

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICROSOFT CORPORATION,
Petitioner,

v.

PARTEC AG,¹
Patent Owner.

IPR2025-00318
Patent 11,537,442 B2

Before JAMESON LEE, MITCHELL G. WEATHERLY, and
JASON J. CHUNG, *Administrative Patent Judges*.

LEE, *Administrative Patent Judge*.

DECISION
Granting Institution of *Inter Partes* Review
35 U.S.C. § 314

¹ ParTec Cluster Competence center GmbH changed its legal name to ParTec AG. Paper 4, 2.

I. INTRODUCTION

Microsoft Corporation (“Petitioner”) filed a Petition for *inter partes* review of claims 1–10 (“challenged claims”) of Patent No. 11,537,442 B2 (“the ’442 Patent”). Paper 1 (“Pet.”). ParTec AG (“Patent Owner”) filed a Preliminary Response. Paper 7 (“Prelim. Resp.”). Patent Owner informed the Board that it had changed its name from ParTec Cluster Competence Center GmbH, as identified by Petitioner on the Petition, to ParTec AG. Paper 4, 2.²

We have authority to determine whether to institute an *inter partes* review. *See* 35 U.S.C. § 314 (2018); 37 C.F.R. § 42.4(a). An *inter partes* review may not be instituted unless the information presented in the Petition “shows that there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition.” 35 U.S.C. § 314(a). The “reasonable likelihood” standard is “a higher standard than mere notice pleading” but “lower than the ‘preponderance’ standard to prevail in a final written decision.” *Hulu, LLC v. Sound View Innovations, LLC*, IPR2018-01039, Paper 29 at 13 (PTAB Dec. 20, 2019) (precedential).

II. BACKGROUND

A. *Real Parties in Interest*

Petitioner identifies Microsoft Corporation as real party in interest. Pet. vii. Patent Owner identifies ParTec AG as real party in interest. Paper 4, 2

² Starting with this paper, we will identify Patent Owner in the case caption as ParTec AG. The parties shall do the same in future filings.

B. Related Matters

The parties identify the following litigation as a related matter:

ParTec AG et al. v. Microsoft Corp., Case No. 2:24-cv-00433
(E.D. Tex.).

Pet. vii; Paper 4, 2.

C. The '442 Patent

The '442 patent issued from Application No. 16/963,749, which has a PCT filing date of January 23, 2019, from PCT Application No. PCT/EP2019/051615, and claims priority to EP 18152903, filed January 23, 2018. Ex. 1001, codes (21), (22), (30). (86).

The '442 patent pertains to a heterogeneous computing system comprising a plurality of computation nodes and a plurality of booster nodes. Ex. 1001, code (57). The heterogeneous nature of the system derives from its having different types of processing circuits, i.e., computation nodes on the one hand and booster nodes on the other. *See id.* at 1:23–45. The '442 patent states:

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.

Ex. 1001, 1:66–2:8.

Initially, in a first iteration of the plurality of sub-tasks, a resource manager determines the assignment of tasks and sub-tasks to the computation nodes and the booster nodes for the first iteration, as a function of the computation task and other parameters. Ex. 1001, 2:19–22.

Thereafter, the distribution of the sub-tasks among the computation and

booster nodes is updated, i.e., changed, for subsequent iterations of the sub-tasks, based on information received from background daemons operating in the computation and booster nodes. *Id.* at 2:44–50. The ’442 patent describes that the information provided by the background daemons in the computation nodes and booster nodes “allows an application manager to determine if the distribution of sub-tasks between computation nodes and booster nodes can be adapted or improved upon for a further computing iteration [of the sub-tasks].” *Id.* at 2:15–18.

Claims 1 and 9 are independent and are reproduced below:³

[1.1]. 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:

[1.2] 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

[1.3] 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.

³ The bracketed headings correspond to those used by Petitioner to reference the claim elements. *See* Pet. 23, 27, 29, 49–52. We use the same headings here for ease of reference, understanding, and consistency.

Ex. 1001, 5:26–43.

[9.1]. A heterogeneous computing system comprising:

[9.2] a plurality of computation nodes and a plurality of booster nodes for computing one or more tasks comprising multiple sub-tasks;

[9.3] a communication interface connecting the plurality of computation nodes with each other and the plurality of booster nodes;

[9.4] 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

[9.5] 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.

Id. at 6:26–43.

D. The Applied Prior Art and Declarations

Petitioner relies on the following references:

Name	Patent Document/Publication	Exhibit
Lippert ⁴	US Pat. Pub. 2013/0282787 A1	1004
Kambatla ⁵	US Pat. Pub. 2018/0074855 A1	1006

⁴ Published October 24, 2013. Ex. 1004, code (43). Petitioner asserts Lippert qualifies as prior art under 35 U.S.C. § 102(a)(1). Pet. 11.

⁵ Published March 15, 2018. Ex. 1006, code (43). Petitioner asserts Kambatla qualifies as prior art under 35 U.S.C. § 102(a)(2). Pet. 53.

Name	Patent Document/Publication	Exhibit
Budenske ⁶	Budenske et al., <i>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</i> , The Journal of Supercomputing, Vol. 12, No. 4, 387–406.	1005

Petitioner relies on the Declaration of Darrell Long, Ph.D. Ex. 1003.
Patent owner relies on the Declaration of David R. Kaeli, Ph.D. Ex. 2009.

E. The Asserted Grounds of Unpatentability

Petitioner asserts that the challenged claims of the '442 patent are unpatentable based on the following grounds (Pet. 2):

Claims Challenged	35 U.S.C. §	References/Basis
1–10	103 ⁷	Lippert, Budenske
2–5, 8–10	103	Lippert, Budenske, Kambatla

III. ANALYSIS

A. Level of Ordinary Skill in the Art

Petitioner asserts that a person of ordinary skill in the art “would have held a bachelor’s degree in computer science, electrical engineering, computer engineering, or a closely related field, and approximately three

⁶ Published October 1998. Ex. 1005, iv. Petitioner asserts Budenske qualifies as prior art under 35 U.S.C. § 102(a)(1). Pet. 13.

⁷ The Leahy-Smith America Invents Act (“AIA”), Pub. L. No. 112–29, 125 Stat. 284, 287–88 (2011), amended 35 U.S.C. §§ 102 and 103. Because the '442 patent does not claim priority to an application filed before March 16, 2013 (the effective date of the amendments), the AIA version of § 103 applies.

years of experience working with distributed computing technologies.”

Pet. 8. Petitioner further asserts that such education and experience “would have included designing, implementing, and maintaining distributed computing systems, such as cluster computing systems, and deploying applications on such systems.” *Id.*

Patent Owner states that for purposes of the Preliminary Response it has applied Petitioner’s recitation of the level of skill in the art. Prelim. Resp. 15. Patent Owner has not proposed its own articulation of the level of ordinary skill.

We adopt Petitioner’s statement of the level of ordinary skill in the art. It is supported by the testimony of Dr. Long and not disputed by Patent Owner. Further, it appears consistent with what is reflected by the content of the applied prior art. *Cf. Okajima v. Bourdeau*, 261 F.3d 1350, 1354–55 (Fed. Cir. 2001) (the applied prior art may reflect an appropriate level of skill).

B. Claim Interpretation

We use the same claim construction standard that would be used to construe a claim in a civil action under 35 U.S.C. § 282(b), including construing the claim in accordance with the ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent. 37 C.F.R. § 42.100(b). The claim construction standard set forth in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc) is applicable.

Claim terms are generally given their ordinary and customary meaning as would be understood by one with ordinary skill in the art in the context of the specification, the prosecution history, other claims, and

extrinsic evidence including expert and inventor testimony, dictionaries, and learned treatises, although extrinsic evidence is less significant than the intrinsic record. *Phillips*, 415 F.3d at 1312–17. Usually, the specification is dispositive, and it is the single best guide to the meaning of a disputed term. *Id.* at 1315.

If an inventor acts as his or her own lexicographer, the definition must be set forth in the specification with reasonable clarity, deliberateness, and precision. *Renishaw PLC v. Marposs Societa' per Azioni*, 158 F.3d 1243, 1249 (Fed. Cir. 1998). Only those claim terms that are in controversy need to be construed, and only to the extent necessary to resolve the controversy. *Realtime Data, LLC v. Iancu*, 912 F.3d 1368 (Fed. Cir. 2019) (“The Board is required to construe ‘only those terms . . . that are in controversy, and only to the extent necessary to resolve the controversy.’” (quoting *Vivid Techs., Inc. v. Am. Sci. & Eng'g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999))).

Petitioner has not proposed an express construction for any claim term. Neither has Patent Owner. For purposes of this Decision, we need not expressly construe any claim term.

*C. Alleged Obviousness of Claims 1–10
over Lippert and Budenske*

1. Overview of Lippert (Exhibit 1004)

Lippert discloses a heterogeneous computer cluster arrangement including both computation nodes and booster nodes (also referred to in Lippert as accelerator nodes). Ex. 1004 ¶¶ 6–12. Specifically, there are a plurality of computation nodes, at least two of which are arranged to jointly compute a first part of the computation task. *Id.* ¶ 10. Also, at least one booster node is arranged to compute at least a second part of the computation task. *Id.* ¶ 11. A resource manager assigns “at least one

booster to at least one of the plurality of computation nodes for computation of the second part of the computation task,” and the assignment is accomplished “as a function of a predetermined assignment metric.” *Id.*

¶ 12.

Figure 2 of Lippert is reproduced below:

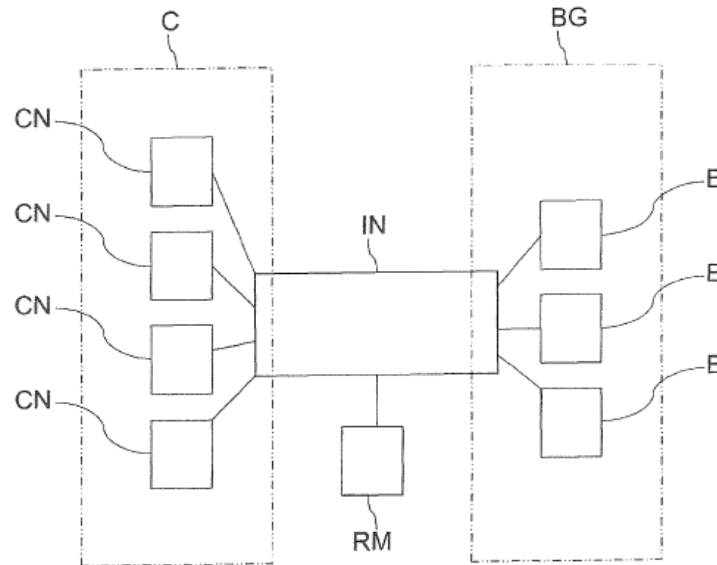


Fig.: 2

Figure 2 is a schematic illustration of a computer cluster arrangement according to Lippert. Ex. 1004 ¶ 59.

The heterogeneous computer cluster arrangement shown in Figure 2 includes 4 computation nodes CN, and 3 booster nodes B. Ex. 1004 ¶ 67.

Each of the booster nodes B can be shared by any of the computation nodes CN. *Id.* Lippert describes:

In the present embodiment, computation tasks are processed by at least one of the computation nodes CN and at least a part of the computation tasks may be forwarded to at least one of the boosters B. The boosters B are arranged to compute specific problems and provide specific processing power. hence, problems can be outsourced from one of the computation nodes

CN to the boosters B, be computed by the booster and the result may be delivered back to the computation node. The assignment of boosters [B] to computation nodes CN can be accomplished by a resource manager, also referred to as RM. The resource manager initializes a first assignment and further on establishes a dynamic assignment of boosters B to computation nodes CN.

Id. ¶ 68.

2. *Overview of Budenske (Exhibit 1005)*

Budenske describes a heterogeneous parallel hardware platform providing processing for an application domain involving iterative tasks. Ex. 1005, 387. Specifically, it provides a methodology for remapping of the application subtasks to the processors in the heterogeneous parallel hardware platform for subsequent iterations of the tasks, during execution of the application. *Id.* Budenske states: “This work is being developed for a class of ATR applications each of which can be modeled as an iterative execution of a set of partially ordered subtasks. Each ATR application in this class is a production job that is executed repeatedly.” *Id.* at 390.

Budenske explains:

For the initial iteration through the set of subtasks, the IOS will use information about the processing environment in its selection of algorithms for the subtasks, and their associated implementations. As part of this, the IOS will decide how to assign resources (e.g., processors) to the subtasks.

After each execution iteration through the set of subtasks, the values of certain dynamic parameters of the application may change, such as the number of objects detected in the current frame of a real-time image stream being processed. It is expected that the values of these parameters will change slowly. After all subtasks have completed execution for a given iteration through the DDG, and before the next iteration begins, the latest values of these dynamic parameters will be reported to the on-line HC Kernel. The HC Kernel will use the most recent values of such dynamic parameters to estimate if it is worthwhile to reconfigure

the assignment of processing resources to subtasks to reduce execution time of the next iteration. If it is desirable, the HC Kernel will select a new assignment to use for the next execution iteration through the subtasks. If not, the same assignment will continue to be used.

Ex. 1005, 391. In similar vein, Budenske describes the following regarding the HC Kernel Monitor:

The HC Kernel Monitor is the on-line component of the IOS (Figure 3) responsible for (1) establishing the initial mapping of the given application onto the hardware platform, and (2) monitoring the execution of the application and, at the end of each iteration through the corresponding DDG, deciding if and how the mapping of the application onto the hardware platform should be changed based on information about the actual values of the dynamic parameters.

Id. at 401–402.

3. *Petitioner’s Manner of Combining Lippert and Budenske*

Petitioner asserts that “[a] POSITA [person of ordinary skill in the art] would have been motivated, with a reasonable expectation of success, to combine Lippert’s teachings on a computer cluster arrangement with Budenske’s teachings on re-mapping sub-tasks between iterations of an iterative application.” Pet. 14. Petitioner states that it proposes to apply the following teachings of Budenske to Lippert:

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., EX 1005, 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).

Pet. 16 (citing Ex. 1003 ¶ 57).

Regarding motivation to combine, Petitioner explains:

A POSITA would have been motivated to implement the heterogeneous computing system of Lippert with the dynamic iterative sub-tasks assignment method of Budenske partly because doing so would facilitate efficient distributions of subtasks and processing resources during execution of an iterative application, thus fully leveraging the flexibility of the heterogeneous architecture and “reduc[ing] execution time” of computing iterations. EX1005, 391; EX1003, ¶59.

Pet. 17. Petitioner further explains:

Second, Budenske provides express motivation from implementing its method in a heterogeneous computing system such as Lippert, namely to “fully utilize the architectural flexibility of such a [heterogeneous parallel computing] system.” EX1005, 390. A POSITA would have found that the dynamic flexibility of the heterogeneous computing system taught by Lippert through the “loose coupling” of computation nodes and boosters (EX1004, [0013], [0015] (describing “a maximum flexibility in providing hardware resources”)), naturally complements the dynamic flexibility of remapping sub-tasks among processing resources to account for the actual processing of the sub-tasks by the processing resources during preceding iterations, as taught by Budenske (EX1005, 391), and that the combination for these teachings would provide increased flexibility for optimally processing iterative applications. EX1003, ¶60.

Id. at 17–18. Petitioner additionally explains:

Third, ... A POSITA would have recognized that Lippert and Budenske follow the same principles of efficient processing of a task, comprising a plurality of sub-tasks, with a heterogeneous computing system, by assigning sub-tasks to processors (e.g., computation nodes or boosters) best suited to execute the sub-task. A POSITA thus would have been motivated

to implement Budenske's teachings with a heterogeneous computing system including boosters as taught by Lippert, partly because "the boosters are implemented to process specific problems at high speed." EX1004, [0028]; EX1003, ¶61.

Id. at 18.

Petitioner provides more reasons why one of ordinary skill in the art would have combined the teachings of Lippert and Budenske as Petitioner has proposed:

Fourth, a POSITA reading Lippert would have understood the need to account for sub-task assignment among nodes for iterative applications, which is taught by Budenske. While Lippert focuses on the configuration of the processing resources of the system itself, Lippert only broadly and generally discusses the nature of the "application" or "computation task" being computed. E.g., EX1004, [0015]-[0016]. . . . A POSITA thus would have naturally turned to Budenske when implementing Lippert for iterative applications, in order to "assign each . . . subtask to the processors where it is best suited for execution," and thus minimize execution time as the processing requirements of the input data and/or computation task evolves over time. *Id.*, 390; EX1003, ¶62.

Fifth, a POSITA would have recognized the compatibility of Lippert and Budenske's teachings. Lippert, for example, recognizes that a computation task can be divided into at least a first party for computing by computation nodes, and a second art for computation by boosters. EX1004, [0009]-[0014], [0016]-[0020], [0068]. Further, a POSITA would have understood that the "first part of the computation task" computed by the computation nodes includes the less scalable, complex sub-tasks, while the "second part of the computation task" computed by the booster(s) includes the highly scalable sub-tasks. *Id.*; EX1014, 5. Further, Lippert recognizes that "[e]ach code has highly scalable and less scalable complex elements," and so a POSITA would have been motivated to change which sub-tasks are assigned to a first part (computed by computation nodes) and a second party (computed by boosters), regardless of a sub-task's

initial classification as complex or scalable (and thus initial mapping to computation nodes or boosters), if information relating to the processing of the sub-task in a preceding iteration suggests that the sub-task would be better processed by a different node. EX1003, ¶63.

Sixth, nothing in these references teaches away from this combination. On the contrary, a POSITA would have recognized that the heterogeneous computing systems of Budenske and Lippert are similar. Specifically, Budenske discloses a heterogeneous computing system including DSP (digital signal processor) and RISC (reduced instructions et computer) processors (EX1005, 391), and a POSITA would have recognized that the DSPs are processors that function as accelerators/boosters and the RISC processors as computing nodes, much like the processing elements in Lippert. A POSITA thus would have understood that the teachings of Lippert and Budenske are compatible and would work well together. EX 1003, ¶64.

Id. at 19–21.

Finally, Petitioner asserts that a POSITA would have reasonably expected to succeed in implementing the Lippert-Budenske combination because (1) “doing so would only entail providing software code, which was well within the skill of a POSITA”; (2) there is nothing incompatible between the two approaches, and because Budenske explains that its teachings “can also be used for other application domains and classes of hardware platforms whose characteristics are similar to those of the application and platforms considered here”; (3) other prior art references available to the POSITA disclose successful implementations of Budenske’s teachings and confirm that Budenske’s approach to re-mapping subtasks is feasible and achieves faster execution times compared to other dynamic task scheduling techniques. Pet. 21–22.

Petitioner additionally notes that the '442 patent provides only a high-level description of adjusting the distribution of sub-tasks between computation nodes and boosters, which assumes that anyone desiring to carry out the process would know the equipment and techniques to be used. *Id.* at 22. According to Petitioner, this general level of disclosure “confirms that POSITAs would have reasonably expected to succeed implementing Lippert-Budenske.” *Id.* at 22–23 (citing Ex. 1003 ¶ 66). Petitioner further asserts: “Lippert and Budenske each provide ample implementation details for achieving the disclosed functionality, and a POSITA would have would have found it simple and straightforward to combine the teachings. EX1003, ¶79.” *Id.* at 32.

4. *Independent Claim 1*

Claim 1’s preamble element [1.1] reads: “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.” Ex. 1001, 5:26–32.

Petitioner asserts that Lippert discloses “operating a heterogeneous computing system comprising a plurality of computation nodes and a plurality of booster nodes, with at least one of the nodes being arranged to compute a computation task comprising a plurality of sub-tasks.” Pet. 24. Petitioner explains:

Specifically, Lippert discloses “computer cluster arrangement for processing a computation task,” where the “computer cluster arrangement” comprises a plurality of computation nodes and a plurality of boosters (booster nodes). EX1004, [0009]-[0013]. Processors applied in computation nodes are different from the

processors applied in boosters. *Id.*, [0029]. A POSITA thus would have understood that the computer cluster arrangement is a heterogeneous computing system because the computer cluster arrangement comprises different types of processors, arranged as computation nodes and booster nodes.

Pet. 24. Petitioner further explains: “Lippert discloses that ‘the computation task may comprise several subproblems, also referred to as sub tasks, which in their entirety describe the overall computation task.’ EX1004, [0016].

The computation task may be divided ‘into several parts,’ and the computer cluster arrangement is arranged ‘to solve the parts of the computation task in parallel or in succession.’ *Id.*, [0016]; EX1003, ¶69.” *Id.*, at 25.

Petitioner refers to Lippert’s Figure 2, reproduced below:

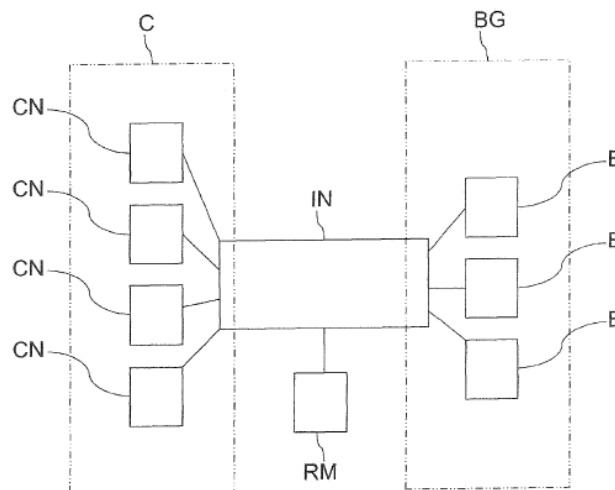


Fig.: 2

Figure 2 is a schematic illustration of a computer cluster arrangement according to Lippert. Ex. 1004 ¶ 59. Petitioner states that Figure 2 shows a computer cluster arrangement comprising a cluster C and a booster group BG, where the cluster includes 4 computation nodes CN and 3 booster nodes B. Pet. 24 (citing Ex. 1004 ¶ 67).

Petitioner's assertions regarding element [1.1] are supported by the cited evidence and not disputed by Patent Owner. We are sufficiently persuaded that Lippert discloses element [1.1].

Limitation [1.2] reads: "in a first computing iteration, assigning and processing the plurality of subtasks 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." Ex. 1001, 5:33–36.

Petitioner explains:

Lippert discloses computing a "first part of a computation task" and a "second part of the computation task" (*the plurality of subtasks*) by, respectively, "at least two of the plurality of computation nodes CN" (*at least a portion of the plurality of computation nodes*) and "at least one booster B" (at least a portion of the plurality of booster nodes). EX1004, [0091]. A POSITA would have understood this computing of the parts of the computation task by the at least two of the plurality of computation nodes and the at least one booster as *assigning and processing* the parts of the computation task *in a first distribution*, at least because Lippert teaches "establish[ing] a static assignment [of a booster to a computation node] at start of a processing of a computation task," where "assignment information [is provided] to the computation nodes for outsourcing parts of the computation tasks from at least one computation node to at least one booster." *Id.*, [0013]-[0014]; EX1003, ¶73.

Pet. 26. Petitioner further explains:

Although Lippert discloses the computation node "outsourcing" parts of the computation task to a booster where the computation node processes the first part in a first step and the booster node processes the second part in a second step (e.g., EX1004, [0091]-[0092]), suggesting that the parts are solved in succession, Lippert discloses also that the system may "solve the parts of the computation task in parallel or in succession." *Id.*, [0016] (emphasis added). Further, Lippert does not expressly

specifically disclose iterative execution of a computation task, and thus does not expressly specifically disclose a “first computing iteration,” but Lippert more generally explains that “the person skilled in the art appreciates that any of the steps [in FIGS. 4 and 5] can be performed iteratively, in a different order and may comprise further substeps.” *Id.*, [0093]; EX1003, ¶74.

Id. at 27–28. Figure 4 shows that in step 100 a first part of the computation task is performed by at least two CNs, and in step 101 a second part of the computation task is performed by a booster B. Ex. 1004 ¶ 91. We understand Petitioner as referring to Lippert describing these two steps as capable of being performed in parallel and iteratively. Pet. 27–28.

Petitioner’s assertions regarding limitation [1.2] are supported by the cited evidence and not disputed by Patent Owner. We are sufficiently persuaded that Lippert discloses limitation [1.2].

Limitation [1.3] reads: “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.” Ex. 1001, 5:37–43.

For this limitation, Petitioner relies on its proposed combination of Lippert and Budenske, i.e., applying Budenske’s sub-task mapping in Lippert’s heterogeneous computing system. Pet. 14–23, 29–33. Petitioner explains:

Budenske teaches “monitor[ing] 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).” 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. 29–30. Petitioner asserts:

Budenske thus teaches selecting a new mapping from a set of candidate mappings that best matches the values of the dynamic parameters (*the information*), and instantiating (*generating*) the further distribution according to the selected mapping for the next computing iteration. *Id.*, 402-03. Budenske also discloses an alternative wherein a fully “on-line heuristic” is used, such that the new mapping is generated in real-time between iterations based on the information. *Id.*, 389-90, 398, 403; EX1003, ¶78.

Id. at 31–32.

Regarding motivation to combine teachings, Petitioner presents multiple reasons, identified and discussed above in Section III.C.3, why one of ordinary skill in the art would have wanted to apply Budenske’s manner of re-mapping sub-tasks to Lippert’s heterogeneous computing system between iterations of an iterative application. Adding emphasis, Petitioner asserts:

The POSITA would have been motivated to implement Lippert-Budenske in this way partly in order “to reduce execution time of the next iteration.” EX1005, 391. The POSITA would have further been motivated to implement Lippert-Budenske in this way because Budenske explains that, “[b]ecause the execution time of application subtasks . . . is highly input-data dependent . . . , this matching and scheduling of application subtasks to processors must be performed dynamically at run time.” *Id.*, 390 (emphasis added).

Pet. 32.

Regarding reasonable expectation of success, Petitioner provides multiple explanations, identified and discussed above in Section III.C.3, why one of ordinary skill in the art would have had a reasonable expectation of success in combining the teachings of Lippert and Budenske in the manner Petitioner proposes.

Petitioner's assertions and explanations are rational and firmly supported by the cited evidence. Notwithstanding Patent Owner's arguments to the contrary, discussed below, we determine that the combined teachings of Lippert and Budenske as proposed by Petitioner disclose limitation [1.3], and that Petitioner has presented proper and sufficient motivation that one of ordinary skill in the art would have combined the teachings of Lippert and Budenske in the manner proposed by Petitioner, and with a reasonable expectation of success.

For limitation [1.3], Patent Owner argues that "Budenske fails to suggest and is incapable of distributing sub-tasks between different types of processors, much less of performing such a distribution of sub-tasks among different processor types in a further computing iteration." Prelim. Resp. 17–18. To explain its position, Patent Owner asserts:

Budenske expressly states that for its teachings to work "it is assumed that if an implementation of a given subtask uses multiple processors, **all processors will be of the same type.**" EX1005 at 392 (emphasis added). By restricting execution to a single processor type, Budenske can avoid dealing with the differences in execution properties and speeds between processor types and instead treat "the expected execution time of a particular multiprocessor implementation of a subtask [as] independent of which fixed-sized subset of the processors **of a given type** are assigned to execute the subtask." *Id.* (emphasis added); EX2009 ¶¶49, 56.

Prelim. Resp. 18. We have read the cited portions of Budenske and disagree with Patent Owner's characterization of the cited description. Budenske does not suggest, much less expressly state, that "for its teachings to work," as Patent Owner has characterized, it is assumed that if an implementation of a given subtask uses multiple processors, all processors will be of the same type. The actual quotation is: "For simplicity, it is assumed that if an implementation of a given subtask uses multiple processors, all processors will be of the same type." Ex. 1005, 392. That is far different from Patent Owner's proposed reading, which amounts to disclaiming or nullifying all disclosures of using a combination of different types of processors to perform a subtask. The description does not support Patent Owner's assertion. On this record, in our view, the description is merely providing a simplified version to illustrate the concept of an initial assignment and subsequent reassignments for performing subtasks in multiple iterations, and does not undermine or otherwise negate Budenske's teachings about using a combination of different types of processors to perform a subtask.

Patent Owner further argues:

As Dr. Kali explains, in addition to the express statements from Budenske that is limited to processors for a single type, a POSITA would also have understood the same restriction relative to Budenske from the discussion of its architecture. EX2009 ¶¶48-49. For example, Petitioner refers to the SHARC DSPs of Budenske as the accelerator/boosters. Pet. 21. Budenske explains that, under its architecture, those DSPs will "physically share a DRAM." EX1005 at 392. Dr. Kaeli notes that in the context of Budenske, a POSITA would understand that those DSPs "all share the same working memory, and if five processors were needed instead of four for the next iteration (such as when the number of objects of interest in the last framed changed from four to five—see example on page 393), all DSP processors would have access to the same data set and could take

over the extra processing load. The same would not be true of different processor types (such as the RISC processors) which would not have direct access to that shared memory pool.” EX2009 ¶49. Stated more simply, the Budenske architecture implements a shared memory that requires the processor types to be the same. The net effect is that a skilled artisan would conclude from Budenske that mixing processor types for particular sub-tasks would be difficult to implement and would require further innovations.

Budenske continues that same line of reasoning, emphasizing that processors of the same type will have symmetric and conflict-free intercommunication. EX1005 at 392. Budenske also emphasizes that expected execution times are independent of the assigned processor because it requires assignment to a single type of processor. *Id.* (“[T]he expected execution time of a particular microprocessor implementation of a subtask is independent of which fixed-size subset of the processors **of a given type** are assigned to execute the subtask.”). As Dr. Kaeli explains, all of this would indicate to a POSITA that Budenske could not assign a subtask to more than one type of processor. EX2009 ¶¶48-49.

Prelim. Resp. 18–19. The above-quoted arguments are equally as unavailing as the earlier argument (Prelim. Resp. 18), discussed above, that Budenske expressly states that for its teachings to work “it is assumed that if an implementation of a given subtask uses multiple processors, **all processors will be of the same type.**” Where, as here, Budenske states that for simplicity purposes it will make a certain assumption in its illustrative examples and disclosures, it is entirely expected that further description in Budenske would be consistent and in conformance with that assumption. As noted above, the assumption of a simplified circumstance for purposes of providing an example or illustration does not disclaim or nullifying disclosures of Budenske on using different types of processors to perform a subtask.

Furthermore, we see nothing in claim 1, and Patent Owner has not shown any limitation of claim 1, which requires the booster nodes themselves to comprise different types of processors. The heterogeneous computing system of claim 1 is met by computation nodes and booster nodes being comprised of different types of processors relative to each other.

Patent Owner asserts:

While Budenske suggests that its disclosure addresses “how to assign resources (e.g., processors) to the subtasks” generally, the fact that the authors felt the need to expressly limit their discussion to using only one type of processor for any given sub-task speaks volumes of the difficulty of distributing sub-tasks between disparate types of processors. *See* EX1005 at 391. This difficulty would have lead a POSITA to conclude that distributing subtasks between different types of processors was too difficult of a problem to have been solved without further innovations. EX2009 ¶¶28-29, 35-39.

Prelim. Resp. 19–20. The argument is unavailing. The fact that there are easier implementations and harder implementations does not establish that a harder implementation necessarily “was too difficult of a problem to have been solved” by one of ordinary skill. Patent Owner shows nothing in Budenske which indicates that the only way the disclosed method works is if all the processors of the booster nodes are of the same type or that all the processors of the computation and booster nodes are of the same type.

Patent Owner further argues:

Further confirming the difficulty of mapping subtasks to different processor types, Dr. Kaeli cites several publications which discuss the difficulty of mapping to multiple types of processors, including an article addressing the difficulty of mapping to multiple different types of GPUs resulting in what it refers to as a “straggler problem” that results in “lower GPU utilization”, EX2009 ¶36 (citing EX2014 at 179-180). He also cites another article which notes that code that must utilize two

different architectures (e.g., general purpose processor as well as a booster) “introduces significantly more complexity than was seen under [] other [] historical models, ‘leading to a dissonance with these traditional HPC system workloads.’” EX2009 ¶ 37 (quoting EX2015 at 1-2). These publications further confirm that a POSITA would not have understood the Budenske disclosure to teach mapping of subtasks to different types of processors at the same time.

Prelim. Resp. 20. The argument is misplaced because it improperly equates relative complexity and even lack of perfection to unworkability. Patent Owner points to nothing indicating that mapping tasks to different types of computers could not have been done by one of ordinary skill in the art. We disagree with Patent Owner’s assertion that “a POSITA would not have understood the Budenske disclosure to teach mapping of subtasks to different types of processors at the same time.”

Phrasing its previous arguments another way, Patent Owner asserts a “teaching away” argument: “As discussed above, Budenske actually teaches away from distributing subtasks between processor types, noting that for its teachings to work all processors utilized for a given task must be of the same type. EX1005 at 392; Section VI.A.1.” Prelim. Resp. 23. We already discussed and rejected above Patent Owner’s contention that Budenske describes its teachings would work only if all processors utilized for a given task are of the same type. Thus, the “teaching away” argument also lacks merit. Alternatively, Patent Owner asserts that even if we reject Patent Owner’s “teaching away” argument, “Budenske’s teachings would have strongly weighed against motivating a skilled artisan to develop a system according to the ‘442 Patent claims.” Prelim. Resp. 23–24. We disagree. As discussed above, Budenske acknowledged that its resource mapping method would be more easily accomplished if all the processors were of the

same type. It nowhere indicates, as Patent Owner argues, that the method would work only if all the processors were of the same type.

Patent Owner asserts that one of ordinary skill in the art would not have had a reasonable expectation of success in implementing Budenske's mapping teachings, as alleged by Petitioner, in other prior art references.

Prelim. Resp. 24. In that regard, Patent Owner explains:

One of the Authors of Budenske (Howard Jay Siegel) noted in a subsequent publication that the purported methods of mapping tasks using "dynamic parameter values" in Budenske were not "ever evaluated in any way, as this was not the focus of Budenske et al." EX1007 at 79. Rather, the focus of Budenske was strictly on "[t]he design of the ATR Kernel and HC kernel." *Id.* Thus, the authors of Budenske recognized and stated publicly that the "dynamic parameter values" and purported mappings that Petitioner relies on were not evaluated or expected to be usable, and a POSITA would have had no expectation of success in using them as taught by Budenske, much less in modifying a combination of Lippert and Budenske to use them in a manner that no reference of record actually taught was possible. The lack of an expectation of success is further reinforced by references cited by Dr. Kaeli that emphasized the difficulty of mapping to processors of different types. *See* Section VI.A.1; EX2009 ¶¶28-29, 35-39.

Prelim. Resp. 24–25. The cited text in Exhibit 1007 does not support Patent Owner's assertion. The referenced publication nowhere states or even suggests that the dynamic parameter values in Budenske were not expected to be usable or adaptable to be usable if evaluated for that purpose. The assertion that "no reference of record actually taught was possible" is directly refuted by Budenske itself, whose disclosure is not limited to tasks being performed only by processors of the same type, as we explained above. With respect to various references cited by Dr. Kaeli, Patent Owner improperly equates relative difficulty to non-performability. Patent Owner

further does not explain what the inventors of the '422 patent discovered or possessed which made a previously allegedly non-performable task possible.

Patent Owner still further argues:

But even in the heterogeneous architecture contemplated by Kambatla, the reference noted that “[t]he amount of computing resources required to process a given task can be difficult to predict. It is inevitably difficult to accurately estimate the resource requirements of a job or its constituent tasks because: (i) resource usage of a task varies over time, and (ii) resource usage can vary across tasks of the same job based on the input they process.” EX1006 at [0005]. Those challenges would be compounded even further in the context of a heterogeneous architecture, wherein resource usage of a task varies not just by time and input, but based on the varied capabilities of different processor types as well. EX2009 ¶ 67. Thus, even years after Budenske, Petitioner’s other references continued to reinforce the difficulty of estimating resource usage and mapping tasks (let alone mapping sub-tasks across disparate processor types during subsequent computing iterations).

Prelim. Resp. 25. Patent Owner again improperly equates and conflates relative complexity to non-performability. Patent Owner specifically identifies nothing in Kambatla which indicates the amount of computing resources required for a task simply could not have been reasonably estimated by one of ordinary skill in the art and thus would not have been ascertainable. We note further that Petitioner’s proposed combination need not be capable of achieving a perfect estimation. A reasonable assessment, while leaving room for later improvement, sufficiently establishes a reasonable expectation of success.

Additionally, Patent Owner presents an argument based on the time gap between the work of Bedenske and Lippert/Kambatla. Prelim. Resp. 26–27. Patent Owner asserts:

Even the significant gap in time between Budenske and Lippert/Kambatla demonstrates that there was no motivation to combine the references. Budenske identifies a purported publishing date of 1998, while Lippert claims a priority date in 2010 and Kambatla claims a priority date in 2016. If the proposed combination was a obvious, workable, and beneficial (without the use of hindsight) as Petitioner suggests, then a POSITA would have adapted the teachings of Budenske to Lippert well before the 2018 priority date of the '442 patent. "The length of the intervening time between the publication dates of the prior art and the claimed invention can [] qualify as an objective indicator of nonobviousness." *Leo Pharm. Prods. v. Rea*, 726 F.3d 1346, 1359 (Fed. Cir. 2013). Correspondingly, the "intervening time between the prior art's teaching of the components and the eventual preparation of a successful composition speaks volumes to the nonobviousness of the [patent]." *Id.*; see also *Merck & Co. v. Teva Pharm. USA, Inc.*, 395 F.3d 1364, 1376 (Fed. Cir. 2005) ("[T]he law presumes an idea would successfully have been brought to market sooner, in response top market forces, had the idea been obvious to persons skilled in the art.").

The effects of this gap in time are amplified by the extreme difference in the state of the art when Budenske was published. As Dr. Kaeli notes, at the time Budenske was purportedly published a heterogeneous system was typically customized for a very specific task (automatic target recognition in the case of Budenske). EX2009 ¶¶30-31, 48-49, 66. This meant a task or subtask would (at best) be assigned to the most appropriate processor type, rather than to a mix of processors. *Id.* That again conforms with Budenske's teachings that a subtask would only be assigned to processors of a single type. EX1005 at 392.

Prelim. Resp. 26–27 (emphasis added).

Patent Owner thus circles back to the already rejected contention, discussed above, that Budenske describes that its method would work only if all tasks are assigned to the same type of processors. Further, the argument based on the earlier dates at which Budenske, Lippert, and Kambatla were

created relative to the filing date of the '442 patent is misplaced. The mere age of references is not persuasive of nonobviousness of the combination of their teachings, absent evidence that, notwithstanding knowledge of the references, the prior art tried and failed to solve the problem. *In re McGuire*, 416 F.2d 1322, 1327 (CCPA 1969).

For the foregoing reasons, Petitioner has shown a reasonable likelihood that it would prevail in establishing that claim 1 would have been obvious over Lippert and Budenske.

5. *Independent Claim 9*

Claim 9's preamble element [9.1] reads: "A heterogeneous computing system, comprising." Ex. 1001, 6:26. Petitioner asserts that each of Lippert and Budenske discloses a heterogeneous computing system. Pet. 49 (citing Ex. 1003 ¶ 101). The assertion is supported by the testimony of Dr. Long and not disputed by Patent Owner. We are sufficiently persuaded that each of Lippert and Budenske discloses a heterogeneous computing system.

Limitation [9.2] recites: "a plurality of computation nodes and a plurality of booster nodes for computing one or more tasks comprising multiple sub-tasks." Ex. 1001, 6:26–28. Petitioner asserts Lippert discloses a plurality of computation nodes and a plurality of booster nodes for computing one or more tasks comprising multiple sub-tasks. Pet. 50 (citing Ex. 1003 ¶ 102). The assertion is supported by the testimony of Dr. Long and not disputed by Patent Owner. We are sufficiently persuaded that Lippert discloses limitation [9.2].

Limitation [9.3] recites: "a communication interface connecting the plurality of computation nodes with each other and the plurality of booster

nodes.” Ex. 1001, 6:29–31. Petitioner asserts Lippert discloses limitation [9.3] because Lippert discloses the plurality of computation nodes and the plurality of booster nodes “interfacing a communication infrastructure.” Pet. 50 (citing Ex. 1004 ¶ 10, 11, 16). Petitioner refers to Lippert’s figure 2 as depicting communication infrastructure IN which interfaces with each computation node CN and each booster node B. *Id.* (citing Ex. 1004 ¶ 67). The assertions are supported by the cited evidence and not disputed by Patent Owner. We are sufficiently persuaded that Lippert discloses limitation [9.3].

Limitation [9.4] recites: “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.” Ex. 1001, 6:32–36. Petitioner adequately accounts for this limitation on pages 14–23, 35–41, and 52 of the Petition. Patent Owner argues that one of ordinary skill in the art would not have combined the teachings of Lippert and Budenske in the manner proposed by Petitioner. Prelim. Resp. 23–27. These same arguments have been addressed and rejected above in our discussion of claim 1.

Limitation [9.5] recites: “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 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.” Ex. 1001, 6:37–43. Petitioner adequately accounts for this limitation on pages 14–23, 33, 34, and 52 of the Petition. Patent Owner presents the same arguments it presents with respect to limitation [1.3].

Prelim. Resp. 16–21. For the same reasons explained above in the context of limitation [1.3], the arguments are equally unavailing in the context of limitation [9.5].

Patent Owner additionally argues that one of ordinary skill in the art would not have combined Lippert and Budenske in the manner proposed by Petitioner. Prelim. Resp. 23–27. These same arguments have been addressed and rejected above in our discussion of claim 1.

For the foregoing reasons, Petitioner has shown a reasonable likelihood that it would prevail in establishing obviousness of claim 9 over Lippert and Budenske.

6. *Dependent Claims 2–8*

Claims 2–8 each depend, directly or indirectly, from claim 1. Ex. 1001, 5:44–6:24. Patent Owner does not present arguments for these claims separate from those it submits for claim 1, which we have rejected above in the context of claim 1. We have reviewed Petitioner’s submissions and determine that Petitioner has shown a reasonable likelihood that it would prevail in establishing obviousness of claims 2–8 over Lippert and Budenske.

7. *Dependent Claim 10*

Claim 10 depends from claim 9. Ex. 1001, 6:44–48. Patent Owner does not present arguments for claim 10 separate from those it submits for claim 9, which we have rejected above in the context of claim 9. We have reviewed Petitioner’s submissions and determine that Petitioner has shown a reasonable likelihood that it would prevail in showing obviousness of claim 10 over Lippert and Budenske.

*D. Alleged Obviousness of Claims 2–5 and 8–10
over Lippert, Budenske, and Kambatla*

1. Overview of Kambatla (Exhibit 1006)

Kambatla relates to distributed computing clusters, more specifically to the allocation and management of computing resources in distributed computing clusters. Ex. 1006 ¶ 2. It discloses techniques for utilization aware cluster scheduling of computing resources in a distributed computing cluster. *Id.* ¶ 17. The techniques are generally referred to as “UBIS” (utilization-based incremental scheduling). *Id.* “UBIS can opportunistically allocate computing resources not utilized by prior allocations.” *Id.*

Figure 1 of Kambatla is reproduced below:

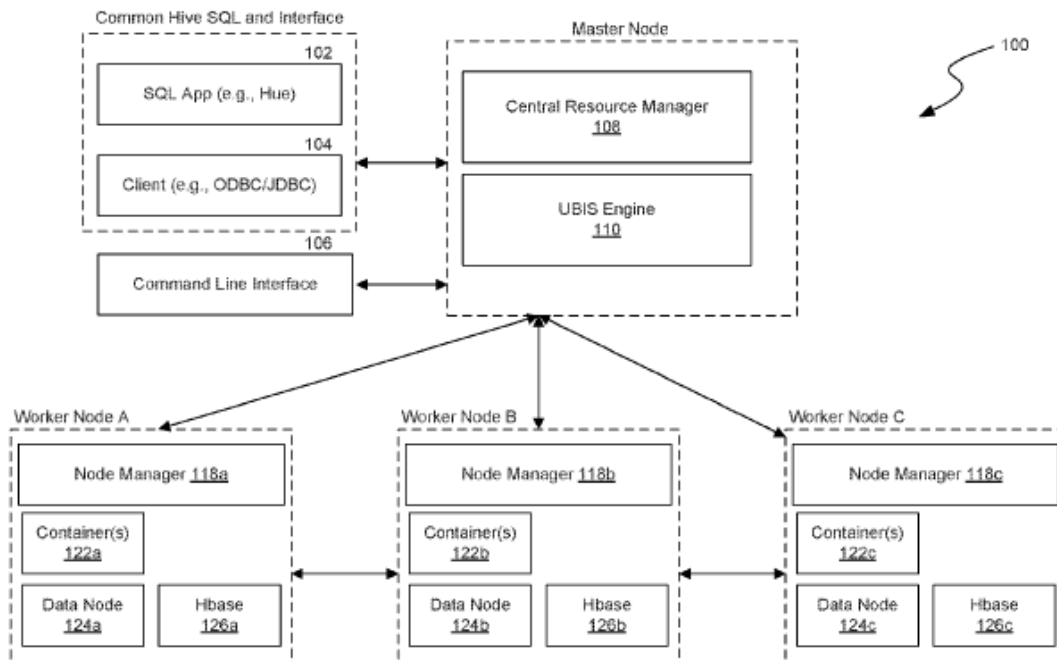


FIG. 1

Figure 1 illustrates an example environment in which utilization-aware resource scheduling can be implemented. Ex. 1006 ¶ 7.

Environment 100 includes a plurality of data nodes 124a-c that comprise cluster of worker nodes in an communication with each other and one or more master nodes. Ex. 1006 ¶ 21. Kambatla describes:

[A] resource manager at a master mode considers actual usage of running tasks and schedules opportunistic work on underutilized worker nodes. The resource manager monitors resource usage on these nodes and preempts 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.

Id. at code (57).

Within the master node is installed a resource manager daemon including a client service module, an administrator service module, an application manager, a scheduler, and a resource tracker. Ex. 1006 ¶¶ 30–31. With regard to the scheduler and the resource tracker, Kambatla describes:

The scheduler 340a is responsible for allocating resources to the various applications subject of constraints queues and policies set by an administrator user. Scheduling is performed based on scheduling algorithms that take into consideration the resource requirements (e.g. memory, processing, data storage, network bandwidth, etc.) of the submitted applications, the administrator policy requirements, and other constraints.

The resource tracker 350a responds to remote procedure calls from the worker nodes. It monitors available resources at the nodes, by receiving status updates from the worker nodes. The resource tracker 350a may also decommission resources at nodes if it does not receive status updates indicating that the node is operational. The resource tracker 350a maintains a list of active and decommissioned nodes and feeds this information to the scheduler 340a to aid in resource allocation.

Id. ¶¶ 35–36.

2. *Claims 2–5 and 8–10*

Petitioner already has made a sufficient showing for these claims, for institution purposes, over Lippert and Budenske. Kambatla is relied on by Petitioner to add to the strength of the prior art teachings. For the reasons presented by Petitioner (Pet. 56–72), it does. Patent Owner has not presented arguments separate from or additional to those it has presented for claims 1 and 9 as unpatentable over Lippert and Budenske, except to extend its argument on motivation to combine and reasonable expectation of success with respect to applying Budenske to Lippert to the context of Kambatla. We have already addressed and rejected Patent Owner’s arguments regarding how Budenske may or may not be applied, when discussing alleged unpatentability of claim 1 over Lippert and Budenske. For similar reasons, Patent Owner’s arguments about motivation to combine and reasonable expectation of success are just as unavailing here.

IV. CONCLUSION

For the reasons discussed above, we determine that Petitioner has demonstrated a reasonable likelihood that it would prevail in showing that at least one of challenged claims 1–10 of the ’442 patent is unpatentable. Our analysis is based on the preliminary record developed thus far and may change after the record is developed fully, during trial.

V. ORDER

Accordingly, it is:

ORDERED that, pursuant to 35 U.S.C. § 314(a), *inter partes* review is instituted as to claims 1–10 of the '442 patent on each of the grounds set forth in the Petition; and

FURTHER ORDERED that, pursuant to 35 U.S.C. § 314(c) and 37 C.F.R. § 42.4(b), *inter partes* review of the '442 patent shall commence on the entry date of this Decision, and notice is hereby given of the institution of a trial.

IPR2025-00318
Patent 11,537,442 B2

For PETITIONER:

Andrew M. Mason
Cameron D. Clawson
Sara Jelsema
Frank Morton-Park
KLARQUIST SPARKMAN, LLP
andrew.mason@klarquist.com
cameron.clawson@klarquist.com
sarah.jelsema@klarquist.com
frank.morton-park@klarquist.com

For PATENT OWNER:

Michael F. Heim
Christopher L. Limbacher
HEIM, PAYNE & CHORUSH, LLP
mheim@hpcllp.com
climbacher@hpcllp.com