

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICROSOFT CORPORATION,
Petitioner,

v.

PARTEC AG,
Patent Owner.

Case IPR2025-00318
Patent 11,537,442 B2

PATENT OWNER'S SUR-REPLY TO PETITIONER'S REPLY

Mail Stop "PATENT BOARD"
Patent Trial and Appeal Board
U.S. Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

EXHIBIT LIST

Exhibit	Description
EX2001	Complaint for Patent Infringement, <i>PARTEC AG and BF EXAQC AG vs. MICROSOFT CORPORATION</i> , Civil Action No. 2:24-cv-433 (E.D. Tex. filed June 10, 2024)
EX2002	Second Amended Docket Control Order, Dkt. No. 48, <i>PARTEC AG and BF EXAQC AG vs. MICROSOFT CORPORATION</i> , Civil Action No. 2:24-cv-433 (E.D. Tex. filed April 9, 2025)
EX2003	United States District Courts – National Judicial Caseload Profile through December 31, 2024, https://www.uscourts.gov/sites/default/files/2025-02/fcms_na_distprofile1231.2024.pdf (accessed April 11, 2025)
EX2004	Defendant Microsoft’s October 30, 2024 Invalidity Contentions, <i>PARTEC AG and BF EXAQC AG vs. MICROSOFT CORPORATION</i> , Civil Action No. 2:24-cv-433 (E.D. Tex.)
EX2005	Defendant Microsoft’s March 6, 2025 Invalidity Contentions, <i>PARTEC AG and BF EXAQC AG vs. MICROSOFT CORPORATION</i> , Civil Action No. 2:24-cv-433 (E.D. Tex.)
EX2006	U.S. Patent Publication No. 2017/0262319
EX2007	Dynamic Process Management with Allocation-internal Co-Scheduling towards Interactive Supercomputing, Clauss et al., 1st COSH Workshop on Co-Scheduling of HPC Applications, 2016
EX2008	PDF comparison generated using PDF-XChange Editor of Petition Sections VII.C.1.c and VII.C.9.e against EX1003 Sections X.C.1.c and X.C.9.e
EX2009	Declaration of Dr. David Kaeli in Support of Patent Owner’s Preliminary Response
EX2010	<i>Curriculum Vitae</i> for Dr. David Kaeli
EX2011	List of Cases for Dr. David Kaeli
EX2012	The DEEP-ER project: I/O and resiliency extensions for the Cluster-Booster architecture, Anke Kreuzer et al., 2018
EX2013	NVIDIA GeForce RTX 2080 Ti Specs TechPowerUp GPU Database, https://www.techpowerup.com/gpu-specs/geforce-rtx-2080-ti.c3305 , accessed April 11, 2025

EX2014	Efficient Data-parallel Distributed DNN Training for Big Dataset under Heterogeneous GPU Cluster, Shinyoung Ahn et al., 2024 IEEE International Conference on Big Data
EX2015	Practice and Experience in using Parallel and Scalable Machine Learning with Heterogenous Modular Supercomputing Architectures, Morris Riedel et al., 2021
EX2016	Definition of “portion,” The American Heritage Dictionary of the English Language, Third Edition, Houghton Mifflin Company, 1992
EX2017	Definitions of Research and Development: An Annotated Compilation of Official Sources, https://nces.nsf.gov/pubs/nces25215#definitions , accessed on September 20, 2025
EX2018	ParTec partners with HDRZ research lab to develop 500 petaflops AI supercomputer, https://www.datacenterdynamics.com/en/news/partec-partners-with-hdrz-research-lab-to-develop-500-petaflops-ai-supercomputer/ , accessed on September 22, 2025
EX2019	ParTec AG and HZDR to Build AI Supercomputer Supporting Research in AI, Quantum Computing, and HPC, https://thequantuminsider.com/2024/10/22/partec-ag-and-hzdr-to-build-ai-supercomputer-supporting-research-in-ai-quantum-computing-and-hpc/ , accessed on September 23, 2025
EX2020	Deposition Testimony of Darrell Long, Ph.D., September 17, 2025
EX2021	Declaration of Dr. David Kaeli in Support of Patent Owner’s Response
EX2022	Exhibits to the September 17, 2025 Deposition of Darrell Long, Ph.D.

TABLE OF CONTENTS

I.	Ground 1	1
A.	Claim 1	1
1.	Further Distribution of Sub-tasks	1
2.	Budenske Is Not Enabling	8
3.	Combining Lippert And Budenske	13
4.	Generating A Further Distribution Using the Claimed Information.....	16
5.	The Claimed Information.....	16
B.	Claim 3: Dynamically Altering Further Distribution.....	18
C.	Claim 5: Computation Node Daemons and Booster Node Daemons	20
D.	Claim 9: Fails For The Same Reasons As Claims 1 and 5	21
II.	Ground 2	21
A.	Claim 3: Ground 2 Fails For The Same Reasons As Ground 1	21
B.	Claim 5: Ground 2 Fails To Correct The Deficiencies In Ground 1	21
C.	Claim 9: Ground 2 Fails To Correct The Deficiencies In Ground 1	23
D.	Combining Kambatla with Lippert-Budenske	23
III.	Petitioner’s Claim Construction Positions.....	23
IV.	Conclusion	24

TABLE OF AUTHORITIES

Cases

<i>ActiveVideo Networks, Inc. v. Verizon Communs., Inc.</i> , 694 F.3d 1312 (Fed. Cir. 2012)	21, 23
<i>Amgen, Inc. v. Chugai Pharm. Co.</i> , 927 F.2d 1200, 1213 (Fed. Cir. 1991)	12
<i>Koninklijke Philips N.V. v. Google LLC</i> , 948 F.3d 1330 (Fed. Cir. 2020)	20
<i>Phillips v. AWH Corp.</i> , 415 F.3d 1303 (Fed. Cir. 2005)	18
<i>Revvo Techs. Inc. v. Cerebrum Sensor Techs, Inc.</i> , IPR2025-00632, Paper 20 (Dir. Squires Nov. 3, 2025).....	24

Rules

37 CFR § 42.104(b)(4).....	8
----------------------------	---

Petitioner’s Reply (Paper 19, the “Reply”) fails to overcome the deficiencies identified in Patent Owner’s Response (Paper 16, the “Response”). In the absence of any express disclosures in its grounds demonstrating the claimed “generating...a further distribution of the plurality of sub-tasks...in a further computing iteration,” Petitioner now reasons (incorrectly) that because the references do not “proscribe” such functionality, then that is sufficient for them to be invalidating. However, that is not the law, and all grounds necessarily fail for at least that reason. The Reply arguments addressing the claimed (1) “information relating to the processing of the plurality of sub-tasks,” and (2) computation node daemons and booster node daemons are similarly deficient, and are unsupported by the evidence.

The Reply also presents a belated *prima facie* obviousness theory for Claim 3, attempting to overcome the flaws in the Petition grounds. Even if such untimely theories were considered, they still fail to identify the claimed element in any reference, instead relying on a new expert declaration to address the missing limitation.

I. Ground 1

A. Claim 1

1. Further Distribution of Sub-tasks

Petitioner’s Reply arguments regarding the “generating...a further distribution” limitation fail to address the specific claim language and the

specification description of the '442 Patent invention. Reply at 1-5. Claim 1 does not, merely require 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.” Reply at 1. Rather, there must be a first distribution of subtasks between computation nodes and booster nodes in a first computing iteration, and then the system must “generat[e]...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 at 5:25-43; Response 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).

The Reply fails to demonstrate the first requirement (processing the plurality of sub-tasks by at least one computation node and at least one booster node during the further computing iteration) in Budenske. As an initial matter, Budenske makes clear (and Petitioner does not contest) that Budenske cannot distribute individual subtasks to multiple processors of different types during a computing iteration, much less distribute individual subtasks to both computation nodes and booster nodes.

EX1005 at 392 (“[I]f an implementation of a given subtask uses multiple processors, all processors will be of the same type.”); Response at 27-29; EX1029 at 100:6-11. Even assuming it is sufficient for the plurality of subtasks to be processed by a combination of computation nodes and booster nodes, Petitioner still fails to identify where that occurs in Budenske. Petitioner’s Reply arguments focus on “mapping of sub-tasks to different processor types” generally, rather than on redistributing the plurality of subtasks to both computation nodes and booster nodes in a further computing iteration. *See* Reply at 2-5.

The Reply also fails to demonstrate the second requirement (a further distribution in which at least one subtask is redistributed to a different node for processing). The ’442 Patent system enables adaptive, performance-driven redistribution of subtasks between different nodes, and the requirement for a further distribution of subtasks is confirmed throughout the ’442 Patent specification. *See, e.g.,* EX1001 at 2:15-18 (“adapt[ing] or improv[ing] upon [the distribution] for a further computing iteration), 2:45-50, 3:59-4:13 (noting that the first division between nodes may not be optimal and suggesting ways to “optimize the computation of the task as a whole”), 4:23-37 (discussing “reassignment of the sub-tasks for a subsequent iteration”), 5:1-24; *see also* EX1029 at 79:22-80:3, 85:6-11. Petitioner’s *Markman* briefing from the District Court confirms the same understanding. *See, e.g.,* EX1031 at 25-26 (“The specification itself consistently

discusses changing the *division* of sub-tasks among booster and computation nodes...”); *id.* (suggesting that the ’442 Patent “feedback mechanism” involves “redistributing the sub-tasks based on that performance data in the next iteration”).

Despite this, Petitioner identifies no such change of distribution in Budenske. Even assuming it is sufficient for a given subtask to be moved from being processed solely by one node to solely by another node in a further iteration (as opposed to being processed by a combination of nodes), the asserted grounds fail to teach or render obvious that capability. *See* Response at 10-13, 18-19, 23-29. As would have been understood by a skilled artisan, Budenske does not teach any situations where a subtask is redistributed from one node to another, much less how to do so. *See* Response at 13-16, 23-29. Petitioner suggests that Budenske “teaches mapping subtasks to different processor types in an iteration,” but never identifies any scenario in which those subtasks are redistributed to a different node. Reply at 3.

In Budenske, a particular sub-task is assigned off-line to a particular processor to which it is best suited, and then assignment of that particular sub-task remains with that same processor for subsequent processing iterations. Budenske’s allocation of processors to specific subtasks or ability to modify the order in which subtasks are executed by a specific processor (“i.e., a matching of application subtasks to processors and a scheduling for the execution order of these subtasks”) does not

satisfy the actual claimed redistribution of subtasks between nodes. *See* EX1005 at 387-88.

Petitioner relies on Budenske's generic statements regarding "architectural flexibility" and "reduc[ing] execution time," suggesting those somehow rectify the lack of disclosure of redistribution of subtasks between nodes. Reply at 2-3. But as Petitioner concedes, "Budenske expressly teaches to 'assign each ... subtask to the processors where it is best suited for execution.'" Reply at 3 (quoting EX1005 at 390). Petitioner does not identify any scenario in Budenske in which the processor "best suited" to handle a given subtask would be modified during execution. This makes sense given the specialized task processing application Budenske was customized for (automatic target recognition), along with its specific need for "**dedicated** heterogeneous parallel hardware." EX1005 at 387 (emphasis added); *see also* EX2020 at 133:4-10 ("[Automatic target recognition] means to capture an image and look for targets."); EX1029 at 100:20-101:25, 103:11-17 ("Everything that's done in terms of mapping and redistribution is only for this one application. And so this is a very customized implementation..."), 109:7-20, 119:21-121:18 (contrasting technology at the time of Budenske); Response at 45. Petitioner's theory relies on stretching statements regarding generic benefits to cover the specific claimed functionality regarding the handling of subtasks that the grounds do not address, much less teach or render obvious.

Petitioner also asserts that Budenske must know how many processors to assign and might need to “reconfigure the assignment of processing resources.” Reply at 3. However, Petitioner still fails to cite anything in Budenske which teaches redistribution of subtasks from one node to another, even if processing resources are “reconfigured.” Instead Budenske, refers to “matching of application subtasks to processors and a scheduling for the execution order of these subtasks” based on the input data received. EX1005 at 387. Budenske would need information such as the number of processors available so that it would be able to determine an assignment of processors, and to determine whether “subtasks c[ould] execute concurrently” or whether they would need to be executed sequentially. EX1005 at 387-88, 390; EX2021 ¶¶52-53, 63-68, 80; EX1007 at 78. Petitioner’s theory that tracking such information somehow implies redistribution of specific subtasks between different nodes is pure hindsight and should be rejected.

Petitioner then attempts to mischaracterize Dr. Kaeli’s testimony as supporting its theory. Reply at 3-4 (quoting parts of EX1029 at 102:8-19). The question Dr. Kaeli was asked whether it was possible for different subtasks to be

distributed to different processor types¹ during the **same** computing iteration (*i.e.*, not a further distribution or redistribution). Dr. Kaeli responded that it was possible that “a portion of the subtasks can be mapped to one processor type and a portion of the subtasks to another processor type,” and that how the processors were allocated might depend on the input. *See, e.g.*, EX2021 ¶¶52-53, 63-68, 80. However, such an allocation between processor types in a single computing iteration has no bearing on whether Budenske addresses **redistributing** subtasks between computation **nodes** and booster **nodes** in a further iteration.

Petitioner then points to Budenske’s discussion of “implementations” of a subtask, theorizing that the implementation of a subtask might change between iterations. Reply at 4-5. Petitioner cites nothing from Budenske that teaches changing the processing of a subtask from one node to another even if the implementation changes. Petitioner tacitly concedes that its theory boils down to the idea that because Budenske does not “proscribe” the claimed redistribution of subtasks, then it must be viewed as teaching it even in the absence of any express disclosure (and the presence of multiple counterindications from the disclosure

¹ The fact that the question related to different “processor types” rather than computation nodes and booster nodes limits any applicability to the Challenged Claims.

itself). Reply at 5. However, such an assertion is facially unreasonable and fails to “specify where each element of the claim is found in the prior art patents or printed publications relied upon,” and thus must necessarily fail. *See* 37 CFR § 42.104(b)(4).

2. Budenske Is Not Enabling

The Reply fails to overcome the fact that Budenske does not enable the claimed the elements upon which Petitioner relies. Reply at 5-12; *See* Response at 30-40.

Petitioner first wrongly asserts that Kwok fully implemented Budenske’s approach, and that Table 4 “confirms” that Budenske was enabling. Reply at 6-7. However, Table 4 only shows data for one iteration (iteration 5), and thus provides no insight into whether the assigned processor changed between iterations or which processor might be associated with different node types (and other parts of the disclosure make it clear that it suffered from the same limitations as Budenske). EX1007 at 90.

Furthermore, because Kwok was still performing “fundamental research” rather than providing a functional implementation, Response at 36-40, 42, it did not perform the experiment with actual processors, but instead used theoretical placeholders referred to as “processor type 0” through “processor type 3.” EX1007 at 82. From this, it would not be possible to determine which of those “processor types” would qualify as computation nodes or booster nodes. Other factors further

reinforce the fact that Kwok was not a functional “implementation” of Budenske. For example, Kwok explains that it is “simulating the methodology” being discussed, and therefore a POSITA would have understood it to be further “fundamental research” and not a functional implementation. EX1007 at 80, 82 (noting assumptions made to simplify this “simulation study” and that “a simplified model is used for subtask execution time and inter-subtask communication time”), 83 (“[W]hen considering the results of the paper the reader should note that the execution time and communication time equations are parts of a simplified model.”), 84 (specifying that the dynamic parameter values are “simulated”), 87 (discussing random generation of the task graphs, including “randomly selected” heterogeneity factors).

Petitioner suggests that Kwok “solves” its fundamental research issue while ignoring the testimony of both Dr. Kaeli and Dr. Long on the nature of fundamental research issues. Reply at 7-8; *see* Response at 36-37, 39, 42. Even if this later Kwok paper had solved a particular fundamental research issue (something that Kwok does not say), that would not mean that a POSITA would be able to successfully make and use the claimed invention. Fundamental research does not provide a readily applicable solution to be implemented, but rather it provides basic building blocks necessitating further research and experimentation. EX2021 ¶¶75-81.

Petitioner also suggests that Kwok's use of only a single heterogeneity factor for its equation predicting subtask execution times is overcome by storing different factors for different processors. Reply at 8. However, that observation ignores that the equation itself only recites a single heterogeneity factor to be used in the calculation, and to function as Petitioner would have the Board believe, the equation would have to include a separate heterogeneity factor " h_{iu} " for every different type of processor used, along with a separate number of processors used " p " for each processor type. *See* Response at 38; EX1007 at 83. Kwok merely teaches determining the heterogeneity factor for the single processor type that has been selected and using that to calculate the subtask execution time.

Petitioner further characterizes Budenske's limiting factors (such as its teachings of certain processors sharing the same DRAM) as simply examples. Reply at 9-10. However, Petitioner fails to identify any "examples" that would show the system functioning as Petitioner asserts. Petitioner again falls back on its overarching argument that the "Budenske disclosures [do not] preclude sub-tasks from being assigned to different processor types," ignoring the fact that a reference not precluding something does not somehow enable a POSITA to make and use concepts not disclosed. Reply at 10.

Next, Petitioner argues that even though Budenske is not enabling, its Petition does not fail because "PO identifies nothing missing from Lippert that would be

required for implementation of Budenske’s teachings.” Reply at 10-12. However, the record of what Lippert does and does not teach regarding the “further distribution” limitation was previously established in front of the USPTO, and nothing of record overcomes those prior findings. *See, e.g.*, Response at 10-13, 18-19. The ’442 Patent explains that it was a further development of Lippert and addresses how it improved upon Lippert’s teachings. EX1001 at 1:23-27; EX1004 at [0001]. Petitioner’s focus on “commercial, off-the-shelf components” and “novel hardware” is wholly misplaced, because it fails to address that it was (*inter alia*) the functioning of the heterogeneous computing system **in a further computing iteration** that the ’442 Patent improved. Reply at 10-11.

Petitioner’s focus on the ’442 Patent disclosure generally and the Monte-Carlo example disclosed in the ’442 Patent specifically is similarly misplaced. Reply at 11-12. The ’442 Patent’s disclosure of a novel way to perform an existing simulation does not somehow make the invention easier to enable such that Petitioner would not have to show the claimed elements in the art. Petitioner cannot overcome the fact that the evidence of record overwhelmingly demonstrates that Budenske would not have enabled the “further distribution...in a further computing iteration” limitations that the ’442 Patent recites, teaches, and enables. Reply at 11-12; *see, e.g.*, EX1001 at 1:52-55 (addressing “when the load offered to the service increases”), 1:66-2:57 (giving exemplary arrangements “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”), 3:50-58 (discussing suitability of subtasks for computation and booster nodes), 3:59-4:3 (discussing handling divisions of subtasks that are not “optimal”), 4:4-15 (discussing architectural arrangements), 4:16-31 (discussing “reassignment of the sub-tasks for a subsequent iteration”), 4:32-37 (discussing exemplary daemon arrangements), 4:48-5:24 (discussing additional architectural, distribution, and assignment embodiments).

Finally, Petitioner suggests that “Patent Owner Fails To Meet Its Burden On Enablement,” but that assertion is wrong for multiple reasons. Reply at 12. Initially, Budenske’s very own disclosure makes clear that it is not enabled, that it was merely teaching “design concepts,” and determining how to use those design concepts would be a “major undertaking.” *See* Response at 30-40 (quoting EX1005 at 388-89). Even ignoring the additional evidence of non-enablement discussed in the Response, those statements alone are enough to demonstrate that Budenske is not a self-enabling reference.

Petitioner faults PO for not analyzing every *Wands* factor, ignoring that the factors are “are illustrative, not mandatory” and “[w]hat is relevant depends on the facts.” *Amgen, Inc. v. Chugai Pharm. Co.*, 927 F.2d 1200, 1213 (Fed. Cir. 1991). Furthermore, Petitioner quite conspicuously declined to engage in any analysis of additional *Wands* factors itself. Even if Petitioner had been able to muster some form

of evidence, the subsequent publications that still were performing “fundamental research” on the same topics as Budenske confirm that the amount of experimentation to make use of its “teachings” would have been unreasonable by any measure. *See* Response at 35-39.

3. Combining Lippert And Budenske

The Reply fails to overcome the reasons a POSITA would not have combined Lippert and Budenske discussed in PO’s Response. Reply at 13-15; Response at 40-46. Petitioner suggests that “Budenske teaches to use its remapping techniques in other ‘iterative’ application domains,” but conspicuously omitted what Budenske actually says on the topic, which is that its “ideas can also be used for other application domains and classes of hardware platforms **whose characteristics are similar to those of the iterative ATR applications.**” EX1005 at 390 (emphasis added); Reply at 13. Budenske never professed to teach the kind of generally applicable solution claimed by the ’442 Patent, as evidenced throughout its disclosure and the expert testimony in this case. *See, e.g.*, EX1005 at 387; EX2020 at 133:4-10; EX1029 at 100:20-101:25, 103:11-17, 109:7-20, 119:21-121:18; Response at 45. Budenske even acknowledged that trying to adapt its “HC Kernel based systems for a variety of different applications and a variety of different target hardware platforms” would require “[f]uture research.” EX1005 at 404; *see also id.* at 390 (“[T]he 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.”).

Petitioner suggests that “both Budenske and Kwok solved and evaluated the ‘fundamental research issue.’” Reply at 13. That conclusion is baffling given the nature of fundamental research—if Budenske had solved it, why would Kwok still consider it to be fundamental 7+ years after Budenske? *See* Response at 36-37, 39, 42. Petitioner suggests Budenske was a more “simplistic” approach to the problem, but the paragraph Petitioner cites states that Budenske’s method of using dynamic parameters was not “ever evaluated in any way, as this was not the focus of Budenske.” Reply at 13-14; EX1007 at 79. Kwok was researching different approaches to “solve” the fundamental research issue of how to select representative mappings for assigning subtasks to a single processor type (not redistributing subtasks between nodes across computing iterations). EX1007 at 80. Furthermore, neither reference purports to have fully “solved” those fundamental issues, but rather to “introduce [] ideas” and offer “future system designers” potential starting points to “evaluate as an alternative for their particular situations.” EX1005 at 390. As discussed previously in section I.A.2, Kwok was not a functional implementation,

but rather an attempt at modeling/simulating various aspects in furtherance of its “fundamental research” goals.

Petitioner also suggests that Budenske overcomes the later Kambatla’s statements regarding “accurately estimating computing resource requirements” because Budenske purportedly “teaches to consider multiple information sources to determine a given subtask’s resource requirements and a given heterogeneous computing system’s resource capabilities.” Reply at 14. Yet, Budenske actually confirms the same lessons as Kambatla regarding estimating resource requirements, noting that “the execution time of application subtasks in an ATR system is highly input-data dependent.” EX1005 at 390.

Finally, Petitioner suggests that the praise for the ’442 Patent from its own expert should be ignored because “[f]ault tolerance is not claimed or even mentioned by the ’442 patent.” Reply at 15. However, Petitioner concedes in a footnote that Lippert (which the ’442 Patent is a further “development” of, EX1001 at 1:23-27; EX1004 at [0001]) does, in fact, address fault tolerance. Reply at 15 n.1. Thus, Petitioner’s halfhearted attempt to distance itself from the knowledge and testimony of its own expert and the express statements of the references it relies on should be rejected.

4. Generating A Further Distribution Using the Claimed Information

Petitioner declined to address PO's Response argument regarding generating a further distribution using the claimed information, instead formulating a strawman argument that when Budenske "instantiates" a mapping that is the same thing as generating it and suggesting that PO did not dispute that generating is the same thing as instantiating. Reply at 15-16. What PO actually explained in the Response was that because Budenske generated all purported mappings offline before any processing of sub-tasks even begins, that it clearly was not generating those mappings "using information relating to the processing of the plurality of sub-tasks." Response at 46-47. The off-line heuristics of Budenske that are not generated during processing of the sub-tasks are clearly not generated using information relating to the processing of the sub-tasks, at least because no processing of the sub-tasks has yet occurred. Petitioner's attempt to redefine "generating" as "instantiating" cannot overcome that deficiency (especially given that Budenske's "off-line statically determined" mapping was its defining element). *See, e.g.*, EX1005 at 387-88.

5. The Claimed Information

Petitioner asserts that the claimed information cannot be the information concerning the processing of the sub-tasks described in the specification because Claim 5 must be narrower than Claim 1 and Claim 5 recites that the daemons generate the information. Reply at 16. However, Claim 5 is narrower than Claim 1

because it specifies where the daemons operate to generate the information; it does not recite anything about the content of the information.

Petitioner also asserts that its theories do not fail despite the dynamic parameters of Budenske being expressly related to the resulting values of input data processing,² because Petitioner suggests that they would “be analogous to and used in the same way as the scalability of a subtask.” Reply at 17-18. Petitioner relies on a single dynamic parameter related to “whether the lighting conditions in the last image frame required requantization for better contrast for display.” Reply at 17. However, the results of evaluating lighting conditions in a data frame being processed is still indisputably the results of processing the data for that image frame, a fact effectively confirmed by Petitioner’s expert declining to render an opinion on that parameter in his Reply declaration. *See* EX1030 ¶¶45-47. That is not surprising, given the fact that Petitioner’s expert had previously testified that same parameter referred to “the data being processed,” not to the scalability of the subtask as Petitioner now argues. EX2020 at 130:1-22.

² *See, e.g.,* Response at 47-50; EX1005 at 390 (noting the “highly input-data dependent” nature of the mapping), 396 (noting that execution time is dependent on input data).

Furthermore, the Reply fails to demonstrate how the “lighting conditions” parameter would be analogous to scalability information. For example, when discussing scalability the ’442 Patent gives the example that “when the load offered to the service increases, an incremental and linear increase in hardware can maintain the same per-user level of service.” EX1001 at 1:53-55. Petitioner made no attempt to explain how the “lighting conditions” parameter would affect, *e.g.*, the load offered to the service or the level of service. Instead, Petitioner offered no more than conclusory statements that the parameter would be “analogous” to scalability information, avoiding entirely any sort of substantive analysis of what it would mean for the parameter to be scalability information.

B. Claim 3: Dynamically Altering Further Distribution

Petitioner asserts that the “further distribution” recited in Claim 3 does not refer back to the “further distribution” recited in Claim 1 because it “lacks an article.” Reply at 18. Even assuming for the sake of argument that the “further distribution” of Claim 1 is not the antecedent basis of the same term used in Claim 3, Petitioner provides no explanation as to why it would have a different meaning (*i.e.*, an **assignment** of nodes to each other—as recited in, *e.g.*, Claims 4, 9, and 10—rather than a **distribution** of tasks). *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1314 (Fed. Cir. 2005) (“[T]he usage of a term in one claim can often illuminate the meaning of the same term in other claims.”).

Petitioner suggests that the '442 Patent “never contemplated” that the resource manager would redistribute sub-tasks among nodes. Reply at 18-19. However, examples of the resource manager doing so appear throughout the specification. *See, e.g.*, EX1001 at 4:13-15 (“The **resource manager** can also reassign computation nodes and boosters to each other, **in addition to the adjusted distribution of tasks.**” (emphasis added)), 4:32-37 (“For each iteration, the daemons...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.” (emphasis added)), 5:4-9. Claim 3 as originally filed also contained the same “resource manager dynamically alters...” language. EX1002 at 27.

Petitioner asserts that the Petition did not “violate” the doctrine of claim differentiation, yet does not identify any way that Claim 4 would be narrower than Claim 3 under its untenable interpretation. Reply at 19-20.

Finally, Petitioner attempts to make out an untimely new *prima facie* theory for Claim 3, suggesting that it would have been obvious to have the claimed resource manager redistribute sub-tasks. Reply at 20-21. However, Petitioner does not identify any reference in which such an arrangement exists, and instead relies solely on its expert to fill in the missing limitation, without citing anything in the prior art. Yet, Petitioner’s expert does no more than parrot from the Reply regarding general

statements of “architectural flexibility” and “fault tolerance.” Nothing in the Reply or the declaration would suggest to a POSITA to specifically implement the heterogeneous computing system as claimed, much less to have the resource manager perform a further distribution of the sub-tasks. Nor has Petitioner offered any evidence that the specific claimed arrangement is “unusually simple” and the technology “particularly straightforward.” *See Koninklijke Philips N.V. v. Google LLC*, 948 F.3d 1330, 1338 (Fed. Cir. 2020). Petitioner is instead completely reliant on improper hindsight for the proposed modifications. Therefore, even if Petitioner’s theory was not untimely, it would fail on the merits as well.

C. Claim 5: Computation Node Daemons and Booster Node Daemons

Petitioner asserts that the claimed computation node daemons and booster node daemons would have been obvious because they would ““beneficially facilitate inter-element communication while providing communication authentication, and to enable independent control of task execution and reporting of task processing at each node.”” Reply at 21-22 (quoting Petition at 42-43). Yet, none of those things are properties that are inherent to daemons. And Petitioner’s own expert testified that there are other ways to implement the same functionality, *see* Response at 56, and Petitioner cites nothing specific to daemons that would have prompted a POSITA to specifically modify Lippert and Budenske to use them over other potential options. *See ActiveVideo Networks, Inc. v. Verizon Communs., Inc.*, 694 F.3d 1312, 1327-28

(Fed. Cir. 2012) (rejecting “testimony [that] is generic and bears no relation to any specific combination of prior art elements” because it “fails to explain why a person of ordinary skill in the art would have combined elements from specific references *in the way the claimed invention does*”).

D. Claim 9: Fails For The Same Reasons As Claims 1 and 5

The Reply fails to remedy the Petition shortcomings for Claim 9 for at least the same reasons discussed for Claims 1 and 5 above.

II. Ground 2

A. Claim 3: Ground 2 Fails For The Same Reasons As Ground 1

The Reply appears to concede that Petitioner is relying on the same Lippert-Budenske functionality as Ground 1 (rather than Kambatla) to teach or render obvious the “dynamically alter[ing] further distribution during the computing of the computation task” limitation of Claim 3. Reply at 23. Thus, Ground 2 fails for Claim 3 for the same reasons discussed above for Ground 1.

B. Claim 5: Ground 2 Fails To Correct The Deficiencies In Ground 1

Petitioner’s new arguments as to why Kambatla would be relevant to heterogeneous computing are again based on hindsight combined with generic statements rather than any express disclosure. Petitioner asserts Kambatla teaches a relevant heterogeneous computing system because it “discloses that its worker nodes may have different hardware” and include different “resource types.” Reply at 23-

24 (internal quotes, citations, and emphasis omitted). However, specific nodes having “different hardware” or including different “resource types” falls short of making something a heterogeneous computing system. The ’442 Patent specifically refers to “a heterogeneous computing system” as “comprising a plurality of computation nodes and a plurality of booster nodes.” *See, e.g.*, EX1001 at Abstract. The heterogeneity stems from this use of both computation nodes and booster nodes, not from unspecified differences in hardware. EX1001 at 3:24-25 (“**The computation nodes** 20 may be identical to one another or **may have differing characteristics**.” (emphasis added)); *see also* EX2020 at 16:10-16. To the extent Petitioner is asserting a new theory based on inherency, Petitioner has failed to demonstrate that the nodes of Kambatla would necessarily include different types of processors (much less nodes).

In the absence of any supporting disclosure from Kambatla, Petitioner asserts that it is “irrelevant” whether Kambatla relates to heterogeneous computing. Reply at 24-25. Petitioner again relies on generic statements regarding “inter-element communications” without any express disclosure specifically relating to use of computation node daemons and booster node daemons. *Id.* For the same reasons discussed in PO’s Response, Petitioner’s theory fails. Response at 54-56.

Petitioner repeats this approach to support its motivation to combine arguments, citing generic statements untethered to the functionality it wishes to

combine from the references. Reply at 25-26 (discussing “optimally scheduling tasks,” “facilitat[ing] inter-element communication,” and “resource fragmentation”). Petitioner does not explain how any of those generic “benefits” would have motivated a POSITA to make its specific proposed modifications. *See ActiveVideo*, 694 F.3d at 1327-28.

C. Claim 9: Ground 2 Fails To Correct The Deficiencies In Ground 1

For the same reasons discussed above for Claims 1 and 5 (of both Grounds), Ground 2 fails for Claim 9. *See* Reply at 26; Response at 23-50, 54-63.

D. Combining Kambatla with Lippert-Budenske

For the same reasons discussed in the Response, a POSITA would not have combined Lippert or Budenske with Kambatla. *See* Response at 64; Reply at 26-27.

III. Petitioner’s Claim Construction Positions

The Petition also fails because Petitioner failed to show how its grounds could meet the narrowing constructions it advanced in the District Court. In the District Court, Petitioner has asserted that to satisfy Claims 1 and 9, **each** sub-task must be repeated in **each** iteration.³ *See, e.g.*, EX1031 at 25 (asking the Court to reject

³ Petitioner’s expert in this IPR testified that Petitioner’s District Court construction was not required by the claim language. EX2020 at 28:24-29:5. Petitioner’s

“ParTec’s argument that it may not be the same sub-tasks repeated in each iteration” (internal quotes and citations omitted)), 26 (“[I]t is the same ‘plurality of sub-tasks’ that is executed in the ‘first computing iteration’ and then repeated in the ‘further computing iteration’...The specification itself consistently discusses changing the division of sub-tasks among booster and computation nodes...not adding or removing sub-tasks.” (emphasis omitted)), 27-28.

Petitioner has failed to show that its Petition grounds render obvious the claims under its District Court construction (in fact, Petitioner did not even attempt to do so beyond a conclusory sentence in its Reply). Reply at 27. Petitioner identifies nothing in its Petition grounds that would show that its Ground 1 or Ground 2 combinations would repeat the exact same sub-tasks in each computing iteration without, e.g., “adding or removing sub-tasks.” Thus, the Petition grounds should be rejected. *See Revvo Techs. Inc. v. Cerebrum Sensor Techs, Inc.*, IPR2025-00632, Paper 20 at 3-5 (Dir. Squires Nov. 3, 2025) (precedential).

IV. Conclusion

Patent Owner respectfully requests that the Board confirm patentability of all challenged claims.

inconsistent positions in the District Court and this proceeding exemplifies the type of gamesmanship denounced by the Director.

Dated: February 3, 2026

Respectfully submitted,

By: /Michael F. Heim /
Michael F. Heim (Reg. No. 32,702)
Attorney for Patent Owner
ParTec AG

CERTIFICATE OF SERVICE

The undersigned certifies that pursuant to 37 C.F.R. § 42.6(e), a copy of the foregoing was served via email to lead and backup counsel of record for Petitioner as follows:

Lead Counsel for Petitioner
Andrew M. Mason (Reg. No. 64,034) andrew.mason@klarquist.com
Back-Up Counsel for Petitioner
Cameron D. Clawson, Reg. No. 73,509 cameron.clawson@klarquist.com
Sarah Jelsema, Reg. No. 70,804 sarah.jelsema@klarquist.com
Frank Morton-Park, Reg. No. 80,750 frank.morton-park@klarquist.com
KLARQUIST SPARKMAN, LLP 121 SW Salmon Street, Suite 1600 Portland, Oregon, 97204 503-595-5300 (phone) 503-595-5301 (fax) Msft-Partec@klarquist.com

Dated: February 3, 2026

Respectfully submitted,

By: /Michael F. Heim /
Michael F. Heim (Reg. No. 32,702)
Attorney for Patent Owner
ParTec AG

CERTIFICATE OF COMPLIANCE

Pursuant to 37 C.F.R. § 42.24(d), the undersigned hereby certifies that this brief complies with the type-volume limitation of 37 C.F.R. § 42.24 because this brief contains 5,299 words.

Dated: February 3, 2026

Respectfully submitted,

By: /Michael F. Heim/
Michael F. Heim (Reg. No. 32,702)
Attorney for Patent Owner
ParTec AG