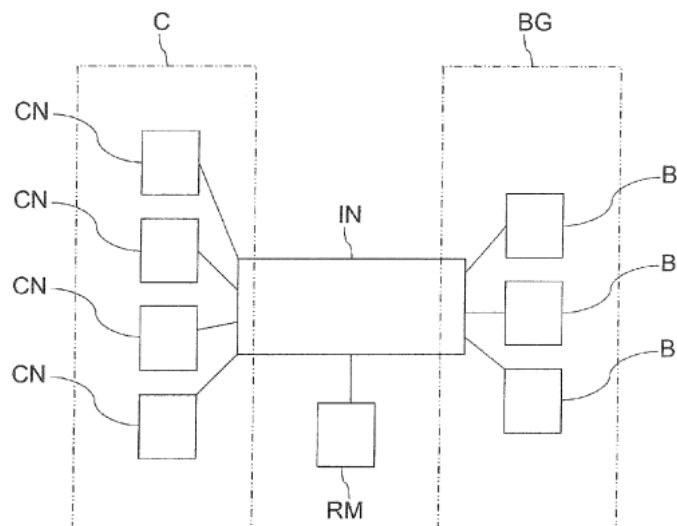
(63) Continuation of application No. PCT/EP2011/
067888, filed on Oct. 13, 2011.

Foreign Application Priority Data

Oct. 13, 2010 (EP) 10187436.0

ABSTRACT

The present invention is directed to a computer cluster arrangement and a method for operation of the introduced computer cluster arrangement. The computer cluster arrangement comprises computation nodes CN, which dynamically outsource specific computation tasks to boosters B. Therefore, an assignment technique of boosters B to computation nodes CN is introduced. The assignment takes dynamically place at runtime. The present invention finds application in high performance cluster technology.



(71) Applicant: **ParTec Cluster Competence Center
GmbH, Munchen (DE)**

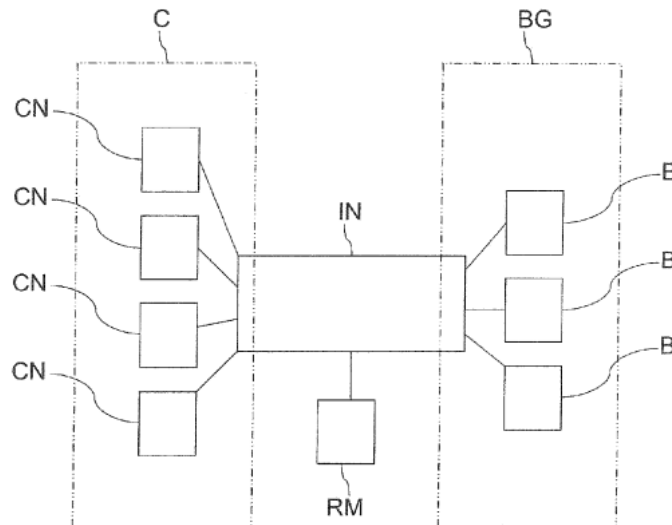
(72) Inventor: **Thomas Lippert, Aschaffenburg (DE)**

[0001] This application is a continuation of PCT Application No. PCT/EP2011/067888 filed 13, Oct. 2011, which was published in the English language on 19, Apr. 2012 as PCT Publication No. WO 2012/049247. The PCT Application claims priority to EP Application No. 10187436.0 filed 13, Oct. 2010. These documents are hereby incorporated herein by reference.

(19) **United States**
 (12) **Patent Application Publication** (10) **Pub. No.: US 2013/0282787 A1**
 Lippert (43) **Pub. Date: Oct. 24, 2013**

(54) **COMPUTER CLUSTER ARRANGEMENT FOR PROCESSING A COMPUTATION TASK AND METHOD FOR OPERATION THEREOF**
 (71) Applicant: **ParTec Cluster Competence Center GmbH, Munchen (DE)**
 (72) Inventor: **Thomas Lippert, Aschaffenburg (DE)**
 (21) Appl. No.: **13/861,429**
 (22) Filed: **Apr. 12, 2013**
 (51) **Int. Cl.** **H04L 29/08** (2006.01)
 (52) **U.S. Cl.** **CPC H04L 29/08081** (2013.01)
USPC 709/201

(57) **ABSTRACT**
 The present invention is directed to a computer cluster arrangement and a method for operation of the introduced computer cluster arrangement. The computer cluster arrangement comprises computation nodes CN, which dynamically outsource specific computation tasks to boosters B. Therefore, an assignment technique of boosters B to computation nodes CN is introduced. The assignment takes dynamically place at runtime. The present invention finds application in high performance cluster technology.
 (63) Continuation of application No. PCT/EP2011/067888, filed on Oct. 13, 2011.
 (30) **Foreign Application Priority Data**
 Oct. 13, 2010 (EP) 10187436.0



[0013] In this computer cluster arrangement acceleration functionality is being provided by independent boosters. The described computer cluster arrangement allows a loose coupling of those boosters to computation nodes, which may also be referred to as compute nodes. Hence, a sharing of accelerators, here in form of boosters, by computation nodes is feasible. For an assignment of a booster to a computation node a resource manager, in form of a resource manager module or resource manager node, may be provided. The resource manager may establish a static assignment at start of a processing of a computation task. Alternatively or additionally it may establish a dynamic assignment at runtime, which means during processing of the computation task.

'442 Patent Relationship to Lippert

(12) United States Patent Lippert et al.

(10) Patent No.: US 11,537,442 B2
(45) Date of Patent: Dec. 27, 2022

(54) APPLICATION RUNTIME DETERMINED
DYNAMICAL ALLOCATION OF
HETEROGENEOUS COMPUTE RESOURCES

(71) Applicant: ParTec Cluster Competence Center
GmbH, Munich (DE)

(72) Inventors: Thomas Lippert, Aschaffenburg (DE);
Bernhard Frohwitter, Munich (DE)

(73) Assignee: ParTec Cluster Competence Center
GmbH, Munich (DE)

(*) Notice: Subject to any disclaimer, the term of this
patent is extended or adjusted under 35
U.S.C. 154(b) by 35 days.

(21) Appl. No.: 16/963,749

(22) PCT Filed: Jan. 23, 2019

(86) PCT No.: PCT/EP2019/051615

§ 371 (c)(1),
(2) Date: Jul. 21, 2020

(87) PCT Pub. No.: WO2019/145354

PCT Pub. Date: Aug. 1, 2019

(65) Prior Publication Data

US 2021/0096923 A1 Apr. 1, 2021

(30) Foreign Application Priority Data

Jan. 23, 2018 (EP) 18152903

(51) Int. CL: G06F 9/50 (2006.01)

(52) U.S. CL: CPC G06F 9/5061 (2013.01); G06F 2209/509
(2013.01)

(58) Field of Classification Search
CPC G06F 9/5061; G06F 2209/509
See application file for complete search history.

(56) References Cited

U.S. PATENT DOCUMENTS

2012/0317168 A1* 12/2012 Driesen G06Q 30/06

2013/0282787 A1* 10/2013 Lippert H04L 29/08081

709/202
709/201

FOREIGN PATENT DOCUMENTS

EP 2442228 A1 4/2012

WO WO-2018065530 A1 4/2018

OTHER PUBLICATIONS

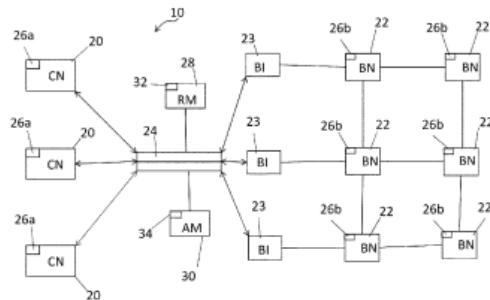
International Search Report for International Application No. PCT/
EP2019/0518615, dated Apr. 29, 2019.
(Continued)

Primary Examiner — Sisley N Kim
(74) Attorney, Agent, or Firm — Marshall, Gerstein &
Borun LLP

(57) ABSTRACT

The present invention provides a method of operating a heterogeneous computing system comprising a plurality of computation nodes and a plurality of booster nodes, at least one of the plurality of computation nodes and plurality of booster nodes being arranged to compute a computation task, the computation task comprising a plurality of sub-tasks, wherein in a first computing iteration, the plurality of sub-tasks are assigned to and processed by ones of the plurality of computation nodes and booster nodes in a first distribution; and information relating to the processing of the plurality of sub-tasks by the plurality of computation nodes and booster nodes is used to generate a further distribution of the sub-tasks between the computation nodes and booster node for processing thereby in a further computing iteration.

10 Claims, 1 Drawing Sheet



(71) Applicant: ParTec Cluster Competence Center
GmbH, Munich (DE)

(72) Inventors: Thomas Lippert, Aschaffenburg (DE);
Bernhard Frohwitter, Munich (DE)

The present invention is a development of the system described in earlier application WO 2012/049247 A1, WO '247, which describes a cluster computer architecture comprising a plurality of computation nodes and a plurality of boosters connected together via a communication interface. A resource manager is responsible for assigning one or more of the boosters and the computation nodes to each other in a dynamic manner during runtime. An example for such dynamic process management is described in Clauss et al. "Dynamic Process Management with Allocation-internal Co-Scheduling towards Interactive Supercomputing", COSH 2016 Jan. 19, Prague, CZ, incorporated herein by reference for all purposes. While the arrangement of WO '247 provides a flexible arrangement for assigning boosters to computation nodes, it does not address how to distribute tasks between computation nodes and boosters.

Lippert and the '442 Patent

Docket No.: 29959/55804
(PATENT)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:
Thomas Lippert et al.

Application No.: 16/963,749

Filed: July 21, 2020

For: APPLICATION RUNTIME DETERMINED
DYNAMICAL ALLOCATION OF
HETEROGENEOUS COMPUTE RESOURCES

Confirmation No.: 6322

Art Unit: 2196

Examiner: D. G. Muller

AMENDMENT IN RESPONSE

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

Dear Sir:

In response to the
identified U.S. patent app

Amendments to
page 2 of this paper.

Remarks/Argument

Lippert describes “a computer cluster arrangement and a method for operation of the introduced computer cluster arrangement” (see Abstract). In particular, Lippert discloses that the computer cluster arrangement solves “the parts of the computation task in parallel or in succession” (see paragraph [0016]). For example, a computation task may be divided into a first part and a second part and the second part may be computed parallel to the first part or subsequent to the first part. However, in contrast, amended claims 1 and 9 recite that the plurality of sub-task are processed in **multiple computing iterations**.

EX1002 ('442 Patent Pros. History) at 174

A Method for the On-Line Use of Off-Line Derived Remappings of Iterative Automatic Target Recognition Tasks onto a Particular Class of Heterogeneous Parallel Platforms

JOHN R. BUDENSKÉ AND RANGA S. RAMANUJAN
Architecture Technology Corporation, P.O. Box 24344, Minneapolis, MN 55424, USA, {budenske,
ranga}@atcorp.com

HOWARD JAY SIEGEL
Parallel Processing Laboratory, School of Electrical and Computer Engineering, Purdue University, West
Lafayette, IN 47907-1285, USA, hj@purdue.edu

Abstract. This study focuses on a particular application domain (iterative automatic target recognition tasks) and an associated specific class of dedicated heterogeneous parallel hardware platforms. For the computational environment considered, a methodology is presented for the on-line operating system to decide heuristically whether to perform a remapping of the application onto the platform based on information generated from input data by the application during execution. If the decision is to remap, the operating system will be able to select a mapping, which is appropriate for the given state of the application, from a stored set of mappings that were previously derived with an off-line heuristic.

Keywords: Algorithm mapping, automatic target recognition, genetic algorithms, heterogeneous computing, real-time processing

1. Introduction

Heterogeneous computing (HC) covers a great variety of situations (e.g., see [9], [17], [22], [23], [26]). This study focuses on a particular application domain (iterative automatic target recognition (ATR) tasks) and an associated specific class of dedicated heterogeneous parallel hardware platforms. The contribution of this paper is that, for the computational environment considered, it presents a methodology for real-time on-line input-data dependent remapping of the application subtasks to the processors in the heterogeneous parallel hardware platform using a previously stored off-line statically determined mapping (i.e., a matching of application subtasks to processors and a scheduling for the execution order of these subtasks). That is, the operating system will be able to make a heuristically-based decision during the execution of the application whether to perform a remapping based on information generated by the application from its input data. If the decision is to remap, the operating system will be able to select a previously heuristically derived and stored mapping that is appropriate for the given state of the application (e.g., the number of objects it is currently tracking).

Abstract. This study focuses on a particular application domain (iterative automatic target recognition tasks) and an associated specific class of dedicated heterogeneous parallel hardware platforms. For the computational environment considered, a methodology is presented for the on-line operating system to decide heuristically whether to perform a remapping of the application onto the platform based on information generated from input data by the application during execution. If the decision is to remap, the operating system will be able to select a mapping, which is appropriate for the given state of the application, from a stored set of mappings that were previously derived with an off-line heuristic.

Thus, the goal of this paper is to introduce the ideas underlying the design and use of the HC Kernel for a particular class of application computational structure and a particular type of target hardware platform. The performance of the HC Kernel in comparison to other methods will depend upon the exact application and exact platform under consideration. The technique presented here is offered to future system designers to evaluate as an alternative for their particular situations.

For simplicity, it is assumed that if an implementation of a given subtask uses multiple processors, all processors will be of the same type. Given this and the symmetry property of the inter-processor communications among processors of the same type, the expected execution time of a particular multiprocessor implementation of a subtask is independent of which fixed-size subset of the processors of a given type are assigned to execute the subtask.

EX1005 (Budenske) at 387-88, 390, 392

Kambatla

(19) **United States**

(12) **Patent Application Publication** (10) **Pub. No.: US 2018/0074855 A1**
Kambatla (43) **Pub. Date: Mar. 15, 2018**

(54) **UTILIZATION-AWARE RESOURCE SCHEDULING IN A DISTRIBUTED COMPUTING CLUSTER**

(52) **U.S. CL.**
 CPC *G06F 9/4881* (2013.01); *G06F 9/5083* (2013.01); *G06F 2209/483* (2013.01); *G06F 2209/5021* (2013.01); *G06F 9/5038* (2013.01)

(71) Applicant: **Cloudera, Inc.**, Palo Alto, CA (US)

(72) Inventor: **Karthik Kambatla**, Mountain View, CA (US)

(57) **ABSTRACT**

(21) Appl. No.: **15/595,713**

(22) Filed: **May 15, 2017**

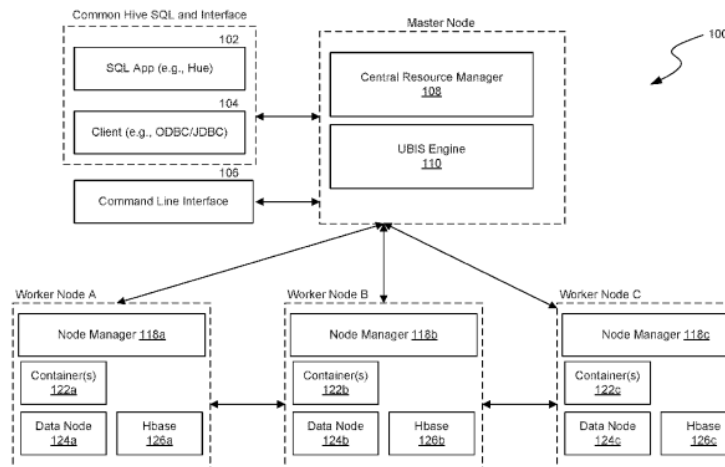
Related U.S. Application Data

(60) Provisional application No. 62/394,660, filed on Sep. 14, 2016.

Publication Classification

(51) **Int. CL.**
G06F 9/48 (2006.01)
G06F 9/50 (2006.01)

Embodiments are disclosed for a utilization-aware approach to cluster scheduling, to address this resource fragmentation and to improve cluster utilization and job throughput. In some embodiments a resource manager at a master node considers actual usage of running tasks and schedules opportunistic work on underutilized worker nodes. The resource manager monitors resource usage on these nodes and pre-empts opportunistic containers in the event this over-subscription becomes untenable. In doing so, the resource manager effectively utilizes wasted resources, while minimizing adverse effects on regularly scheduled tasks.



TECHNICAL FIELD

[0002] The present disclosure generally relates to distributed computing clusters, and more particularly to the allocation and management of computing resources in distributed computing clusters.

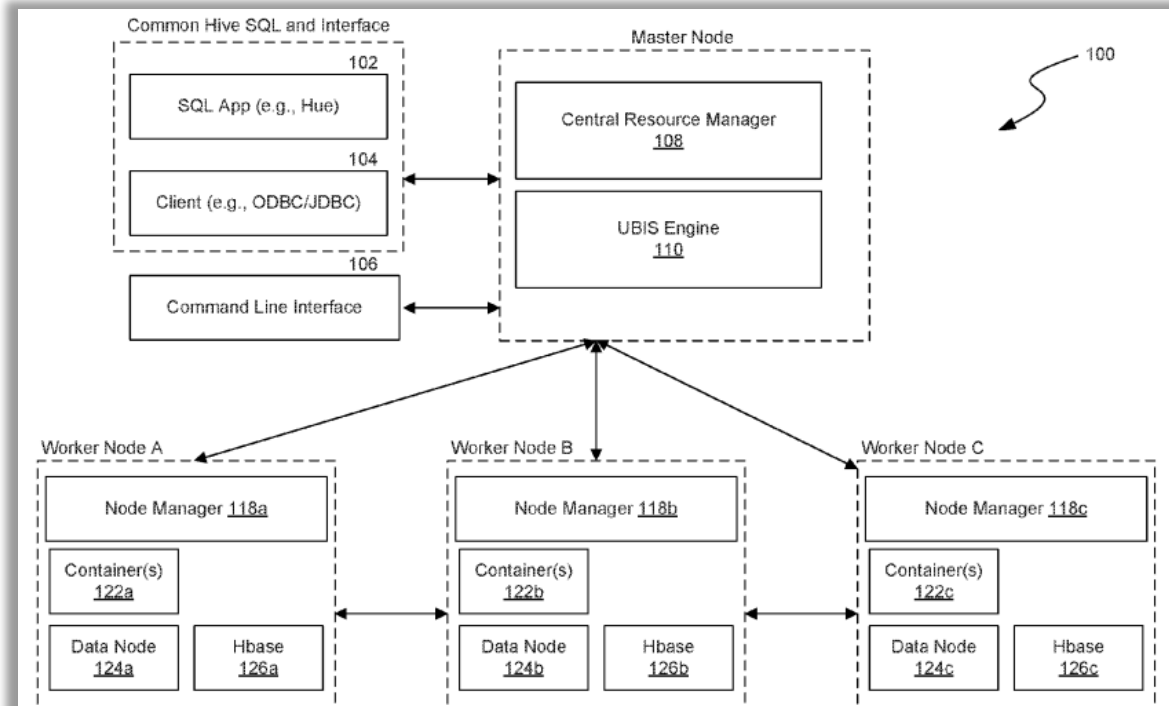


FIG. 1

EX1006 (Kambatla) at Front Page, [0002], Fig. 1

Outline



The '442 Patent



Asserted Prior Art Overview



Petition Ground Deficiencies

Outline of Petition Deficiencies

Petition Deficiencies

- **Claims 1 and 9 (All Grounds)**
 - **Further Distribution of Sub-Tasks between CN's and BN's**
 - Budenske not Enabling
 - **Generate Further Distribution Using Information Related to Processing**
 - **Budenske's Dynamic Parameters Not Process-Related Information**
 - **Petition Fails to Address Petitioner's District Court Constructions**
- **Claim 3 (All Grounds) — Dynamically Alters Further Distribution**
- **Claims 5 and 9 (All Grounds) — CN & BN Daemons provide claimed information**
- **All Claims (All Grounds) — No Motivation to Combine**

Outline of Petition Deficiencies

Petition Deficiencies

- **Claims 1 and 9 (All Grounds)**
 - **Further Distribution of Sub-Tasks between CN's and BN's**
 - Budenske not Enabling
 - **Generate Further Distribution Using Information Related to Processing**
 - **Budenske's Dynamic Parameters Not Process-Related Information**
 - **Petition Fails to Address Petitioner's District Court Constructions**
- **Claim 3 (All Grounds) — Dynamically Alters Further Distribution**
- **Claims 5 and 9 (All Grounds) — CN & BN Daemons provide claimed information**
- **All Claims (All Grounds) — No Motivation to Combine**

Further/Updated Distribution of Sub-Tasks

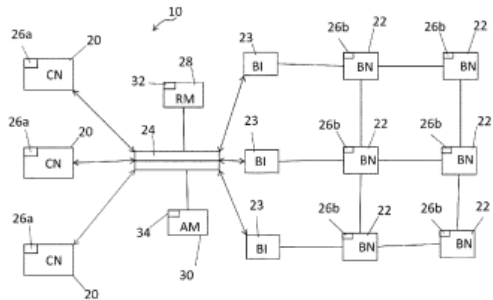
1. A method of operating a heterogeneous computing system comprising a plurality of computation nodes and a plurality of booster nodes, at least one of the plurality of computation nodes and a plurality of booster nodes being arranged to compute a computation task, the computation task comprising a plurality of sub-tasks, the method comprising:

in a first computing iteration, assigning and processing the plurality of sub-tasks by at least a portion of the plurality of computation nodes and at least a portion of the plurality of booster nodes in a first distribution; and generating, using information relating to the processing of the plurality of sub-tasks by at least the portion of the plurality of computation nodes and at least the portion of the plurality of booster nodes, a further distribution of the plurality of sub-tasks between the plurality of computation nodes and the plurality of booster nodes for processing thereby in a further computing iteration.

9. A heterogeneous computing system comprising:
a plurality of computation nodes and a plurality of booster nodes for computing one or more tasks comprising multiple sub-tasks;
a communication interface connecting the plurality of computation nodes with each other and the plurality of booster nodes;
a resource manager for assigning at least a portion of the plurality of booster nodes and at least a portion of the plurality of computation nodes to each other for the computing of the one or more tasks in a first computing iteration; and
an application manager configured to receive information from daemons operating in at least the portion of the plurality of computation nodes and at least the portion of the plurality of booster nodes to update a distribution of the multiple sub-tasks between the plurality of computation nodes and the plurality of booster nodes in a further computing iteration.

'442 Patent: Optimal Further Sub-Task Division

(12)	United States Patent	(10) Patent No.:	US 11,537,442 B2
	Lippert et al.	(45) Date of Patent:	Dec. 27, 2022
(54)	APPLICATION RUNTIME DETERMINED DYNAMICAL ALLOCATION OF HETEROGENEOUS COMPUTE RESOURCES	(56)	References Cited
			U.S. PATENT DOCUMENTS
(71)	Applicant: ParTec Cluster Competence Center GmbH , Munich (DE)	2012/0317168 A1 * 12/2012	Driesen G06Q 30/06 709:202
(72)	Inventors: Thomas Lippert , Aschaffenburg (DE); Bernhard Frohwitter , Munich (DE)	2013/0282787 A1 * 10/2013	Lippert H04L 29/08081 709:201
(73)	Assignee: ParTec Cluster Competence Center GmbH , Munich (DE)		
(*)	Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 35 days.		FOREIGN PATENT DOCUMENTS
		EP 2442228 A1 4/2012	
		WO WO-2018065530 A1 4/2018	
			OTHER PUBLICATIONS
			International Search Report for International Application No. PCT/EP2019/0518615, dated Apr. 29, 2019.
(21)	Appl. No.: 16/963,749		(Continued)
(22)	PCT Filed: Jan. 23, 2019		
(26)	PCT No.: PCT/EP2019/051615		<i>Primary Examiner</i> — Sisley N Kim
	§ 371 (c)(1), (2) Date: Jul. 21, 2020		(74) <i>Attorney, Agent, or Firm</i> — Marshall, Gerstein & Borum LLP
(87)	PCT Pub. No.: WO2019/145354	(57)	ABSTRACT
	PCT Pub. Date: Aug. 1, 2019		The present invention provides a method of operating a heterogeneous computing system comprising a plurality of computation nodes and a plurality of booster nodes, at least one of the plurality of computation nodes and plurality of booster nodes being arranged to compute a computation task, the computation task comprising a plurality of sub-tasks, wherein in a first computing iteration, the plurality of sub-tasks are assigned to and processed by ones of the plurality of computation nodes and booster nodes in a first distribution; and information relating to the processing of the plurality of sub-tasks by the plurality of computation nodes and booster nodes is used to generate a further distribution of the sub-tasks between the computation nodes and booster node for processing thereby in a further computing iteration.
(65)	Prior Publication Data		
	US 2021/0096923 A1 Apr. 1, 2021		
(30)	Foreign Application Priority Data		
	Jan. 23, 2018 (EP) 18152903		
(51)	Int. Cl. <i>G06F 9/50</i> (2006.01)		
(52)	U.S. Cl. CPC <i>G06F 9/5061</i> (2013.01); <i>G06F 2209/509</i> (2013.01)		
(58)	Field of Classification Search		
	CPC <i>G06F 9/5061</i> ; <i>G06F 2209/509</i>		
	See application file for complete search history.		
			10 Claims, 1 Drawing Sheet



If a task is computed with a first division of sub-tasks between a computation node and a booster, it is possible that such a division is not an optimal division for the computation of the task. Certain sub-tasks which were assigned to a booster on the first iteration might in fact not be suitable for processing by the booster such that a processing of the sub-tasks by a computation node rather than a booster might optimize the computation of the task as a whole. Accordingly, for a second and possible further iterations, if needed, of the task with an altered second and/or further sub-task distribution might improve the efficiency of the computation of the task.

EX1001 ('442 Patent) at 3:59-4:3

'442 Patent: Reassignment of Sub-tasks in Subsequent Iteration

(12) **United States Patent**
Lippert et al.

(10) **Patent No.:** US 11,537,442 B2
(45) **Date of Patent:** Dec. 27, 2022

(54) **APPLICATION RUNTIME DETERMINED DYNAMICAL ALLOCATION OF HETEROGENEOUS COMPUTE RESOURCES**

(71) Applicant: **ParTec Cluster Competence Center GmbH**, Munich (DE)

(72) Inventors: **Thomas Lippert**, Aschaffenburg (DE); **Bernhard Frohwitter**, Munich (DE)

(73) Assignee: **ParTec Cluster Competence Center GmbH**, Munich (DE)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 35 days.

(21) Appl. No.: **16/963,749**

(22) PCT Filed: **Jan. 23, 2019**

(86) PCT No.: **PCT/EP2019/051615**

§ 371 (c)(1),
(2) Date: **Jul. 21, 2020**

(87) PCT Pub. No.: **WO2019/145354**

PCT Pub. Date: **Aug. 1, 2019**

(65) **Prior Publication Data**

US 2021/0096923 A1 Apr. 1, 2021

(30) **Foreign Application Priority Data**

Jan. 23, 2018 (EP) 18152903

(51) **Int. Cl.**
G06F 9/50 (2006.01)

(52) **U.S. Cl.**
CPC **G06F 9/5061** (2013.01); **G06F 2209/509** (2013.01)

(58) **Field of Classification Search**
CPC **G06F 9/5061**; **G06F 2209/509**
See application file for complete search history.

(56)

References Cited

U.S. PATENT DOCUMENTS

2012/0317168 A1* 12/2012 Driesen G06Q 30/06

2013/0282787 A1* 10/2013 Lippert H04L 29/08081

709/202

709/201

FOREIGN PATENT DOCUMENTS

EP 2442228 A1 4/2012

WO WO-2018065530 A1 4/2018

OTHER PUBLICATIONS

International Search Report for International Application No. PCT/EP2019/0518615, dated Apr. 29, 2019.

(Continued)

Primary Examiner — Sisley N Kim

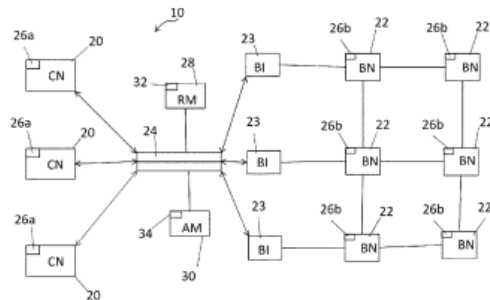
(74) *Attorney, Agent, or Firm* — Marshall, Gerstein & Borun LLP

(57)

ABSTRACT

The present invention provides a method of operating a heterogeneous computing system comprising a plurality of computation nodes and a plurality of booster nodes, at least one of the plurality of computation nodes and plurality of booster nodes being arranged to compute a computation task, the computation task comprising a plurality of sub-tasks, wherein in a first computing iteration, the plurality of sub-tasks are assigned to and processed by ones of the plurality of computation nodes and booster nodes in a first distribution; and information relating to the processing of the plurality of sub-tasks by the plurality of computation nodes and booster nodes is used to generate a further distribution of the sub-tasks between the computation nodes and booster node for processing thereby in a further computing iteration.

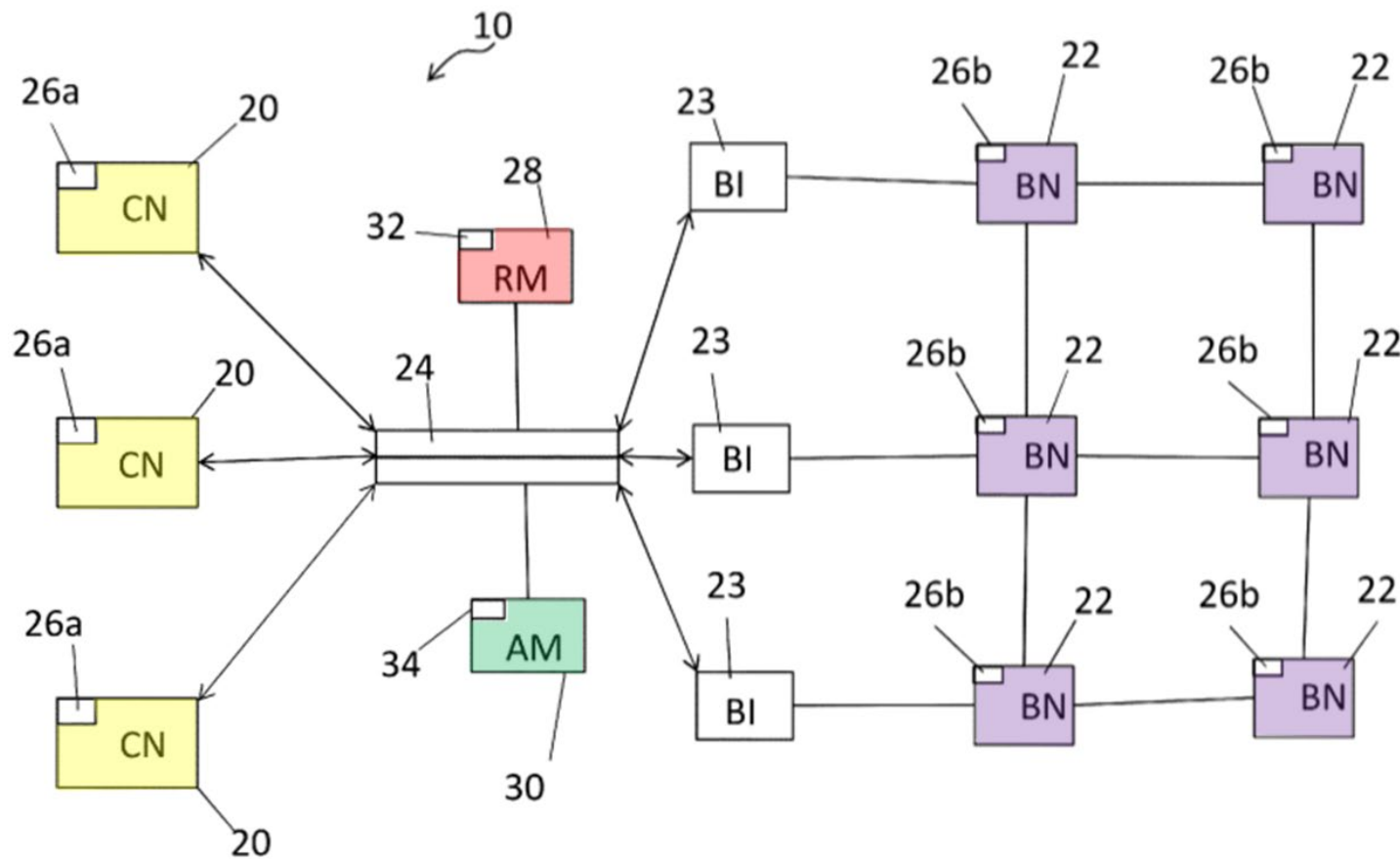
10 Claims, 1 Drawing Sheet



ment of boosters to computation nodes. During a first iteration of the task, the results of the execution of the sub-tasks are collected together with the information from the daemons concerning the processing of the sub-tasks and the status of the nodes. The application manager then performs a reassignment of the sub-tasks for a subsequent iteration of the task and passes this updated assignment information to the resource manager which may also adjust the assignment of boosters to nodes accordingly.

EX1001 ('442 Patent) at 4:23-31

'442 Patent: Further Adjustments to Allocation of Sub-tasks



Accordingly, the system 10 includes a mechanism whereby each of the computation nodes and the cluster nodes are arranged such that daemons 26a, 26b and 32 feed back information to the daemon 34 relating to the processing of sub-tasks and a current state of the respective processing entity. The daemon 34 uses the information provided by the daemons 26a, 26b and 32 to determine whether the distribution of sub-tasks to computation nodes and boosters could be adjusted to optimize or adapt the computation of the task for a subsequent iteration. The resource manager can also reassign computation nodes and boosters to each other, in addition to the adjusted distribution of tasks.

EX1001 ('442 Patent) at 4:4-15

For each iteration, the daemons operating in the computation nodes and the boosters report status information to the application manager and the resource manager enabling the calculation of subsequent iterations to be optimized by a further adjustments to the allocation of sub-tasks to computation nodes and boosters.

EX1001 ('442 Patent) at 4:32-38

Further Distribution of Sub-Tasks

generating, using information relating to the processing of the plurality of sub-tasks by at least the portion of the plurality of computation nodes and at least the portion of the plurality of booster nodes, a further distribution of the plurality of sub-tasks between the plurality of computation nodes and the plurality of booster nodes for processing thereby in a further computing iteration.

EX1001 ('442 Patent)
at 5:37-44

at 5-8, 21, 23-29. That entails at least two components that the Petition grounds fail to meet: (1) the plurality of sub-tasks must be processed by at least one computation node and at least one booster node during the further computing iteration, and (2) a “further distribution” (or redistribution) of sub-tasks between the computation nodes and booster nodes must be generated in a further computing iteration (*i.e.*, at least one subtask is redistributed to a different node for processing).

Surreply at 2

(1) Further Distribution to both CNs and BNs

There is no dispute that claim 1 requires that, in a given computing iteration, the plurality of sub-tasks are distributed among both (i) one or more computing nodes and (ii) one or more booster nodes. *E.g.*, POR, 23-27.

Petitioner Reply at 1

Q. Is it your view that "a portion" means one or more in this situation?

MR. CLAWSON: Objection. Form.

THE WITNESS: I think taken together, "at least a portion," the "at least" means you got to have some. So for me, it means one or more.

EX2020 (Long Depo.) at 47:18-23; see also *id.* 46:18-47:23

Budenske's ATR example distributes subtasks to same processor type

A Method for the On-Line Use of Off-Line Derived Remappings of Iterative Automatic Target Recognition Tasks onto a Particular Class of Heterogeneous Parallel Platforms

JOHN R. BUDENSKIE AND RANGA S. RAMANUJAN
Architecture Technology Corporation, P.O. Box 24344, Minneapolis, MN 55424
ranga.j@atcorp.com

HOWARD JAY SIEGEL
Parallel Processing Laboratory, School of Electrical and Computer Engineering, Purdue University,
Lafayette, IN 47907-1285, USA, hj@purdue.edu

Abstract. This study focuses on a particular application domain and an associated specific class of dedicated heterogeneous parallel environment considered, a methodology is presented for the on-line whether to perform a remapping of the application onto the platform data by the application during execution. If the decision is to remap a mapping, which is appropriate for the given state of the application previously derived with an off-line heuristic.

Keywords: Algorithm mapping, automatic target recognition, general real-time processing

1. Introduction

Heterogeneous computing (*HC*) covers a great variety of applications [22], [23], [26]). This study focuses on a particular automatic target recognition (*ATR*) tasks) and an associated heterogeneous parallel hardware platforms. The computational environment considered, it presents a input-data dependent remapping of the application heterogeneous parallel hardware platform using a predetermined *mapping* (i.e., a matching of application subtasks to the execution order of these subtasks). That is, the operating system will be able to make a heuristically-based decision during the execution of the application whether to perform a remapping based on information generated by the application from its input data. If the decision is to remap, the operating system will be able to select a previously heuristically derived and stored mapping that is appropriate for the given state of the application (e.g., the number of objects it is currently tracking).

For simplicity, it is assumed that if an implementation of a given subtask uses multiple processors, all processors will be of the same type. Given this and the symmetry property of the inter-processor communications among processors of the same type, the expected execution time of a particular multiprocessor implementation of a subtask is independent of which fixed-size subset of the processors of a given type are assigned to execute the subtask.

The IOS design should be capable of working with any hardware platform of the type described above. A given hardware platform may be used for many different ATR applications. It is assumed that when a given ATR application is executing on a platform, that platform is dedicated to that application.

EX1005 (Budenske) at 392; see also EX1007 (Kwok) at 81-82; EX1008 (Theys) at 156

(2) No Redistribution of Subtasks

65. Furthermore, I have reviewed the example “dynamic parameters”

Budenske describes on page 393:

The task is the tracking of vehicles on the ground, and the Application Developer has chosen four dynamic parameters: (1) number of objects of interest being tracked in the last image frame, (2) average size of the objects currently being tracked (in terms of percent of the total number of pixels in the image), (3) whether the lighting conditions in the last image frame required requantization for better contrast for display, and (4) whether there was cloud cover detected in the last image frame that required special processing. The representative choices specified for the dynamic parameters are: (1) 0, 1, 2, 4, 8, 12, (2) 5%, 10%, 15%, 20%, 40%, (3) yes or no, and (4) yes or no.

EX2021 (Kaeli Decl.) ¶¶ 65-66

66. This example further supports the fact the Budenske is simply disclosing assigning subtasks to one type of processor. Budenske is processing image frames in this example, and I cannot think of any combination of parameters that would result in a subtask being moved from a DSP to RISC processor. For example, DSP processors are specifically designed to process array-based data, and typically include multiply/accumulate units that are well suited for image processing tasks. RISC processors typically are not optimized to run these tasks, and almost certainly would not have been optimized to run such tasks at the time Budenske was published.

Petitioner's Expert Agrees No Redistribution

Q. (By Mr. Limbacher) So given the example here in Budenske of these dynamic parameters, one, two, three, and four, can you think of a change in value that would likely result in moving a sub-task from one processor type to the other processor type?

MR. CLAWSON: Objection. Form.

THE WITNESS: Not for this task.

EX2020 (Long Depo.) at 136:23-137:4

Budenske assigns subtasks to best-suited processors

A Method for the On-Line Use of
Remappings of Iterative Automatic
Tasks onto a Particular Class of H
Parallel Platforms

JOHN R. BUDENSKÉ AND RANGA S. RAMANUJAN
Architecture Technology Corporation, P.O. Box 24344, Minneapolis,
ranga[at]atcorp.com

HOWARD JAY SIEGEL
Parallel Processing Laboratory, School of Electrical and Computer E
Lafayette, IN 47907-1285, USA, hj[at]purdue.edu

Abstract. This study focuses on a particular application domain (it
and an associated specific class of dedicated heterogeneous parallel hardware platforms. For the computational
environment considered, a methodology is presented for the on-line operating system to decide heuristically
whether to perform a remapping of the application onto the platform based on information generated from input
data by the application during execution. If the decision is to remap, the operating system will be able to select
a mapping, which is appropriate for the given state of the application, from a stored set of mappings that were
previously derived with an off-line heuristic.

Keywords: Algorithm mapping, automatic target recognition, genetic a
real-time processing

1. Introduction

Heterogeneous computing (*HC*) covers a great variety of
[22], [23], [26]). This study focuses on a particular appli
matic target recognition (*ATR*) tasks) and an associated s
erogeneous parallel hardware platforms. The contribution
computational environment considered, it presents a meth
input-data dependent remapping of the application subta
heterogeneous parallel hardware platform using a previo
determined *mapping* (i.e., a matching of application subtas
uling for the execution order of these subtasks). That is, the
to make a heuristically-based decision during the execution
perform a remapping based on information generated by
data. If the decision is to remap, the operating system will
heuristically derived and stored mapping that is appropri
application (e.g., the number of objects it is currently track

ibility of such a system. Such a high-level operating system must be able to assign each
ATR application subtask to the processors where it is best suited for execution. Often,
subtasks can execute concurrently, sharing resources. Because the execution time of ap-
plication subtasks in an ATR system is highly input-data dependent (e.g., number of
currently located objects), this matching and scheduling of application subtasks to pro-
cessors must be performed dynamically at run time.

EX1005 (Budenske) at 390

[22], [23], [26]). This study focuses on a particular application domain (iterative auto-
matic target recognition (*ATR*) tasks) and an associated specific class of dedicated het-
erogeneous parallel hardware platforms. The contribution of this paper is that, for the
computational environment considered, it presents a methodology for real-time on-line
input-data dependent remapping of the application subtasks to the processors in the
heterogeneous parallel hardware platform using a previously stored off-line statically
determined *mapping* (i.e., a matching of application subtasks to processors and a sched-
uling for the execution order of these subtasks). That is, the operating system will be able

EX1005 (Budenske) at 387

Petitioner's Reply Regarding (2) Changing Division

subtask (in later iterations). Thus, the sentence does not proscribe *other* sub-tasks being mapped to different processor types or that a given sub-task is re-mapped to a different type in subsequent iterations. EX1030, ¶¶17-19. *See also* EX1007, 156; EX1008, 81; EX1030, ¶¶20-21.

Reply at 5

can be used"). EX1030, ¶¶24-25 Third, neither of PO's cited Budenske disclosures preclude sub-tasks from being assigned to different processor types in either a single computing iteration or in subsequent iterations. Budenske does not

Reply at 10

Outline of Petition Deficiencies

Petition Deficiencies

- **Claims 1 and 9 (All Grounds)**
 - **Further Distribution of Sub-Tasks between CN's and BN's**
 - **Budenske not Enabling**
 - **Generate Further Distribution Using Information Related to Processing**
 - **Budenske's Dynamic Parameters Not Process-Related Information**
 - **Petition Fails to Address Petitioner's District Court Constructions**
- **Claim 3 (All Grounds) — Dynamically Alters Further Distribution**
- **Claims 5 and 9 (All Grounds) — CN & BN Daemons provide claimed information**
- **All Claims (All Grounds) — No Motivation to Combine**

Prior Art Enablement Case Law



“[I]f an obviousness case is based on a non-self-enabled reference, and no other prior art reference or evidence would have enabled a skilled artisan to make the claimed invention, then the invention cannot be said to have been obvious.”

Raytheon Techs. Corp. v. GE, 993 F.3d 1374, 1377 (Fed. Cir. 2021)

Budenske does not enable assigning subtasks to a mix of CN's and BN's in a further distribution

A Method for the On-Line Use of Off-Line Derived Remappings of Iterative Automatic Target Recognition Tasks onto a Particular Class of Heterogeneous Parallel Platforms

JOHN R. BUDENSKIE AND RANGA S. RAMANUJAN

Architecture Technology Corporation, P.O. Box 24344, Minneapolis, MN 55424, USA, {budenske, ranga}@atcorp.com

HOWARD JAY SIEGEL

Parallel Processing Laboratory, School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907-1285, USA, hj@purdue.edu

Abstract. This study focuses on a particular application domain (iterative automatic target recognition tasks) and an associated specific class of dedicated heterogeneous parallel hardware platforms. For the computational environment considered, a methodology is presented for the on-line operating system to decide heuristically whether to perform a remapping of the application onto the platform based on information generated from input data by the application during execution. If the decision is to remap, the operating system will be able to select a mapping, which is appropriate for the given state of the application, from a stored set of mappings that were previously derived with an off-line heuristic.

Keywords: Algorithm mapping, automatic target recognition, genetic algorithms, heterogeneous computing, real-time processing

1. Introduction

Heterogeneous computing (HC) covers a great variety of situations (e.g., see [9], [17], [22], [23], [26]). This study focuses on a particular application domain (iterative automatic target recognition (ATR) tasks) and an associated specific class of dedicated heterogeneous parallel hardware platforms. The contribution of this paper is that, for the computational environment considered, it presents a methodology for real-time on-line input-data dependent remapping of the application subtasks to the processors in the heterogeneous parallel hardware platform using a previously stored off-line statically determined *mapping* (i.e., a matching of application subtasks to processors and a scheduling for the execution order of these subtasks). That is, the operating system will be able to make a heuristically-based decision during the execution of the application whether to perform a remapping based on information generated by the application from its input data. If the decision is to remap, the operating system will be able to select a previously heuristically derived and stored mapping that is appropriate for the given state of the application (e.g., the number of objects it is currently tracking).

needed for the HC Kernel to operate is collected. The IOS has not been implemented; such an implementation is a major undertaking and outside the scope of this paper, which is the design concepts for the HC Kernel.

EX1005 (Budenske) at 389

Thus, the goal of this paper is to introduce the ideas underlying the design and use of the HC Kernel for a particular class of application computational structure and a particular type of target hardware platform. The performance of the HC Kernel in comparison to other methods will depend upon the exact application and exact platform under consideration. The technique presented here is offered to future system designers to evaluate as an alternative for their particular situations.

EX1005 (Budenske) at 390

For simplicity, it is assumed that if an implementation of a given subtask uses multiple processors, all processors will be of the same type. Given this and the symmetry property of the inter-processor communications among processors of the same type, the expected execution time of a particular multiprocessor implementation of a subtask is independent of which fixed-size subset of the processors of a given type are assigned to execute the subtask.

EX1005 (Budenske) at 392

Budenske is not Enabling

The goal of this paper has been to introduce the concepts of the HC Kernel, explain when it can be applied, and to give a framework for developing it. Future research, by our team and other groups, includes building HC Kernel based systems for a variety of different applications and a variety of different target hardware platforms to determine where the HC Kernel methodology can be best exploited.

EX1005 (Budenske) at 404

Q. Do you have any reason to doubt the statement in Budenske that such an implementation would be a major undertaking?

MR. CLAWSON: Objection. Form.

THE WITNESS: I think it's a qualitative statement. I know that they worked on this project for a number of years, so clearly it was not trivial.

EX2020 (Long Depo) at 156:19-25

Budenske is not Enabling

A semi-static approach to mapping dynamic iterative tasks onto heterogeneous computing systems[☆]

Yu-Kwong Kwok^a, Anthony A. Maciejewski^b, Howard Jay Siegel^{b,c}, Ishfaq Ahmad^{d,*},
Arif Ghafoor^e

^aDepartment of Electrical and Electronic Engineering, The University of Hong Kong, Pokfulam Road, Hong Kong, Hong Kong

^bDepartment of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523-1373, USA

^cDepartment of Computer Science, Colorado State University, Fort Collins, CO 80523-1873, USA

^dDepartment of Computer Science and Engineering, University of Texas at Arlington, Arlington, TX 76019-0015, USA

^eSchool of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907-1285, USA

Received 17 February 2005; received in revised form 2 June 2005; accepted 23 June 2005

Available online 10 November 2005

EX1007 (Kwok) at 77

The design of the ATR Kernel and HC Kernel was the focus of Budenske et al. [10]. In addition, two methods were suggested for determining the representative choices for the dynamic parameter values to use for each static mapping. The first was to have the application developer specify what these representative values should be. The second was to have the application developer specify a range for each dynamic parameter, and then the representative values are uniformly distributed over that range. Neither of these methods was ever evaluated in any way, as this was not the focus of Budenske et al. [10]. In particular, the performance of off-line mappings based on points selected in either of these ways was never determined. In contrast, our current work focuses just on the semi-static mapping approach. The cur-

EX1007 (Kwok) at 79

The implementation of the On-Off methodology entails addressing the fundamental research issue of how to select representative mappings off-line for on-line use. To solve

EX1007 (Kwok) at 80

Fundamental Research Defined

76. “Fundamental research” is a term of art in the research community. It is sometimes referred to as “Basic Research,” but “basic” in this context does not mean simple, but rather that it relates to the building blocks on which further research and experimentation will be based. For example, the National Center for Science and Engineering Statistics describes “Basic Research,” excerpted below:

Basic research is systematic study directed toward greater knowledge or understanding of the fundamental aspects of phenomena and of observable facts without specific applications towards processes or products in mind. It includes all scientific study and experimentation directed toward increasing fundamental knowledge and understanding in those fields of the physical, engineering, environmental, and life sciences related to long-term national security needs. It is farsighted high payoff research that provides the basis for technological progress. Basic research may lead to: (a) subsequent applied research and advanced technology developments in Defense-related technologies, and (b) new and improved military functional capabilities in areas such as communications, detection, tracking, surveillance, propulsion, mobility, guidance and control, navigation, energy conversion, materials and structures, and personnel support.

See EX2021 (Kaeli Decl.) ¶¶ 76-78 (quoting National Center for Science and Engineering Statistics definition of fundamental research)

Budenske is not Enabling

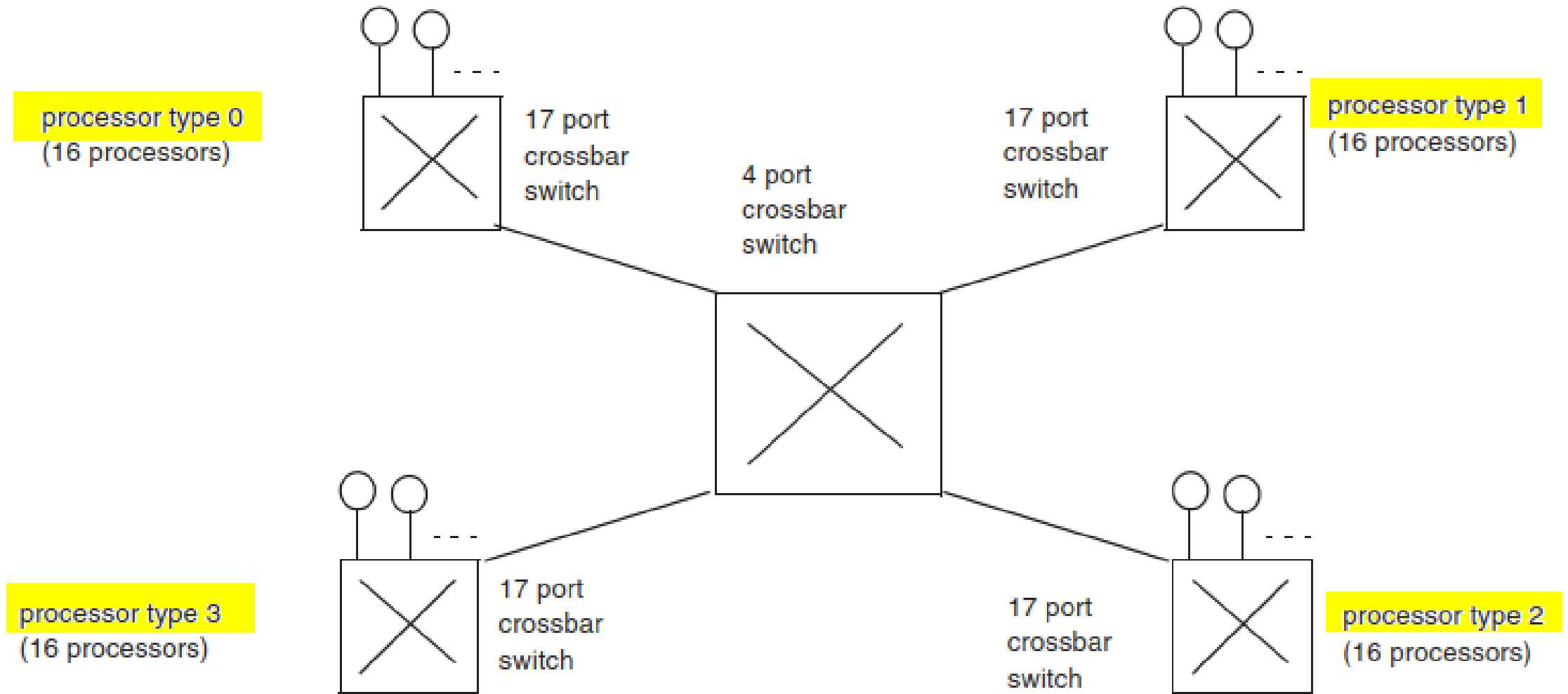
THE WITNESS: I would say most of my research is considered fundamental research. And according to the NSF definition, it's also basic research.

Q. (By Mr. Limbacher) What's the NSF definition?

A. It's -- the short version is curiosity-driven research for unsolved problems. If you already have a solution, then it's not really considered fundamental.

EX2020 (Long Depo) at 123:16-22

Budenske is not Enabling



Budenske is not Enabling

Notation	Definition
s_i	Subtask i
n	Number of subtasks in the application task
<i>Subtask execution time equation variables</i>	
$t_u(s_i)$	Execution time of subtask s_i on virtual machine u
h_{iu}	Heterogeneity factor indicating the relative speed of the i th subtask on machine u
a_i	Coefficient for the parallel portion of subtask i 's execution time
b_i	Coefficient for the parallelization overhead portion of subtask i 's execution time
c_i	Coefficient for the serial portion of subtask i 's execution time
p	Number of processors used in a virtual machine
<i>Inter-subtask communication</i>	
d_{ij}	
e_{ij}	
C_{uv}	
S_{uv}	
R_{uv}	
<i>Dynamic parameters</i>	
α	
β	
γ	Dynamic parameter impacting the serial workload
μ	Dynamic parameter impacting the size of the variable portion of data transfer

different machines. The execution time $t_u(s_i)$ of subtask s_i on virtual machine u is modeled by the expression and s_j

$$t_u(s_i) = h_{iu} \cdot (a_i \alpha / p + b_i \beta \log p + c_i \gamma). \tag{1}$$

Outline of Petition Deficiencies

Petition Deficiencies

- **Claims 1 and 9 (All Grounds)**
 - **Further Distribution of Sub-Tasks between CN's and BN's**
 - Budenske not Enabling
 - **Generate Further Distribution Using Information Related to Processing**
 - **Budenske's Dynamic Parameters Not Process-Related Information**
 - **Petition Fails to Address Petitioner's District Court Constructions**
- **Claim 3 (All Grounds) — Dynamically Alters Further Distribution**
- **Claims 5 and 9 (All Grounds) — CN & BN Daemons provide claimed information**
- **All Claims (All Grounds) — No Motivation to Combine**

Generate Further Distribution Using Processing Information

1. A method of operating a heterogeneous computing system comprising a plurality of computation nodes and a plurality of booster nodes, at least one of the plurality of computation nodes and a plurality of booster nodes being arranged to compute a computation task, the computation task comprising a plurality of sub-tasks, the method comprising:

in a first computing iteration, assigning and processing the plurality of sub-tasks by at least a portion of the plurality of computation nodes and at least a portion of the plurality of booster nodes in a first distribution; and generating, using information relating to the processing of the plurality of sub-tasks by at least the portion of the plurality of computation nodes and at least the portion of the plurality of booster nodes, a further distribution of the plurality of sub-tasks between the plurality of computation nodes and the plurality of booster nodes for processing thereby in a further computing iteration.

9. A heterogeneous computing system comprising:
a plurality of computation nodes and a plurality of booster nodes for computing one or more tasks comprising multiple sub-tasks;
a communication interface connecting the plurality of computation nodes with each other and the plurality of booster nodes;
a resource manager for assigning at least a portion of the plurality of booster nodes and at least a portion of the plurality of computation nodes to each other for the computing of the one or more tasks in a first computing iteration; and
an application manager configured to receive information from daemons operating in at least the portion of the plurality of computation nodes and at least the portion of the plurality of booster nodes to update a distribution of the multiple sub-tasks between the plurality of computation nodes and the plurality of booster nodes in a further computing iteration.

Budenske Uses Off-Line Static Mappings

A Method for the On-Line Use of **Off-Line Derived Remappings** of Iterative Automatic Target Recognition Tasks onto a Particular Class of Heterogeneous Parallel Platforms

[22], [23], [26]). This study focuses on a particular application domain (iterative automatic target recognition (*ATR*) tasks) and an associated specific class of dedicated heterogeneous parallel hardware platforms. The contribution of this paper is that, for the computational environment considered, it presents a methodology for real-time on-line input-data dependent remapping of the application subtasks to the processors in the heterogeneous parallel hardware platform **using a previously stored off-line statically determined mapping** (i.e., a matching of application subtasks to processors and a scheduling for the execution order of these subtasks). That is, the operating system will be able

Budenske Selects From Previously Derived Mappings

A Method for the On-Line Use of Off-Line Derived Remappings of Iterative Automatic Target Recognition Tasks onto a Particular Class of Heterogeneous Parallel Platforms

JOHN R. BUDENSKÉ AND RANGA S. RAMANUJAN

Architecture Technology Corporation, P.O. Box 24344, Minneapolis, MN 55424, USA, {budenske, ranga}@atcorp.com

HOWARD JAY SIEGEL

Parallel Processing Laboratory, School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907-1285, USA, hj@purdue.edu

Abstract. This study focuses on a particular application domain (iterative automatic target recognition tasks) and an associated specific class of dedicated heterogeneous parallel hardware platforms. For the computational environment considered, a methodology is presented for the on-line operating system to decide heuristically whether to perform a remapping of the application onto the platform based on information generated from input data by the application during execution. If the decision is to remap, the operating system will be able to select a mapping, which is appropriate for the given state of the application, from a stored set of mappings that were previously derived with an off-line heuristic.

Keywords: Algorithm mapping, automatic target recognition, genetic algorithms, heterogeneous computing, real-time processing

1. Introduction

Heterogeneous computing (HC) covers a great variety of situations (e.g., see [9], [17], [22], [23], [26]). This study focuses on a particular application domain (iterative automatic target recognition (ATR) tasks) and an associated specific class of dedicated heterogeneous parallel hardware platforms. The contribution of this paper is that, for the computational environment considered, it presents a methodology for real-time on-line input-data dependent remapping of the application subtasks to the processors in the heterogeneous parallel hardware platform using a previously stored off-line statically determined *mapping* (i.e., a matching of application subtasks to processors and a scheduling for the execution order of these subtasks). That is, the operating system will be able to make a heuristically-based decision during the execution of the application whether to perform a remapping based on information generated by the application from its input data. If the decision is to remap, the operating system will be able to select a previously heuristically derived and stored mapping that is appropriate for the given state of the application (e.g., the number of objects it is currently tracking).

Abstract. This study focuses on a particular application domain (iterative automatic target recognition tasks) and an associated specific class of dedicated heterogeneous parallel hardware platforms. For the computational environment considered, a methodology is presented for the on-line operating system to decide heuristically whether to perform a remapping of the application onto the platform based on information generated from input data by the application during execution. If the decision is to remap, the operating system will be able to select a mapping, which is appropriate for the given state of the application, from a stored set of mappings that were previously derived with an off-line heuristic.

The HC Kernel differs from other real-time HC mapping techniques in that it allows on-line real-time use of off-line precomputed mappings. This is significant because it is possible for off-line heuristics to produce better mappings than on-line heuristics. This is a result of the fact that off-line heuristics can have much longer execution times to search

EX1005 (Budenske) at 387, 389

Budenske selects a mapping based on dynamic parameter values

A Method for the On-Line Use of Off-Line Derived Remappings of Iterative Automatic Target Recognition Tasks onto a Particular Class of Heterogeneous Parallel Platforms

JOHN R. BUDENSKIE AND RANGA S. RAMANUJAN
Architecture Technology Corporation, P.O. Box 24344, Minneapolis, MN 55424
ranga[at]atcorp.com

HOWARD JAY SIEGEL
Parallel Processing Laboratory, School of Electrical and Computer Engineering,
Lafayette, IN 47907-1285, USA, hj[at]purdue.edu

Abstract. This study focuses on a particular application domain (iterative automatic target recognition (ATR) tasks) and an associated specific class of dedicated heterogeneous parallel hardware platform. In the environment considered, a methodology is presented for the on-line operating whether to perform a remapping of the application onto the platform based on information generated by the application during execution. If the decision is to remap, the operating system will be able to select a previously determined mapping, which is appropriate for the given state of the application, from a set of mappings previously derived with an off-line heuristic.

Keywords: Algorithm mapping, automatic target recognition, genetic algorithm, real-time processing

1. Introduction

Heterogeneous computing (HC) covers a great variety of situations [22], [23], [26]). This study focuses on a particular application domain (iterative automatic target recognition (ATR) tasks) and an associated specific class of dedicated heterogeneous parallel hardware platform. The contribution of this study to the computational environment considered, it presents a methodology for the input-data dependent remapping of the application subtasks onto a particular heterogeneous parallel hardware platform using a previously determined mapping (i.e., a matching of application subtasks to hardware resources) for the execution order of these subtasks. That is, the operating system will be able to select a previously determined mapping that is appropriate for the given state of the application (e.g., the number of objects it is currently tracking).

Figure 3 shows the on-line components of the IOS. Once in the field, the Application User interacts with the *ATR Kernel User Interface*, including the Application Menus. The Application Menus will be used on-line by the Application User to select a particular initial MDDG to use to invoke an application (with an associated set of Application User specified values for static parameters and initial choices of allowable representative dynamic parameters). That MDDG is passed to the *HC Kernel Monitor* to use as the initial mapping. Then, as the application is executing, the HC Kernel Monitor monitors the run-time values of the dynamic parameters at the end of each iteration through the underlying DDG to make a heuristically-based decision whether to continue with the current mapping, or to select and instantiate a new mapping (for the next iteration) from among the entries of the relevant MDDG Table (which were determined off-line). Thus, the off-line processing provides a set of predetermined mappings that the on-line processing can index in real time.

EX1005 (Budenske) at 395

Generate Further Distribution Using Process-Related Information

1. A method of operating a heterogeneous computing system comprising a plurality of computation nodes and a plurality of booster nodes, at least one of the plurality of computation nodes and a plurality of booster nodes being arranged to compute a computation task, the computation task comprising a plurality of sub-tasks, the method comprising:

in a first computing iteration, assigning and processing the plurality of sub-tasks by at least a portion of the plurality of computation nodes and at least a portion of the plurality of booster nodes in a first distribution; and
generating, using information relating to the processing of the plurality of sub-tasks by at least the portion of the plurality of computation nodes and at least the portion of the plurality of booster nodes, a further distribution of the plurality of sub-tasks between the plurality of computation nodes and the plurality of booster nodes for processing thereby in a further computing iteration.

Outline of Petition Deficiencies

Petition Deficiencies

- **Claims 1 and 9 (All Grounds)**
 - **Further Distribution of Sub-Tasks between CN's and BN's**
 - Budenske not Enabling
 - **Generate Further Distribution Using Information Related to Processing**
 - **Budenske's Dynamic Parameters Not Process-Related Information**
 - Petition Fails to Address Petitioner's District Court Constructions
- **Claim 3 (All Grounds) — Dynamically Alters Further Distribution**
- **Claims 5 and 9 (All Grounds) — CN & BN Daemons provide claimed information**
- **All Claims (All Grounds) — No Motivation to Combine**

Information Related to Processing the Sub-Tasks

1. A method of operating a heterogeneous computing system comprising a plurality of computation nodes and a plurality of booster nodes, at least one of the plurality of computation nodes and a plurality of booster nodes being arranged to compute a computation task, the computation task comprising a plurality of sub-tasks, the method comprising:

in a first computing iteration, assigning and processing the plurality of sub-tasks by at least a portion of the plurality of computation nodes and at least a portion of the plurality of booster nodes in a first distribution; and generating, using information relating to the processing of the plurality of sub-tasks by at least the portion of the plurality of computation nodes and at least the portion of the plurality of booster nodes, a further distribution of the plurality of sub-tasks between the plurality of computation nodes and the plurality of booster nodes for processing thereby in a further computing iteration.

9. A heterogeneous computing system comprising:
a plurality of computation nodes and a plurality of booster nodes for computing one or more tasks comprising multiple sub-tasks;
a communication interface connecting the plurality of computation nodes with each other and the plurality of booster nodes;
a resource manager for assigning at least a portion of the plurality of booster nodes and at least a portion of the plurality of computation nodes to each other for the computing of the one or more tasks in a first computing iteration; and
an application manager configured to receive information from daemons operating in at least the portion of the plurality of computation nodes and at least the portion of the plurality of booster nodes to update a distribution of the multiple sub-tasks between the plurality of computation nodes and the plurality of booster nodes in a further computing iteration.

Information Related to Processing Sub-Tasks

1. A method of operating a heterogeneous computing system comprising a plurality of computation nodes and a plurality of booster nodes, at least one of the plurality of computation nodes and a plurality of booster nodes being arranged to compute a computation task, the computation task comprising a plurality of sub-tasks, the method comprising:

in a first computing iteration, assigning and processing the plurality of sub-tasks by at least a portion of the plurality of computation nodes and at least a portion of the plurality of booster nodes in a first distribution; and generating, using information relating to the processing of the plurality of sub-tasks by at least the portion of the plurality of computation nodes and at least the portion of the plurality of booster nodes, a further distribution of the plurality of sub-tasks between the plurality of computation nodes and the plurality of booster nodes for processing thereby in a further computing iteration.

the execution, a task is analyzed by the application manager and sub-tasks of the task are divided into computation node suitable sub-tasks and booster suitable sub-tasks and this information is passed to the resource manager for assignment of boosters to computation nodes. During a first iteration of the task,^[1] the results of the execution of the sub-tasks are collected together with^[2] the information from the daemons concerning the processing of the sub-tasks and the status of the nodes. The application manager then performs a reassignment of the sub-tasks for a subsequent iteration of the task and passes this updated assignment information to the resource manager which may also adjust the assignment of boosters to nodes accordingly.

EX1001 ('442 Patent) at 4:19-31 (bracketed numbers added); *see also id.* at 4:32-37

'442 Patent: Information Learned About Processing Sub-Task

(12) United States Patent Lippert et al.

(10) Patent No.: US 11,537,442 B2
(45) Date of Patent: Dec. 27, 2022

(54) APPLICATION RUNTIME DETERMINED
DYNAMICAL ALLOCATION OF
HETEROGENEOUS COMPUTE RESOURCES

(71) Applicant: ParTec Cluster Competence Center
GmbH, Munich (DE)

(72) Inventors: Thomas Lippert, Aschaffenburg (DE);
Bernhard Frohwitter, Munich (DE)

(73) Assignee: ParTec Cluster Competence Center
GmbH, Munich (DE)

(*) Notice: Subject to any disclaimer, the term of this
patent is extended or adjusted under 35
U.S.C. 154(b) by 35 days.

(21) Appl. No.: 16/963,749

(22) PCT Filed: Jan. 23, 2019

(86) PCT No.: PCT/EP2019/051615

§ 371 (c)(1),
(2) Date: Jul. 21, 2020

(87) PCT Pub. No.: WO2019/145354

PCT Pub. Date: Aug. 1, 2019

(65) Prior Publication Data

US 2021/0096923 A1 Apr. 1, 2021

(30) Foreign Application Priority Data

Jan. 23, 2018 (EP) 18152903

(51) Int. CL G06F 9/50 (2006.01)

(52) U.S. CL CPC G06F 9/5061 (2013.01); G06F 2209/509
(2013.01)

(58) Field of Classification Search
CPC G06F 9/5061; G06F 2209/509
See application file for complete search history.

(56) References Cited

U.S. PATENT DOCUMENTS

2012/0317168 A1* 12/2012 Driesen G06Q 30/06

709/202

2013/0282787 A1* 10/2013 Lippert H04L 29/08081

709/201

FOREIGN PATENT DOCUMENTS

EP 2442228 A1 4/2012

WO WO-2018065530 A1 4/2018

OTHER PUBLICATIONS

International Search Report for International Application No. PCT/
EP2019/0518615, dated Apr. 29, 2019.

(Continued)

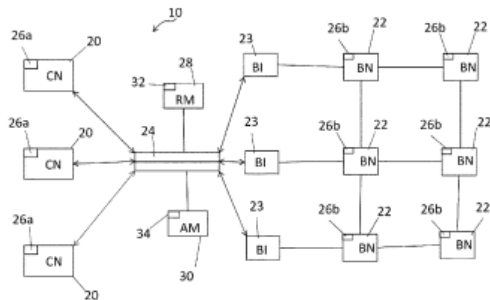
Primary Examiner — Sisley N Kim

(74) Attorney, Agent, or Firm — Marshall, Gerstein &
Borun LLP

(57) ABSTRACT

The present invention provides a method of operating a heterogeneous computing system comprising a plurality of computation nodes and a plurality of booster nodes, at least one of the plurality of computation nodes and plurality of booster nodes being arranged to compute a computation task, the computation task comprising a plurality of sub-tasks, wherein in a first computing iteration, the plurality of sub-tasks are assigned to and processed by ones of the plurality of computation nodes and booster nodes in a first distribution; and information relating to the processing of the plurality of sub-tasks by the plurality of computation nodes and booster nodes is used to generate a further distribution of the sub-tasks between the computation nodes and booster node for processing thereby in a further computing iteration.

10 Claims, 1 Drawing Sheet



As well as adjusting the distribution of sub-tasks between computation nodes and boosters based on a scalability of the sub-task, the distribution may also be influenced by information learned about the processing of the sub-task and any need to call further sub-tasks during the processing. If a first sub-task being handled by a booster requires input from a second sub-task not being handled by the booster, this may lead to an interruption in the processing of the first sub-task. Accordingly, the daemon at the booster handling the first sub-task can report this situation to the application manager such that in a further iteration both the first and second sub-tasks are handled by the booster. Accordingly, the application manager is arranged using the information provided by the daemons operating in the computation nodes and the boosters to adjust groupings of sub-tasks for assignment to the computation nodes and boosters.

EX1001 ('442 Patent) at 4:48-63

Petition Relies on Budenske's Processing Results

EX1005, 395. Budenske thus discloses using values of **dynamic parameters** *(information relating to the processing of the sub-tasks by at least the portion of the plurality of computation nodes and at least the portion of the plurality of booster nodes)* to generate a “new assignment” *(further distribution)* of sub-tasks to resources “to use for the next execution iteration through the subtasks” *(in a further computing iteration)*. *Id.*, 391; EX1003, ¶77.

Pet. at 30

parameters are “computed by the application as it executes”). Therefore, the result of processing the sub-tasks—the values of the dynamic parameters—is information relating to the processing of the sub-tasks. The “most recent values of such dynamic

Pet. at 31

The Claimed Information-Petition Grounds

And my question is: Are you mapping the dynamic parameters of Budenske to the information relating to the processing of sub-tasks recited in the claim?

A. Yes.

EX2020 (Long Depo) at 129:10-14

Budenske Dynamic Parameters are Computed by the Application

A Method for the On-Line Use of Off-Line Derived Remappings of Iterative Automatic Target Recognition Tasks onto a Particular Class of Heterogeneous Parallel Platforms

JOHN R. BUDENSKÉ AND RANGA S. RAMANUJAN
Architecture Technology Corporation, P.O. Box 24344, Minneapolis, MN 55424
ranga.j@atcorp.com

HOWARD JAY SIEGEL
Parallel Processing Laboratory, School of Electrical and Computer Engineering
Lafayette, IN 47907-1285, USA, hj@purdue.edu

Abstract. This study focuses on a particular application domain (iterative automatic target recognition (ATR) tasks) and an associated specific class of dedicated heterogeneous parallel hardware environment considered, a methodology is presented for the on-line operation whether to perform a remapping of the application onto the platform based on data by the application during execution. If the decision is to remap, the operation is a mapping, which is appropriate for the given state of the application, from a previously derived with an off-line heuristic.

Keywords: Algorithm mapping, automatic target recognition, genetic algorithm, real-time processing

1. Introduction

Heterogeneous computing (HC) covers a great variety of situations [22], [23], [26]). This study focuses on a particular application domain (iterative automatic target recognition (ATR) tasks) and an associated specific class of dedicated heterogeneous parallel hardware environment considered, a methodology is presented for the on-line operation whether to perform a remapping of the application onto the platform based on data by the application during execution. If the decision is to remap, the operation is a mapping, which is appropriate for the given state of the application, from a previously derived with an off-line heuristic.

Each of these sets of Application and Input Data Characteristics can be divided into static and dynamic parameters. *Static parameters* are those Input Data Characteristics, such as image size, and Application Characteristics, such as type of object of interest (e.g., tank), that will not change during the execution of the application task. *Dynamic parameters*, in contrast, are those Input Data Characteristics, such as amount of clutter, and Application Characteristics, such as number of located objects to be identified, that will change during run time and can be computed by the application as it executes. When an Application Developer instantiates a DDG to implement an application in a given environment, the values for the static characteristics are known and provided by the Application Developer. For the dynamic characteristics, the Application Developer is expected to define a small set of dynamic parameters that will have the most impact on the execution time of the subtasks in the application. It is the job of the *Scenario Generator* component of the IOS to use information provided by the Application Developer to derive representative values for these dynamic parameters.

EX1005 (Budenske) at 393

Budenske Dynamic Parameters

A Method for the On-Line Use of Off-Line Derived Remappings of Iterative Automatic Target Recognition Tasks onto a Particular Class of Heterogeneous Parallel Platforms

JOHN R. BUDENSKÉ AND RANGA S. RAMANUJAN

Architecture Technology Corporation, P.O. Box 24344, Minneapolis, MN 55424, USA, {budenske, ranga}@atcorp.com

HOWARD JAY SIEGEL

Parallel Processing Laboratory, School of Electrical and Computer Engineering, Purdue University, West Lafayette, IN 47907-1285, USA, hj@purdue.edu

Abstract. This study focuses on a particular application domain (iterative automatic target recognition tasks) and an associated specific class of dedicated heterogeneous parallel hardware platform environment considered, a methodology is presented for the on-line operating system whether to perform a remapping of the application onto the platform based on information data by the application during execution. If the decision is to remap, the operating system a mapping, which is appropriate for the given state of the application, from a stored mapping previously derived with an off-line heuristic.

Keywords: Algorithm mapping, automatic target recognition, genetic algorithms, heuristic, real-time processing

1. Introduction

Heterogeneous computing (HC) covers a great variety of situations [22], [23], [26]). This study focuses on a particular application domain: automatic target recognition (ATR) tasks and an associated specific class of heterogeneous parallel hardware platforms. The contribution of this paper to the computational environment considered, it presents a methodology for input-data dependent remapping of the application subtasks to the heterogeneous parallel hardware platform using a previously stored determined mapping (i.e., a matching of application subtasks to processing units for the execution order of these subtasks). That is, the operating system to make a heuristically-based decision during the execution of the application to perform a remapping based on information generated by the application data. If the decision is to remap, the operating system will be able to select a previously heuristically derived and stored mapping that is appropriate for the given state of the application (e.g., the number of objects it is currently tracking).

Consider the following hypothetical example. The task is the tracking of vehicles on the ground, and the Application Developer has chosen four dynamic parameters: (1) number of objects of interest being tracked in the last image frame, (2) average size of the objects currently being tracked (in terms of percent of the total number of pixels in the image), (3) whether the lighting conditions in the last image frame required requantization for better contrast for display, and (4) whether there was cloud cover detected in the last image frame that required special processing. The representative choices specified for the dynamic parameters are: (1) 0, 1, 2, 4, 8, 12, (2) 5%, 10%, 15%, 20%, 40%, (3) yes or no, and (4) yes or no. Thus, $C_0 = 6$, $C_1 = 5$, $C_2 = 2$, $C_3 = 2$, and $S = 120$.

EX1005 (Budenske) at 393

The Claimed Process-Related Information

1. A method of operating a heterogeneous computing system comprising a plurality of computation nodes and a plurality of booster nodes, at least one of the plurality of computation nodes and a plurality of booster nodes being arranged to compute a computation task, the computation task comprising a plurality of sub-tasks, the method comprising:

in a first computing iteration, assigning and processing the plurality of sub-tasks by at least a portion of the plurality of computation nodes and at least a portion of the plurality of booster nodes in a first distribution; and generating, using information relating to the processing of the plurality of sub-tasks by at least the portion of the plurality of computation nodes and at least the portion of the plurality of booster nodes, a further distribution of the plurality of sub-tasks between the plurality of computation nodes and the plurality of booster nodes for processing thereby in a further computing iteration.

5. The method according to claim 1, wherein daemons operate in the plurality of computation nodes and the plurality of booster nodes to generate the information.

Outline of Petition Deficiencies

Petition Deficiencies

- **Claims 1 and 9 (All Grounds)**
 - **Further Distribution of Sub-Tasks between CN's and BN's**
 - Budenske not Enabling
 - **Generate Further Distribution Using Information Related to Processing**
 - **Budenske's Dynamic Parameters Not Process-Related Information**
 - **Petition Fails to Address Petitioner's District Court Constructions**
- **Claim 3 (All Grounds) — Dynamically Alters Further Distribution**
- **Claims 5 and 9 (All Grounds) — CN & BN Daemons provide claimed information**
- **All Claims (All Grounds) — No Motivation to Combine**

Petitioner's Claim Construction Positions



The Board's claim construction rules are designed to ensure that the Board correctly construes claim terms and to minimize inconsistency in claim construction between forums. *See* Changes to the Claim Construction Standard for Interpreting Claims in Trial Proceedings Before the Patent Trial and Appeal Board, 83 Fed. Reg. 51,340, 51,349 (Oct. 11, 2018). To that end, the rules discourage petitioners from seeking broader constructions at the Board to support a patentability challenge while seeking narrower constructions in litigation to avoid infringement liability. *See id.* at 51,350.

Revvo Techs. Inc. v. Cerebrum Sensor Techs, Inc., IPR2025-00632, Paper 20 at 3-5 (Dir. Squires Nov. 3, 2025) (precedential).

Petitioner's District Court CC Argument

1. “First computing iteration” / “further computing iteration” (’442 claims 1, 3, 7, 9)

ParTec’s Proposed Construction	Microsoft’s Proposed Construction
Plain and ordinary meaning	“an initial execution of the plurality of sub-tasks” / “a repeated execution of the plurality of sub-tasks”

EX1031 (Pet. CC Brief) at 24

definition, involves a repeated execution. The only genuine dispute is whether each iteration involves the *same* plurality of sub-tasks.⁶ The claims answer that question unambiguously—they do. Microsoft’s construction reflects that straightforward reading of the claim language and thus should be adopted, whereas ParTec’s argument that “it may not be the same sub-tasks repeated in each iteration” (Br. at 25) should be rejected as flatly inconsistent with the claims and specification.

EX1031 (Pet. CC Brief) at 25

Petitioner's District Court CC Argument

at *4-5 (Fed. Cir. Oct. 15, 2025) (quoting *Baldwin*). That rule alone answers the question: it is the *same* “plurality of sub-tasks” that is executed in the “first computing iteration” and then repeated in the “further computing iteration.”

EX1031 (Pet. CC Brief) at 26

Microsoft's construction is also fully consistent with the specification and the goal of the invention. The specification itself consistently discusses changing the *division* of sub-tasks among booster and computation nodes (based on whether they are “more or less suitable” for each), *not* *adding or removing sub-tasks*. See Ex. C ('442 patent), 3:50-4:37. Moreover, the patent describes

EX1031 (Pet. CC Brief) at 26

Petitioner's District Court CC Argument

claims. ParTec also points to the specification's discussion of “any need to call further sub-tasks during processing” (Br. at 25 (quoting Ex. C ('442 patent), 4:48-52) (emphasis omitted)) but the remainder of the passage concerns the *distribution* of sub-tasks between boosters, not the creation of entirely new sub-tasks. Indeed, the very next sentence addresses an example where “a first sub-task being handled by a booster [that] requires input from a second sub-task not being handled by the booster” could create a problem that could be resolved if “in a further iteration both the first and second sub-tasks are handled by the booster.” See Ex. C ('442 patent), 4:52-59. The passage thus addresses moving sub-tasks between boosters to improve efficiency, not adding new sub-tasks.

EX1031 (Pet. CC Brief) at 27

Court's CC Order: Same Sub-tasks in Claimed Computing Iterations

The parties dispute whether “it is the same ‘plurality of sub-tasks’ that is executed in the ‘first computing iteration’ and then repeated in the ‘further computing iteration.’” Dkt. No. 123 at

EX1035 (CC Order)
at 22

than “iteration.” Dkt. No. 124 at 7. Microsoft, however, says the sub-tasks must be the same “under black letter principles of claim construction” and grammar, and ParTec’s assertion otherwise would rewrite the claim language. Dkt. No. 123 at 26.

The Court agrees with Microsoft. “Subsequent use of the definite articles ‘the’ or ‘said’ in

EX1035 (CC Order)
at 23

The Court will give this term a “plain and ordinary meaning” construction, but clarifies “the plurality of sub-tasks” refers to “a plurality of sub-tasks” in the preamble, which are the same sub-tasks throughout performance of the method.

EX1035 (CC Order)
at 24

Petitioner's Reply

IPR2025-00632, Paper 20 (Director Nov. 3, 2025). Here, Petitioner has offered clarifying constructions in district court for the terms “first computing iteration” and “further computing iteration.” *See* EX1031, 24-25 (proposing “an initial execution of the plurality of sub-tasks” and “a repeated execution of the plurality of sub-tasks,” respectively). Those constructions are consistent, however, with the plain meaning and how Petitioner has mapped the claims in this IPR. They were proffered in district court simply to confirm for the jury that “the plurality of subtasks” refer to the same plurality of subtasks.

Reply at 27

Petitioner's Expert in this IPR

Q. (By Mr. Limbacher) Sure.

In the context of Claim 1, is it a requirement that every sub-task is completed in every computing iteration?

MR. CLAWSON: Objection. Form.

THE WITNESS: I don't see that requirement there.

EX2020 (Long Depo) at 28:24-29:5

Petitioner's Claim Construction Positions

1. A method of operating a heterogeneous computing system comprising a plurality of computation nodes and a plurality of booster nodes, at least one of the plurality of computation nodes and a plurality of booster nodes being arranged to compute a computation task, the computation task comprising a plurality of sub-tasks, the method comprising:

in a first computing iteration, assigning and processing the plurality of sub-tasks by at least a portion of the plurality of computation nodes and at least a portion of the plurality of booster nodes in a first distribution; and generating, using information relating to the processing of the plurality of sub-tasks by at least the portion of the plurality of computation nodes and at least the portion of the plurality of booster nodes, a further distribution of the plurality of sub-tasks between the plurality of computation nodes and the plurality of booster nodes for processing thereby in a further computing iteration.

9. A heterogeneous computing system comprising:
a plurality of computation nodes and a plurality of booster nodes for computing one or more tasks comprising multiple sub-tasks;
a communication interface connecting the plurality of computation nodes with each other and the plurality of booster nodes;
a resource manager for assigning at least a portion of the plurality of booster nodes and at least a portion of the plurality of computation nodes to each other for the computing of the one or more tasks in a first computing iteration; and
an application manager configured to receive information from daemons operating in at least the portion of the plurality of computation nodes and at least the portion of the plurality of booster nodes to update a distribution of the multiple sub-tasks between the plurality of computation nodes and the plurality of booster nodes in a further computing iteration.

Outline of Petition Deficiencies

Petition Deficiencies

- **Claims 1 and 9 (All Grounds)**
 - **Further Distribution of Sub-Tasks between CN's and BN's**
 - Budenske not Enabling
 - **Generate Further Distribution Using Information Related to Processing**
 - **Budenske's Dynamic Parameters Not Process-Related Information**
 - **Petition Fails to Address Petitioner's District Court Constructions**
- **Claim 3 (All Grounds) — Dynamically Alters Further Distribution**
- **Claims 5 and 9 (All Grounds) — CN & BN Daemons provide claimed information**
- **All Claims (All Grounds) — No Motivation to Combine**

Claim 3: Dynamically Alters Further Distribution

1. A method of operating a heterogeneous computing system comprising a plurality of computation nodes and a plurality of booster nodes, at least one of the plurality of computation nodes and a plurality of booster nodes being arranged to compute a computation task, the computation task comprising a plurality of sub-tasks, the method comprising:

in a first computing iteration, assigning and processing the plurality of sub-tasks by at least a portion of the plurality of computation nodes and at least a portion of the plurality of booster nodes in a first **distribution**; and generating, using information relating to the processing of the plurality of sub-tasks by at least the portion of the plurality of computation nodes and at least the portion of the plurality of booster nodes, a **further distribution** of the plurality of sub-tasks between the plurality of computation nodes and the plurality of booster nodes for processing thereby in a further computing iteration.

2. The method according to claim 1, wherein an **application manager** receives the information and **determines the further distribution**.

3. The method according to claim 2, wherein a resource manager determines the assignment of the plurality of sub-tasks to the plurality of computation nodes and the plurality of booster nodes for the first computing iteration as a function of the computation task and wherein **the application manager** receives the information and processes the information as input to the resource manager such that the resource manager **dynamically alters further distribution** during the computing of the computation task.

Altering Further Distribution-Petition Grounds

The Petition relies on Lippert for this Claim 3 limitation:

determines the further distribution (*processes the information*). Lippert-Budenske further includes a resource manager that “establish[es] a dynamic assignment at runtime, which means during processing of the computation task” (*dynamically alters further distribution during the computing of the computation task*), as taught by Lippert. EX1004, [0013]; *supra* Section VII.C.3.a). EX1003, ¶86.

Pet. at 39

Lippert [0013]: Assignment of Booster to Computation Node

[0013] In this computer cluster arrangement acceleration functionality is being provided by independent boosters. The described computer cluster arrangement allows a loose coupling of those boosters to computation nodes, which may also be referred to as compute nodes. Hence, a sharing of accelerators, here in form of boosters, by computation nodes is feasible. For an assignment of a booster to a computation node a resource manager, in form of a resource manager module or resource manager node, may be provided. The resource manager may establish a static assignment at start of a processing of a computation task. Alternatively or additionally it may establish a dynamic assignment at runtime, which means during processing of the computation task.

United States
Patent Application Publication
Lippert

(19) Pub. No.: US 2013/0282787 A1
(43) Pub. Date: Oct. 24, 2013

(54) COMPUTER CLUSTER ARRANGEMENT FOR PROCESSING A COMPUTATION TASK AND METHOD FOR OPERATION THEREOF

(71) Applicant: ParTec Cluster Competence Center GmbH, München (DE)

(72) Inventor: Thomas Lippert, Aschaffenburg (DE)

(21) Appl. No.: 13/861,429

(22) Filed: Apr. 12, 2013

Related U.S. Application Data

(63) Continuation of application No. PCT/EP2011/067888, filed on Oct. 13, 2011.

Foreign Application Priority Data

Oct. 13, 2010 (EP) 10187436.0

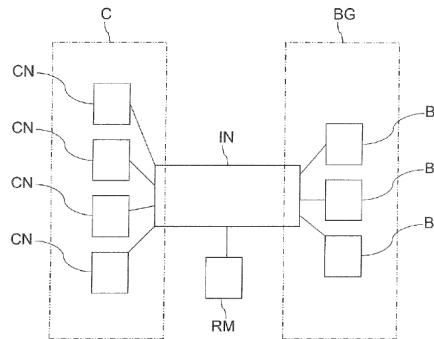
Publication Classification

(51) Int. Cl. H04L 29/08 (2006.01)

(52) U.S. CL. CPC 1104L 29/08081 (2013.01) USPC 709/201

(57) ABSTRACT

The present invention is directed to a computer cluster arrangement and a method for operation of the introduced computer cluster arrangement. The computer cluster arrangement comprises computation nodes CN, which dynamically outsource specific computation tasks to boosters B. Therefore, an assignment technique of boosters B to computation nodes CN is introduced. The assignment takes dynamically place at runtime. The present invention finds application in high performance cluster technology.



MICROSOFT CORP.
EXHIBIT 1004

Page 1 of 18

EX1004 (Lippert) [0013]

Altering Further Distribution-Petition Grounds

The Petition points to Claim 4 limitations to satisfy Claim 3 :

A POSITA would have understood that “the resource manager dynamically alters further distribution” as recited in this claim element *encompasses* “the resource manager dynamically alters assignment of the plurality of computation nodes and the plurality of booster nodes to each other,” because claim 4 depending from claim 3 specifies this, and further because the ’442 patent describes the resource manager adjusting booster assignments according to updated assignment information from the application manager. EX1001, 4:27-31. Thus, in view of the

Pet. at 40

Petition's Position Renders Claim 4 Superfluous

3. The method according to claim 2, wherein a resource manager determines the assignment of the plurality of sub-tasks to the plurality of computation nodes and the plurality of booster nodes for the first computing iteration as a function of the computation task and wherein the application manager receives the information and processes the information as input to the resource manager such that the resource manager dynamically alters further distribution during the computing of the computation task.

4. The method according to claim 3, wherein the resource manager receives the information such that the resource manager dynamically alters assignment of the plurality of computation nodes and the plurality of booster nodes to each other during the computation of the computation task.

EX1001 ('442 Patent) at
5:47-6:11

Petition's Position Renders Claim 4 Superfluous

Q. And then Claim 4, Lines, looks like, about 7 to 11, says, "The resource manager dynamically alters assignment of the plurality of computation nodes and a plurality of booster nodes to each other during the computation of the computation task."

Are you saying the limitation I read from Claim 3 is essentially the same as the limitation I read from Claim 4?

A. Pretty much.

MR. CLAWSON: Objection. Form.

THE WITNESS: The distribution -- the distribution -- those seem like different ways of saying pretty much the same thing.

EX2020 (Long Depo) at
141:1-13

Claim 9

9. A heterogeneous computing system comprising:
a plurality of computation nodes and a plurality of booster nodes for computing one or more tasks comprising multiple sub-tasks;
a communication interface connecting the plurality of computation nodes with each other and the plurality of booster nodes;
a resource manager for assigning at least a portion of the plurality of booster nodes and at least a portion of the plurality of computation nodes to each other for the computing of the one or more tasks in a first computing iteration; and
an application manager configured to receive information from daemons operating in at least the portion of the plurality of computation nodes and at least the portion of the plurality of booster nodes to update a distribution of the multiple sub-tasks between the plurality of computation nodes and the plurality of booster nodes in a further computing iteration.

EX1001 ('442 Patent)
at 6:26-43

Resource Manager Adjusts Sub-Task Assignments During Runtime

(12) **United States Patent**
Lippert et al.

(10) **Patent No.:** US 11,537,442 B2
(45) **Date of Patent:** Dec. 27, 2022

(54) **APPLICATION RUNTIME DETERMINED DYNAMICAL ALLOCATION OF HETEROGENEOUS COMPUTE RESOURCES**

(71) Applicant: **ParTec Cluster Competence Center GmbH**, Munich (DE)

(72) Inventors: **Thomas Lippert**, Aschaffenburg (DE); **Bernhard Frohwitter**, Munich (DE)

(73) Assignee: **ParTec Cluster Competence Center GmbH**, Munich (DE)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 35 days.

(21) Appl. No.: **16/963,749**

(22) PCT Filed: **Jan. 23, 2019**

(86) PCT No.: **PCT/EP2019/051615**
§ 371 (c)(1),
(2) Date: **Jul. 21, 2020**

(87) PCT Pub. No.: **WO2019/145354**
PCT Pub. Date: **Aug. 1, 2019**

(65) **Prior Publication Data**
US 2021/0096923 A1 Apr. 1, 2021

(30) **Foreign Application Priority Data**
Jan. 23, 2018 (EP) 18152903

(51) **Int. CL**
G06F 9/50 (2006.01)

(52) **U.S. CL**
CPC **G06F 9/5061** (2013.01); **G06F 2209/509** (2013.01)

(58) **Field of Classification Search**
CPC **G06F 9/5061**; **G06F 2209/509**
See application file for complete search history.

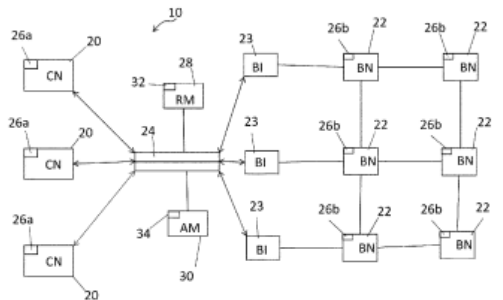
(56) **References Cited**
U.S. PATENT DOCUMENTS
2012/0317168 A1* 12/2012 Driesen G06Q 30/06 709/202
2013/0282787 A1* 10/2013 Lippert H04L 29/08081 709/201

FOREIGN PATENT DOCUMENTS
EP 2442228 A1 4/2012
WO WO-2018065530 A1 4/2018

OTHER PUBLICATIONS
International Search Report for International Application No. PCT/EP2019/0518615, dated Apr. 29, 2019.
(Continued)

Primary Examiner — Sisley N Kim
(74) *Attorney, Agent, or Firm* — Marshall, Gerstein & Borun LLP

(57) **ABSTRACT**
The present invention provides a method of operating a heterogeneous computing system comprising a plurality of computation nodes and a plurality of booster nodes, at least one of the plurality of computation nodes and plurality of booster nodes being arranged to compute a computation task, the computation task comprising a plurality of sub-tasks, wherein in a first computing iteration, the plurality of sub-tasks are assigned to and processed by ones of the plurality of computation nodes and booster nodes in a first distribution; and information relating to the processing of the plurality of sub-tasks by the plurality of computation nodes and booster nodes is used to generate a further distribution of the sub-tasks between the computation nodes and booster node for processing thereby in a further computing iteration.



for a subsequent iteration. The resource manager can also reassign computation nodes and boosters to each other, in addition to the adjusted distribution of tasks.

EX1001 ('442 Patent) at 4:13-15

For each iteration, the daemons operating in the computation nodes and the boosters report status information to the application manager and the resource manager enabling the calculation of subsequent iterations to be optimized by a further adjustments to the allocation of sub-tasks to computation nodes and boosters.

EX1001 ('442 Patent) at 4:32-37

New Obviousness Theory in Reply

(describing the AM and RM of Lippert-Budenske). A POSITA would have been further motivated to implement the RM to dynamically alter the further distribution of the plurality of sub-tasks in order to match how the RM dynamically altered assignment of computation nodes and booster nodes to each other, because doing so would “fully utilize the architectural flexibility” (EX1005, 390; EX1004, [0015]), and “provide fault tolerance in case of accelerator failures” (EX1004, [0003]). For

Reply at 20

Outline of Petition Deficiencies

Petition Deficiencies

- **Claims 1 and 9 (All Grounds)**
 - **Further Distribution of Sub-Tasks between CN's and BN's**
 - Budenske not Enabling
 - **Generate Further Distribution Using Information Related to Processing**
 - **Budenske's Dynamic Parameters Not Process-Related Information**
 - Petition Fails to Address Petitioner's District Court Constructions
- **Claim 3 (All Grounds) — Dynamically Alters Further Distribution**
- **Claims 5 and 9 (All Grounds) — CN & BN Daemons provide claimed information**
- **All Claims (All Grounds) — No Motivation to Combine**

CN Daemons and BN Daemons

5. The method according to claim 1, wherein daemons operate in the plurality of computation nodes and the plurality of booster nodes to generate the information.

9. A heterogeneous computing system comprising:
a plurality of computation nodes and a plurality of booster nodes for computing one or more tasks comprising multiple sub-tasks;
a communication interface connecting the plurality of computation nodes with each other and the plurality of booster nodes;
a resource manager for assigning at least a portion of the plurality of booster nodes and at least a portion of the plurality of computation nodes to each other for the computing of the one or more tasks in a first computing iteration; and
an application manager configured to receive information from daemons operating in at least the portion of the plurality of computation nodes and at least the portion of the plurality of booster nodes to update a distribution of the multiple sub-tasks between the plurality of computation nodes and the plurality of booster nodes in a further computing iteration.

CN Daemons and BN Daemons-Petition Grounds

Ground 1:

during each iteration (*e.g.*, EX1005, 402). In order to send the dynamic parameter values taught by Budenske, as well as information regarding computing capacities as taught by Lippert, from each computation node and each booster node to a central manager to determine a re-mapping of sub-tasks, a POSITA thus would have implemented Lippert-Budenske with daemons at each computation node and each booster node configured to collect and send this information. POSITAs knew that a

Pet. at 41-42

CN Daemons and BN Daemons



violates *Arendi*. In *Arendi*, we cautioned that although “common sense and common knowledge have their proper place in the obviousness inquiry,” (a) invoking “common sense . . . to supply a limitation that was admittedly *missing* from the prior art” should generally only be done when “the [missing] limitation in question [is] unusually simple and the technology particularly straightforward;” and (b) references to common sense “cannot be used as a wholesale substitute for reasoned analysis and evidentiary support.” 832 F.3d at 1361–62. We concluded in *Arendi* that the Board erred in relying on common sense because such reliance was based merely upon “conclusory statements and unspecific expert testimony.” *Id.* at 1366.

Koninklijke Philips N.V. v. Google LLC, 948 F.3d 1330, 1338 (Fed. Cir. 2020)

CN Daemons and BN Daemons

Q. Would it be possible to implement the same thing without using daemons?

A. Sure.

EX2020 (Long Depo)
at 128:5-7

Q. (By Mr. Limbacher) Can you give an example of a way to do so without using daemons?

...

A. Well, each application itself, instead of relying on a daemon for gathering information, could do its own instrumentation for the information back.

You don't need a daemon to collect and collate the information; you could have that done by, say, the application manager itself or the resource manager itself. Daemons are a nice way to structure code.

EX2020 (Long Depo)
at 128:9-21

CN Daemons and BN Daemons

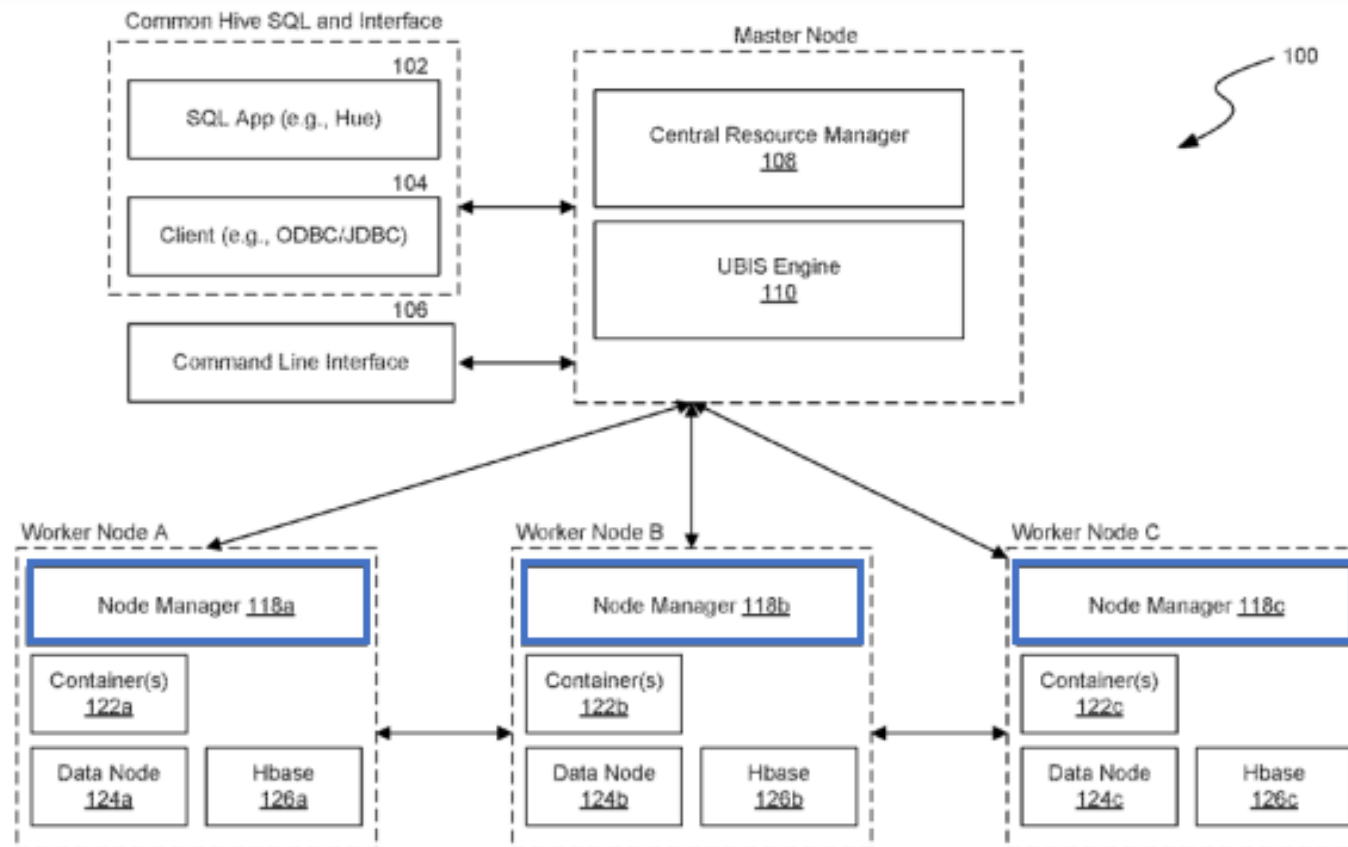
90. The term “daemon” is sometimes used to refer to as a background process, but a daemon operating in one context will not function the same (or be obvious based on) a daemon operating in another context. The context and operation of the nodes are important. In the context of claim 5, there is one set of daemons operating in the computation nodes and one set of daemons operating in the booster nodes. Those daemons generate the “information relating to the processing of the plurality of sub-tasks by at least the portion of the plurality of computation nodes and at least the portion of the plurality of booster nodes” recited in 442 patent claim 1. EX1001 at 5:37-40.

91. I also note that using daemons was not the only way to generate information in computation nodes and booster nodes. For example, dedicated hardware could be used to generate such information without the use of daemons operating in computation nodes and booster nodes. I understand that when asked to

CN Daemons and BN Daemons-Petition Grounds

Section VII.C.5. Moreover, Kambatla provides further implementation details by expressly disclosing a “resource manager daemon installed at worker nodes in a distributed computing cluster.” EX1006, [0037]. The “resource manager daemon

Ground 2:



CN Daemons and BN Daemons

Q. Would the daemons in the computation nodes be different than the daemons in the booster nodes?

MR. CLAWSON: Objection. Form.

THE WITNESS: The answer is maybe. In the case of the Xeon Phi, probably not. In the case of a truly heterogeneous system, almost certainly yes.

EX2020 (Long Depo) at 53:16-21

Outline of Petition Deficiencies

Petition Deficiencies

- **Claims 1 and 9 (All Grounds)**
 - **Further Distribution of Sub-Tasks between CN's and BN's**
 - Budenske not Enabling
 - **Generate Further Distribution Using Information Related to Processing**
 - **Budenske's Dynamic Parameters Not Process-Related Information**
 - **Petition Fails to Address Petitioner's District Court Constructions**
- **Claim 3 (All Grounds) — Dynamically Alters Further Distribution**
- **Claims 5 and 9 (All Grounds) — CN & BN Daemons provide claimed information**
- **All Claims (All Grounds) — No Motivation to Combine**

No Motivation to Combine

- A resource manager module implemented as a daemon, as taught by Lippert (*e.g.*, EX1004, [0014]), arranged to:
 - Assign booster nodes to computation nodes for processing a computation task, as taught by Lippert (*e.g.*, EX1004, [0014]);
 - Assign, according to an initial mapping, sub-tasks of the computation task to the nodes for processing in a first iteration of the computation task, as taught by Budenske (*e.g.*, EX1005, 401-02); and
 - Dynamically assign, during processing of the computation task, computation nodes and booster nodes to each other, as taught by Lippert (*e.g.*, EX1004, [0013]);

- An application manager module, as taught by Budenske (*e.g.*, EX1005, 388, 402), arranged to:
 - Deploy the sub-tasks to the computation nodes and the booster nodes, as assigned for a current iteration, for execution of the sub-tasks by the nodes in the current iteration, as taught by Budenske (*e.g.*, EX1005, 402);
 - Monitor execution of sub-tasks to collect information relating to processing of the sub-tasks in the current iteration, as taught by Budenske (*e.g.*, EX1005, 402); and
 - Assign, according to a new mapping obtained using the information related to the processing of the sub-tasks, the sub-tasks to the nodes for a subsequent iteration of the computation task, as taught by Budenske (*e.g.*, EX1005, 401-02).