

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

MICROSOFT CORPORATION,
Petitioner,

v.

PARTEC AG,
Patent Owner.

IPR2025-00318
Patent No. 11,537,442

PETITIONER'S REPLY TO THE PATENT OWNER RESPONSE

TABLE OF CONTENTS

	Page
EXHIBIT LIST	iv
I. INTRODUCTION	1
II. GROUND 1	1
A. Claim 1: Budenske Teaches Distributing And Redistributing Sub-Tasks To Both Computation Nodes And Booster Nodes.....	1
1. PO’s Reading Would Contradict Budenske’s Main Purpose	2
2. PO’s Expert Agreed That Budenske Distributes Subtasks To Different Processor Types	3
3. PO Continues To Rely On A Single Out-Of-Context Statement In Arguing Its Strained Read of Budenske	4
B. Claim 1: Lippert-Budenske Is Enabling.....	5
1. Budenske’s Pertinent Teachings Were Successfully Implemented Over A Decade Before The ’442 Priority Date	6
2. Budenske Enables The Relied-Upon Disclosures	8
3. Lippert-Budenske “As A Whole” Enables A System That Meets The Challenged Claims.....	10
4. Patent Owner Fails To Meet Its Burden On Enablement	12
C. Claim 1: A POSITA Would Have Combined Lippert And Budenske.....	13
D. Claim 1: Budenske Generates In Real-Time A Further Distribution Of Sub-Tasks Using Information Relating To The Processing Of The Plurality Of Sub-Tasks.....	15
E. Claim 1: Budenske’s Dynamic Parameters Are Information Relating To The Processing Of The Plurality Of Subtasks	16

F. Claim 3: Patent Owner Improperly Construed The Claim
To Require Acts Never Contemplated By The '442 Patent.....18

G. Claim 5 21

1. Lippert-Budenske Renders Obvious
Computation Node And Booster Node Daemons.....21

H. Claim 9 22

III. GROUND 2 23

A. Claim 3 23

B. Claim 5 23

C. Claim 9 26

D. A POSITA Would Have Been Motivated To
Combine Kambatla With Lippert-Budenske.....26

IV. PARALLEL DISTRICT COURT
CLAIM CONSTRUCTION POSITIONS.....27

CERTIFICATE OF COMPLIANCE.....28

CERTIFICATE OF SERVICE 29

TABLE OF AUTHORITIES

Page

Board Decisions

<i>Revvo Techs., Inc. v. Cerebrum Sensor Techs, Inc.,</i> IPR2025-00632, Paper 20 (Director Nov. 3, 2025)	27
<i>Samsung Elecs. Co. v. Sung,</i> IPR2024-00533, Paper 39 (PTAB Sep. 17, 2025)	5

Cases

<i>Agilent Techs., Inc. v. Synthego Corp.,</i> 139 F.4th 1319 (Fed. Cir. 2025)	5
<i>AK Steel Corp. v. Sollac & Ugine,</i> 344 F.3d 1234 (Fed. Cir. 2003)	20
<i>Apple Inc. v. Corephotonics, Ltd.,</i> 861 F. App'x 443 (Fed. Cir. 2021)	5
<i>In re Epstein,</i> 32 F.3d 1559 (Fed. Cir. 1994)	11
<i>In re Morsa,</i> 803 F. 3d 1374 (Fed. Cir. 2015)	12
<i>In re Mouttet,</i> 686 F. 3d 1322 (Fed. Cir. 2012)	22
<i>In re Wands,</i> 858 F.2d 731 (Fed. Cir. 1988)	12
<i>Raytheon Techs. Corp. v. Gen. Elec. Co.,</i> 993 F.3d 1374 (Fed. Cir. 2021)	5, 6

EXHIBIT LIST

LIST OF NEWLY FILED EXHIBITS

Exhibits concurrently filed with Petitioner's Reply To Patent Owner Response:

No.	Description
1028	Digital Technical Journal, Vol. 10, No. 1 (Dec. 1998)
1029	Transcript of Deposition of David R. Kaeli, Ph.D. dated December 9, 2025
1030	Reply Declaration of Darrell Long dated December 22, 2025
1031	Defendant Microsoft Corporation's Responsive Claim Construction Brief, <i>Partec AG, et al., v. Microsoft Corporation</i> , Case No. 2:24-cv-00433-RWS-RSP (E.D. Tex., Nov. 14, 2025)
1032	Intel Xeon Processor E5 v4 Product Family Datasheet (June 2016)
1033	Intel Xeon Phi Coprocessor x200 Product Family Datasheet (Apr. 2017)
1034	Mellanox InfiniBand FAQ (Dec. 2014)

LIST OF PREVIOUSLY FILED EXHIBITS

No.	Description
1001	U.S. Patent No. 11,537,442 (“ 442 patent ”)
1002	File History of U.S. Patent No. 11,537,442
1003	Declaration of Darrell Long, dated December 20, 2024 (“ Long Decl. ” or “ Long ”)
1004	U.S. Patent App. Pub. No. U.S. 2013/0282787 (“ Lippert ”)
1005	John R. Budenske et al., <i>A Method for the On-Line Use of Off-Line Derived Remappings of Iterative Automatic Target Recognition Tasks</i>

No.	Description
	<i>onto a Particular Class of Heterogeneous Parallel Platforms</i> , The Journal of Supercomputing, Vol. 12, No. 4 (Oct. 1998) (“ Budenske ”)
1006	U.S. Patent App. Pub. No. U.S. 2018/0074855 (“ Kambatla ”)
1007	Yu-Kwong Kwok et al., <i>A Semi-Static Approach to Mapping Dynamic Iterative Tasks onto Heterogeneous Computing Systems</i> , Journal of Parallel and Distributed Computing (2005) (“ Kwok ”)
1008	Mitchell D. Theys et al., <i>Mapping Tasks onto Distributed Heterogeneous Computing Systems Using a Genetic Algorithm Approach</i> , Solutions to Parallel and Distributed Computing Problems (A. Zomaya, F. Ercal & S. Olariu eds., 2001) (“ Theys ”)
1009	U.S. Patent No. 3,496,551 (“ Driscoll ”)
1010	Yuan-Chieh Chow and Walter H. Kohler, <i>Models for Dynamic Load Balancing in a Heterogeneous Multiple Processor System</i> , IEEE Transactions on Computers, Vol. C-28, No. 5 (May 1979) (“ Chow ”)
1011	David M. Nicol and Paul F. Reynolds, Jr., <i>Optimal Dynamic Remapping of Data Parallel Computations</i> , IEEE Transactions on Computers, Vol. 39, No. 2 (Feb. 1990) (“ Nicol ”)
1012	Arun C. Murthy, et al., Apache Hadoop YARN (2014) (“ Murthy ”)
1013	Morris Jette and Mark Grondona, <i>SLURM: Simple Linux Utility for Resource Management</i> , ClusterWorld Conference and Expo (“ Jette ”)
1014	Norbert Eicker et al., <i>The DEEP Project: An Alternative Approach to Heterogeneous Cluster-Computing in the Many-Core Era</i> , Concurrency and Computation: Practice and Experience (2016) (“ Eicker ”)
1015	U.S. Patent App. Pub. No. U.S. 2017/0262319 (“ Newburn ”)
1016	Carsten Clauss et al., <i>Dynamic Process Management with Allocation-internal Co-Scheduling towards Interactive Supercomputing</i> , Proceedings of the First Workshop on Co-Scheduling of HPC Applications (2016) (“ Clauss ”)

No.	Description
1017	Declaration of Rachel J. Watters dated December 13, 2024 (“ Watters Decl. ”)
1018	Samik Raychaudhuri, <i>Introduction to Monte Carlo Simulation</i> , Proceedings of the 2008 Winter Simulation Conference (2008) (“ Raychaudhuri ”)
1019	David W. Bauer, Jr. and Mojdeh Mohtashemi, <i>An Application of Parallel Monte Carlo Modeling for Real-Time Disease Surveillance</i> , Proceedings of the 2008 Winter Simulation Conference (2008) (“ Bauer ”)
1020	List of Challenged Claims
1021	U.S. Patent No. 5,881,227 (“ Brenner ”)
1022	U.S. Patent No. 7,047,299 (“ Curtis ”)
1023	N. Metropolis, <i>The Beginning of the Monte Carlo Method</i> , Los Alamos Science, Special Issue (1987) (“ Metropolis ”)
1024	U.S. District Court, Eastern District of Texas, Calendar Events Set for 4/1/2026-4/30/2026, Judge Robert W. Schroeder, III, Presiding
1025	Time to Milestones Report, Judge Robert W. Schroeder, III, DocketNavigator, https://search.docketnavigator.com/patent/judge/15325/15?print=true (last visited Dec. 19, 2024)
1026	Judge Schroeder Calendar Events June 15, 2026 (last visited May 14, 2025)
1027	Petitioner Sotera Stipulation, Served May 9, 2025

I. INTRODUCTION

The Patent Owner Response (“POR”) repeats arguments based on misreading a single sentence from Budenske. In doing so, it ignores Budenske’s teachings on how to distribute and redistribute sub-tasks among different processor types. Moreover, the relied-on sentence presents a simplifying assumption that on its face allows for assigning different sub-tasks to different processor types in a given iteration. Even Patent Owner’s (“PO”) expert testified at deposition that in Budenske “a portion of the subtasks can be mapped to one processor type and a portion of the subtasks to another processor type.” The POR also raises misplaced enablement arguments, which ignore Budenske’s teachings and are further disproven by evidence that others implemented Budenske’s teachings over a decade prior to the ’442 patent. As explained further below, these and other POR arguments should be rejected, and the Board should hold all challenged claims unpatentable.

II. GROUND 1

A. Claim 1: Budenske Teaches Distributing And Redistributing Sub-Tasks To Both Computation Nodes And Booster Nodes

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

The Petition shows how this requirement is satisfied by the Lippert-Budenske combination because *inter alia* Budenske teaches assigning each of a plurality of subtasks to either a computation or booster node of Lippert, with each sub-task assigned to its best-suited node for a given iteration. Petition (Paper 1), 13-14 (describing Budenske), 15-20 (describing the combination, including Budenske's role), 27-32; *see also* DI (Paper 10), 10-15.

The POR does not rebut this showing. The Board should (again) reject PO's argument that Budenske does not assign different sub-tasks to different processor types.

1. PO's Reading Would Contradict Budenske's Main Purpose

Budenske's mapping of sub-tasks to different processor types allows it to "fully utilize the architectural flexibility of such a [heterogeneous parallel computing] system" (EX1005, 390 (cited by Petition, 17-18)) to reduce execution times from iteration to iteration (EX1005, 390-91 (cited by Petition, 14, 17 & 32)). This purpose is wholly inconsistent with Patent Owner's unsupported theories that all subtasks must be assigned to the same processor type. Indeed, Patent Owner's strained reading would leave the flexible architecture grossly underutilized, thus wasting resources and directly contradicting Budenske's teachings. EX1030, ¶15.

Budenske repeatedly teaches mapping (and dynamically remapping) different subtasks to different processor types for a given computing iteration. For example,

Budenske expressly teaches to “assign each ... subtask to the processors where it is best suited for execution” (EX1005, 390) within a heterogenous system that may contain “up to four different types of processors” (*id.*, 391). To reduce execution times, Budenske teaches evaluating after each given iteration, whether “it is worthwhile to reconfigure the assignment of processing resources to subtasks to reduce execution time of the next iteration.” EX1005, 391 (quoted by DI (Paper 10), 10-11). EX1030, ¶14.

The information used to determine possible subtask-to-processor mappings also confirms that Budenske teaches mapping subtasks to different processor types in an iteration. EX1005, 392, 396-97 EX1030, ¶16. For example, the system relies on “the total number *and type of processors*” to “know *how many processors of each type it has available to assign*.” EX1005, 396 (emphasis added). It also relies on “the time it takes for *each processor in the platform* to communicate with *every other processor in the platform*” to “determine the expected inter-subtask communication times for possible mappings of subtasks.” *Id.*, 396-97.

2. PO’s Expert Agreed That Budenske Distributes Subtasks To Different Processor Types

PO’s expert, Dr. Kaeli, admitted in deposition that Budenske’s mappings involve distributing subtasks to different processor types:

Q. Does Budenske describe being able to distribute different subtasks to different processor types in the same computing iteration? ...

A. So Budenske describes that a portion of the subtasks can be mapped to one processor type and a portion of the subtasks to another processor type, then based on the input ... make a change in that allocation of ... which of the subtasks are mapped to which computing type.

EX1029, 102:8-18.

3. PO Continues To Rely On A Single Out-Of-Context Statement In Arguing Its Strained Read of Budenske

In maintaining this argument, PO continues to focus on a single sentence of Budenske: “For simplicity, it is assumed that if an implementation of a given subtask uses multiple processors, all processors will be of the same type.” EX1005, 392 (quoted by POR, 14, 27-28).

The Board should reject this argument, as it did at institution. The sentence “does not undermine or otherwise negate Budenske’s teachings about using a combination of different types of processors to perform a subtask” and “is merely providing a simplified version to illustrate the concept of an initial assignment and subsequent reassignments.” Paper 10, 20-22.

Budenske confirms the Board’s prior understanding. The sentence refers to only a single “implementation of a given subtask” and Budenske explains that an implementation of a subtask can change from iteration to iteration. *See* EX1005, 402 (the decision whether a remapping is desirable “is made in real time *after all subtasks’ implementations for the current iteration have finished executing*”); *id.*,

400 (at each iteration, the system “selects the implementation for the subtask that has the smallest expected execution time”). The sentence does not speak to *other* sub-tasks in an iteration and does not speak to *subsequent* implementations of the subtask (in later iterations). Thus, the sentence does not proscribe *other* sub-tasks being mapped to different processor types or that a given sub-task is re-mapped to a different type in subsequent iterations. EX1030, ¶¶17-19. *See also* EX1007, 156; EX1008, 81; EX1030, ¶¶20-21.

B. Claim 1: Lippert-Budenske Is Enabling

“Prior art disclosures are presumed enabling.” *Agilent Techs., Inc. v. Synthego Corp.*, 139 F.4th 1319, 1327 (Fed. Cir. 2025). Patent Owner “has the burden to prove nonenablement for such prior art.” *Apple Inc. v. Corephotonics, Ltd.*, 861 F. App’x 443, 450 (Fed. Cir. 2021) (non-precedential); *see also, e.g., Samsung Elecs. Co. v. Sung*, IPR2024-00533, Paper 39 at 24-25 (PTAB Sep. 17, 2025). Moreover, prior art references are not considered in isolation but rather “the prior art, *taken as a whole*, must enable a skilled artisan to make and use the claimed invention.” *Raytheon Techs. Corp. v. Gen. Elec. Co.*, 993 F.3d 1374, 1380 (Fed. Cir. 2021) (emphasis added).

For at least three reasons, the Board should reject Patent Owner’s argument that Budenske is not enabled (POR, 30-40).

1. Budenske's Pertinent Teachings Were Successfully Implemented Over A Decade Before The '442 Priority Date

Unlike the *Raytheon* case relied on by PO, Budenske provides fully enabling disclosures and does not rely on “unattainable” technology. *See Raytheon*, 993 F.3d at 1378, 1381-82 (finding prior art non-enabled when relied-on disclosures involved composite materials that were “undisputedly unattainable” before the critical date).

Indeed, more than a decade before the 2018 priority date of the '442 patent, Kwok described a successful implementation of Budenske's teachings: “*The objective of this paper is to implement and evaluate a semi-static methodology ... whose conceptual structure was proposed in [Budenske].*” EX1007, 79 (emphasis added) (denoting the Budenske approach as “On-Off”). Upon implementing Budenske, Kwok found that “the semi-static approach is effective in that it consistently outperformed a fast dynamic mapping heuristic, and yields reasonable performance....” EX1007, 94-95. Kwok presents experimental data comparing Budenske's approach against three other subtask scheduling algorithms in a heterogeneous computing system having four different processor types. EX1007, 81-83. Table 4 of Kwok compares Budenske's “On-Off” subtask mapping to an ECT-based mapping. The leftmost columns of FIG. 4 show that, for a particular iteration, 10 processed subtasks (designated s_0 - s_9) were mapped among four processor types (designated 0-3):

Table 4
Mappings for iteration 5 of Profile A used by the (a) On-Off approach (adopted after iteration 1), (b) ECT approach (adopted after iteration 4), (c) GA on-line approach, and (d) Ideal approach

s_i	Proc type	ρ	$t_u(s_i)$	Comm.	C_{uv}
(a) <i>On-Off approach</i>					
s_0	1	10	11,217	$s_0 \rightarrow s_1$	77
s_1	1	13	10,864	$s_0 \rightarrow s_2$	184
s_2	2	10	13,367	$s_0 \rightarrow s_3$	34
s_3	2	5	12,199	$s_0 \rightarrow s_4$	2551
s_4	3	6	4190	$s_0 \rightarrow s_5$	212
s_5	2	2	18,553	$s_0 \rightarrow s_8$	1523
s_6	0	9	12,175	$s_1 \rightarrow s_7$	64
s_7	2	8	18,346	$s_2 \rightarrow s_9$	3048
s_8	0	15	9355	$s_4 \rightarrow s_6$	1562
s_9	3	10	4738	$s_6 \rightarrow s_7$	2281
				$s_6 \rightarrow s_8$	223
				$s_6 \rightarrow s_9$	7600
(b) <i>ECT approach</i>					
s_0	0	10	8060	$s_0 \rightarrow s_1$	2321
s_1	3	4	9906	$s_0 \rightarrow s_2$	1526
s_2	1	16	18,558	$s_0 \rightarrow s_3$	792
s_3	3	16	4928	$s_0 \rightarrow s_4$	3839
s_4	3	4	4825	$s_0 \rightarrow s_5$	194
s_5	0	14	21,664	$s_0 \rightarrow s_8$	2281
s_6	0	2	41,444	$s_1 \rightarrow s_7$	1551
s_7	0	16	13,440	$s_2 \rightarrow s_9$	176
s_8	2	12	11,798	$s_4 \rightarrow s_6$	1562
s_9	1	14	14,655	$s_6 \rightarrow s_7$	167
				$s_6 \rightarrow s_8$	3040
				$s_6 \rightarrow s_9$	2534

EX1007, 90, Table 4 (annotated and excerpted). This confirms that Budenske’s “mapping” assigned different subtasks to different processor types in a given computing iteration. EX1030, ¶¶27-28.

The POR misapprehends several aspects of Kwok’s disclosures:

- Kwok describes a “fundamental research issue of how to select representative mappings off-line for on-line use.” EX1007, 80. Kwok continues: “To solve this problem, a novel dynamic parameter space partitioning and sampling

scheme is proposed” and that experimental results “indicate that this semi-static approach is effective.” *Id.* Therefore, Kwok itself solves its purported “fundamental research issue.” EX1030, ¶29.

- Kwok does not contemplate only a single “heterogeneity factor” (designated as h_{iu}) for the equation provided for predicting subtask execution times. *See* POR, 38 (citing EX1007, 83). Kwok teaches each “given machine” as having its own heterogeneity factor. EX1007, 83; EX1030, ¶31. This is confirmed by Figure 6(d) of Kwok, which presents heterogeneity factors for each of processor types 0-3 for each of subtasks s_1 - s_9 . EX1007, 88; EX1030, ¶31. These heterogeneity factors reflect each subtask’s affinity for a given processor type, i.e., the concept that a subtask will perform better or worse depending on the processor type executing it. EX1030, ¶32. Contrary to PO’s assertion (*see* POR, 38), affinity does not limit a sub-task to a single processor type. *Id.* Indeed, the ’442 patent recognizes this same concept of affinity in explaining that each sub-task “may be more or less suitable” for a given processor type. EX1001, 3:51-53.

2. Budenske Enables The Relied-Upon Disclosures

Budenske itself provides copious enabling disclosures. The Lippert-Budenske combination relies on Budenske primarily for functionalities performed by its HC Kernel. *See, e.g.,* Petition, 15-16. As the Petition explains, the HC Kernel or HC Kernel Monitor “(1) establishes an initial mapping of the sub-tasks onto the

processing resources of the heterogeneous computing system, (2) monitors the execution of the sub-tasks, and (3) at the end of each iteration, decides how to change the mapping of the sub-tasks onto the hardware based on information about dynamic parameters....” Petition, 13-14 (citing EX1005, 401-02).

Budenske describes the relied-on HC Kernel including, *inter alia*, its components, functionality, information required to implement it, and how such information is used. *See* EX1005, 396-404. And Budenske details how the HC Kernel interfaces and communicates with the Basic Kernel and how the Basic Kernel operates to execute an application task. *Id.*, 388, 402; EX1030, ¶¶22, 30; *see also* Petition, 33-34. The POR does not address these disclosures, which provides details on how to implement the relied-on teachings of Budenske.

Instead, PO misrepresents certain disclosures on exemplary architectures as allegedly restricting subtasks to a single processor type. For example, PO cites Budenske’s exemplary disclosure of six SHARC processors sharing a DRAM (POR, 33-34 (citing EX1005, 392)) and a statement about “symmetric” communications between same-type processors (POR, 34-35 (citing EX 1005, 392)). These disclosures don’t support PO’s position for multiple reasons. First, Budenske discloses processors sharing DRAM as part of an “example” (EX1005, 392) and nowhere requires same-type processors to share memory (e.g., DRAM). EX1030, ¶24. Second, Budenske’s broader architectural description makes clear that

processors of different types are interconnected via system buses (e.g., a VME bus), reflecting that data movement between processor types is both expected and supported. EX1005, 392 (“All the processors of all types will have communication paths to one another.”), *id.* (“To connect processors of different types, a VME bus can be used”). EX1030, ¶¶24-25 Third, neither of PO’s cited Budenske disclosures preclude sub-tasks from being assigned to different processor types in either a single computing iteration or in subsequent iterations. Budenske does not require data to remain resident in a particular memory associated with a given processor type and contemplates communications between different types of processors as well as data migration when resources are reconfigured. EX1005, 403 (emphasis added). EX1030, ¶¶24-26.

**3. Lippert-Budenske “As A Whole” Enables
A System That Meets The Challenged Claims**

Even Budenske if were non-enabling (it is not), the Lippert-Budenske combination provides enabling disclosure. Lippert discloses, *inter alia*, a heterogenous computing architecture including computation nodes, booster nodes, and an interface to flexibly couple the nodes. Petition, 15-16. The ’442 patent is a “development of the system described” in Lippert; does not purport to invent novel hardware; and admits that its computation nodes, booster nodes, and interconnect may be commercial, off-the-shelf components. *See* EX1001, 3:24-36; EX1029,

29:7-17, 30:6-8, 117:1-22; EX1032; EX1033; EX1034. EX1030, ¶34. In arguing enablement, PO identifies nothing missing from Lippert that would be required for implementation of Budenske's teachings. A POSITA would not expect undue experimentation in applying Budenske's techniques within Lippert's system, and the '442 patent's own lack of disclosure confirms this. EX1030, ¶¶34-35.

Indeed, the '442 patent itself shows that POSITAs knew how to distribute sub-tasks to different processor types in a heterogeneous system, as it does not disclose any particular way of distributing tasks to a mixture of processors. EX1030, ¶34. Instead, the '442 patent simply states that its application manager does so. *See* EX1001, 4:27-29 ("The application manager then performs a reassignment of the sub-tasks for a subsequent iteration of the task."). *See In re Epstein*, 32 F.3d 1559, 1568 (Fed. Cir. 1994) (affirming prior art enablement where "appellant's system fails to provide the same detailed information concerning the claimed invention" and that, in the absence of such a specific description, a POSITA "would know of the equipment and techniques to be used."). Moreover, the '442 patent discloses no novel computing hardware that is missing from Lippert-Budenske.

Finally, the POR and Dr. Kaeli assert that two publications "confirm[] the difficulty of mapping subtasks to different processor types." POR, 39-40 (citing EX2014, EX2015). These articles do not support PO's enablement argument. Each addresses problems associated with specific machine learning and deep neural

network applications requiring high computational performance. *See* EX2014, 1; EX2015, 1. Yet the only example of an iterative computing job disclosed in the '442 patent is a “Monte-Carlo” based simulation (EX1001, 3:44-49), which date to the 1940s (EX1003, ¶28). *See also* EX1029, 110:17-24; Petition, 10 (citing EX1018, 91; EX1019, 1029-36; EX1023, 125-30). The above articles do not suggest that Lippert-Budenske would be non-enabled for conventional iterative computing jobs such as a Monte-Carlo simulation. EX1030, ¶¶36-37. *See In re Morsa*, 803 F. 3d 1374, 1377 (Fed. Cir. 2015) (explaining that prior art “need only enable a single embodiment of the claim.”).

4. Patent Owner Fails To Meet Its Burden On Enablement

The touchstone of enablement is whether undue experimentation would have been required to practice the claimed invention, an issue that implicates up to eight relevant *Wands* factors. *In re Wands*, 858 F.2d 731, 737 (Fed. Cir. 1988). The POR neither cites *Wands* nor provides any meaningful analysis of the *Wands* factors. For example, while the POR mentions “undue experimentation” twice in passing (POR, 34, 35), it does not explain what experimentation would allegedly be required to enable the challenged claims based on Lippert-Budenske, or why such experimentation would be undue.

C. Claim 1: A POSITA Would Have Combined Lippert And Budenske

The Petition presents numerous well-supported reasons why a POSITA would have been motivated, with a reasonable expectation of success, to combine the relied-on teachings of Budenske and Lippert. Petition, 16-23. The POR addresses none of those reasons and instead largely rehashes its other arguments to argue against the combination. PO's arguments should be rejected.

First, Budenske's subtask mapping/remapping is not limited to using a single type of processor (*supra* Section II.A) and thus does not weigh against the combination (*see* POR, 40-42). Nor does an alleged "focus on ATR tasks" teach away from the combination (POR, 42), as Budenske teaches to use its remapping techniques in other "iterative" application domains (EX1005, 390). Moreover, the challenged '442 claims are not limited to any specific application; what the '442 patent supposedly "*allows*" (POR, 42) does not limit what its claims encompass. EX1030, ¶39.

Second, Kwok (EX1007) does not undermine the reasonable expectation of success, as both Budenske and Kwok solved and evaluated the "fundamental research" issue (*supra* Section II.B.1-2) and there was no hurdle in applying that solution in a system like Lippert's (II.B.3). Kwok states that Budenske did not evaluate two proposed methods for "determining the representative choices for the dynamic parameter values to use for each static mapping." EX1007, 79. PO does not

explain why implementing these methods would require undue experimentation. And Kwok proposed and evaluated a more complicated method for doing so (“statistically sampling each region of the multidimensional dynamic parameter space”) while describing Budenske’s proposed approach as the “simplistic approach.” *Id.* This confirms that a POSITA could implement Budenske’s proposed approaches or similar approaches for determining representative dynamic parameter values. EX1030, ¶40.

Third, Kambatla is relied on in Ground 2 for its teachings on resource management daemons. Whether Kambatla teaches “distributing sub-tasks in a more complicated heterogeneous architecture” is irrelevant to whether a POSITA would make the combination, as the Petition relies on Budenske and Lippert for those teachings. And while Kambatla states that accurately estimating computing resource requirements may be difficult, Budenske specifically address this issue. Budenske teaches to consider multiple information sources to determine a given subtask’s resource requirements and a given heterogeneous computing system’s resource capabilities to optimally distribute sub-tasks among various processor types. EX1030, ¶41; EX1005, 396; *supra* Section II.A.

Finally, non-obviousness is not supported by deposition testimony that fault tolerance through dynamic reassignment is “kind of cool.” POR, Fault tolerance is not claimed or even mentioned by the ’442 patent.¹

D. Claim 1: Budenske Generates In Real-Time A Further Distribution Of Sub-Tasks Using Information Relating To The Processing Of The Plurality Of Sub-Tasks

While executing a given task, Budenske’s HC Kernel monitors dynamic parameter values and, based on those observed values, selects and “*instantiates*” (i.e., *generates*) an appropriate mapping of subtasks between different processor types in the next computing iteration. Petition 30-32; *see also* EX1005, 395 (“as the application is executing, ... at the end of each iteration ... make a heuristically-based decision whether to ... select and *instantiate* a new mapping (for the next iteration)”) (emphasis added), EX1005, 398 (describing “use off-line precomputed mappings to *reconfigure resources in real time*.”) (emphasis added), *id.*, 403. This instantiation generates a further distribution of the sub-tasks to different processing resources, according to the selected mapping. *Id.*; EX1030, ¶¶42-43.

The POR does not dispute that claim 1’s “generating ...” is met by Budenske’s online instantiation of each selected mapping at each iteration, as argued by the Petition. That details of the mapping may have been previously determined off-line (POR, 46-47) does not change that each mapping is generated within the system

¹ The Lippert prior art does mention it. *See* EX1004, [0015].

during execution to carry out the distribution of resources specified by the mapping. Accordingly, in Budenske, information relating to the processing of sub-tasks is used to generate a distribution of sub-tasks in a further computing iteration. EX1030, ¶44.

E. Claim 1: Budenske’s Dynamic Parameters Are Information Relating To The Processing Of The Plurality Of Subtasks

Even under PO’s unduly narrow construction, Budenske’s dynamic parameters include “information relating to the processing of the plurality of sub-tasks.” Petition, 30-32. PO asserts that a single sentence of the ’442 patent’s specification distinguishes “results of the execution of the sub-tasks” and “the information from the daemons concerning the processing of the sub-tasks.” POR, 48 (quoting EX1001, 4:23-27). But the ’442 patent demonstrates that a POSITA would not understand the claimed “information” to be limited to “the information from the daemons concerning the processing of the sub-tasks.” EX1030, ¶45. For example, claim 5 further narrows claim 1 to claim that “daemons operate in the plurality of computation nodes and the plurality of booster nodes to generate the information.” EX1001, 6:12-14. Claim differentiation dictates that the claimed “information” is broader than just such information generated by daemons.

Further, the ’442 patent’s prosecution history shows that the claimed “information” includes at least the scalability of a subtask. EX1002, 174-75 (“[A]mended claims 1 and 9 allow for redistribution based on information relating

to the processing of the plurality of subtasks, *including adjusting the distribution of sub-tasks between computation nodes and boosters based on a scalability of the sub-task.*)” (emphasis added). EX1030, ¶46. While the ’442 patent contrasts “scalability of the sub-task” with “information learned about the processing of the sub-task” (e.g., EX1001, 4:48-52), the prosecution history unmistakably identifies scalability as “information relating to the processing of the plurality of subtasks.”

Moreover, Budenske’s dynamic parameters are not merely “resulting values of the data processing.” *See* POR 48-49. As the Petition explains, the dynamic parameters are information about the data to be processed at least because they “change during run time” and “have the most impact on the execution time of the subtasks in the application.” Petition 30-31 (quoting EX1005, 393). As their name suggests, these dynamic parameters are not merely resulting output data, but rather a *parameter* characterizing the processing of that data. EX1030, ¶47. This is confirmed by one example dynamic parameter, “whether the lighting conditions in the last image frame required requantization for better contrast for display” (EX1005, 393), which reflects an observation about the processing of the subtask.

A POSITA would therefore understand dynamic parameters disclosed in Budenske to be analogous to and used in the same way as the scalability of a subtask utilized in the ’442 patent. The ’442 patent explains that scalability is a property of the processed subtask may be determined automatically by the application manager

(EX1001, 4:38-43) and is used to determine distribution of subtasks to appropriate nodes; Budenske similarly determines its dynamic parameters during processing and adjusts subtask mapping accordingly. Therefore, a POSITA would understand “information relating to the processing of the plurality of subtasks” to include the dynamic parameters disclosed in Budenske. EX1030, ¶47.

F. Claim 3: Patent Owner Improperly Construed The Claim To Require Acts Never Contemplated By The '442 Patent

PO wrongly construes the claimed “resource manager dynamically alters further distribution” in a manner unsupported by the claim, arguing that “further distribution” refers to the “further distribution of the plurality of sub-tasks” recited in claim 1. POR, 50-51. PO’s narrowing construction should be rejected for at least three reasons.

First, the claim does not recite the language that PO seeks to import. The term “further distribution” in claim 3 lacks an article and has no antecedent to claim 1’s “a further distribution of the plurality of sub-tasks”. The term “further distribution” is not followed by “of the plurality of sub-tasks” in claim 3, and thus is not limited to a further distribution of the plurality of sub-tasks.

Second, PO’s construction would improperly alter the claim to require an action never contemplated by the '442 patent: that the resource manager (“RM”) redistributes sub-tasks among the booster and computation nodes. The '442 patent

repeatedly describes the RM as altering the assignment of computation and booster nodes to each other. EX1001, 2:26-29, 2:41-44, 2:51-55, 4:9-15, 4:18-23, 4:27-31; EX1030, ¶48. But it never describes the RM as altering the assignment of subtasks to computation and booster nodes.² Instead, the application manager (“AM”) is repeatedly described as performing this reassignment of sub-tasks. EX1001, 2:45-50, 4:18-37, 4:52-63, 5:4-9; EX1030, ¶49. Like claim 3, the ’442 patent’s “general description” broadly describes the resource manager altering “further distributions,” without specifying what is encompassed by that term. *See* EX1001, 2:22-25. A POSITA would therefore understand that the broadly worded “resource manager dynamically alters further distribution” encompasses a resource manager dynamically altering a further distribution of computation and booster nodes to each other. EX1030, ¶50. And the Petition explains how Lippert-Budenske satisfies Element 3.2 under this understanding. Petition, 39-41.

Third, the Petition’s arguments are consistent with the doctrine of claim differentiation. “Under the doctrine of claim differentiation, dependent claims are presumed to be of narrower scope than the independent claims from which they

² The ’442 patent describes RM performing the *initial assignment* of subtasks in the *first iteration* of the computing task but never describes the RM altering such assignment in later iterations. EX1001, 2:19-22.

depend.” *AK Steel Corp. v. Sollac & Ugine*, 344 F.3d 1234, 1242 (Fed. Cir. 2003). The Petition’s application of Lippert-Budenske to claims 3 and 4 do not violate this presumption because the “dynamically alters further distribution” limitation of claim 3 is broader than and encompasses the “dynamically alters assignment of the plurality of computation nodes and the plurality of booster nodes to each other” limitation of claim 4.

Nevertheless, even if PO’s narrow construction of claim 3 is correct, the combination of Lippert-Budenske still renders it obvious. As explained in the Petition, the AM of Lippert-Budenske determines a new distribution of sub-tasks and provides this new distribution to the RM of Lippert-Budenske such that the RM dynamically alters the distribution of computation nodes and booster nodes to each other “in order to implement the new mapping.” Petition, 39-40; *id.*, 33-39 (describing the AM and RM of Lippert-Budenske). A POSITA would have been further motivated to implement the RM to dynamically alter the further distribution of the plurality of sub-tasks in order to match how the RM dynamically altered assignment of computation nodes and booster nodes to each other, because doing so would “fully utilize the architectural flexibility” (EX1005, 390; EX1004, [0015]), and “provide fault tolerance in case of accelerator failures” (EX1004, [0003]). For example, if there is a “lack of resources” (EX1004, [0003]) due to other jobs being processed and/or booster failure, the RM may be limited in how it can dynamically

alter the computation node-booster assignment and may need to adapt the sub-task assignment according to what resources are available for the next iteration. The POSITA would have reasonably expected to succeed in implementing the RM in this way because the RM already “handles an initial mapping of sub-tasks” (Petition, 35), and for the same reasons discussed in the Petition. Petition, 21-23; EX1030, ¶51.

G. Claim 5

**1. Lippert-Budenske Renders Obvious
Computation Node And Booster Node Daemons**

The Petition explains that both Lippert and Budenske disclose collecting processing information, including detecting computing capacities of boosters and dynamic parameters. Petition, 41-43. This suggests the existence of logic for performing such information collection and, as the Petition and Dr. Long explain, a POSITA would understand the need to include a logic for performing such detection and would implement such logic as a background process for performing that specific task (i.e., a daemon). *Id.*; EX1003, ¶90; *see also* EX2021, ¶90 (Dr. Kaeli stating that a daemon “is sometimes used to refer to as [sic] a background process.”). Lippert explicitly discloses daemons, explaining that its resource manager may be implemented as a daemon. EX1004, [0014]; *see also* Petition, 15. And PO does not dispute that daemons were conventional in the art. *See* Petition 9-10; EX1016;

EX1001, 3:13-18 (describing EX1016 as “disclos[ing] daemons as addressed herein”). The Petition and Dr. Long explain why a POSITA would be motivated to implement such background processes at individual computation and booster nodes, i.e. to “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.” Petition, 42-43; EX1003, ¶90; EX1030, ¶52. The POR does not dispute this explained motivation for a POSITA to implement daemons in such a manner, instead arguing that “there were other ways to implement the proposed combination.” POR, 56. And the existence of possible alternatives does not negate the POSITA’s motivation to implement daemons as presented in the Petition. *In re Mouttet*, 686 F. 3d 1322, 1334 (Fed. Cir. 2012) (“[T]he mere disclosure of alternative designs does not teach away.”) (quotation omitted).

H. Claim 9

For claim 9, the POR relies on “the same reasons” discussed with regard to claims 1 and 5. For the reasons discussed in Sections II.A-E & G, *supra*, those arguments also fail with respect to claim 9.

III. GROUND 2

PO asserts that “Ground 2 challenges should be rejected for the same reasons as identified for Ground 1.” POR, 58. Those arguments fail for the same reasons explained for Ground 1, above.

A. Claim 3

Regarding claim 3, the POR misunderstands the proposed combination of Lippert-Budenske-Kambatla. *See* POR, 59. The proposed combination relies on the heterogeneous computing hardware elements (computation nodes, booster nodes, and communication infrastructure) of Lippert. Petition, 56. As the Petition explains, Lippert-Budenske-Kambatla relies on the same resource manager functionality of Lippert-Budenske of dynamically altering further distribution of computing nodes and booster nodes, and a POSITA would be motivated to include such a module as part of a central resource manager as taught by Kambatla to “achiev[e] the benefits of Kambatla’s resource manager and the dynamic assignment of Lippert-Budenske’s resource manager.” Petition, 66-67; EX1030, ¶55. The POR does not dispute that a POSITA would be motivated to make such combination.

B. Claim 5

The POR asserts that Lippert-Budenske-Kambatla fails to disclose daemons in computation and booster nodes. The POR argues that Kambatla does not “address the concept of heterogeneous computing for processing of tasks.” POR, 60. First,

that assertion is incorrect because Kambatla discloses that its “worker nodes may have different hardware.” EX1006, [0059], [0071]. Further Kambatla discloses that the aggressiveness of its scheduling can be controlled through variable parameters designated as T_{alloc} and T_{preempt} . Kambatla discloses that these parameters may be “specific to a worker node given the characteristics and/or capabilities of the worker node” and that the parameters can “be generalized to a vector, with dimensions representing *different resource types*.” *Id.* Kambatla therefore contemplates scheduling based on worker nodes comprising different resource types in its platform. EX1030, ¶56. Second, whether Kambatla addresses the concept of heterogeneous computing is irrelevant. PO does not assert that Kambatla is not analogous art and that a POSITA would therefore not seek to combine it with Lippert-Budenske, and PO does not explain why a POSITA would find Kambatla’s teachings concerning the facilitation of inter-element communications (*see* Petition, 59) to be inapplicable to the heterogeneous platforms taught by Budenske and Lippert. PO does not dispute that each worker node disclosed in Kambatla (i.e., each processing node) includes a daemon and a POSITA implementing the teachings of Kambatla into Lippert-Budenske would implement such a daemon at each processing node of Lippert-Budenske, regardless of whether such nodes are booster or computation nodes. EX1030, ¶56. And as explained above, the ’442 patent does not disclose any novel node architecture (allowing computation and booster nodes

to comprise off-the-shelf commercial processors) that would prevent a POSITA from doing so. On the contrary, the '442 patent acknowledged that its use of daemons is conventional and reflected in prior art. *See* Petition 9-10; EX1016; EX1001, 3:13-18.

The POR inaccurately paraphrases (POR, 61) and ignores much of the Petition's analysis concerning a POSITA's motivation to combine the daemons of Kambatla with Lippert-Budenske (*see* Petition, 58-61, 70). For example, the Petition explains that Kambatla complements Lippert-Budenske through "provid[ing] techniques for optimally scheduling tasks across the processing resources." Petition 58, (citing EX1006, [0004], [0017]). A POSITA would not understand such scheduling techniques to be inapplicable to heterogeneous computing systems. EX1030, ¶57. The Petition also describes "the need to facilitate inter-element communication," and while other ways to facilitate information gathering were known, the existence of potential alternatives does not negate a POSITA's motivation to combine. EX1030, ¶52. And the third motivation presented in the Petition does not rely on different processor types. Instead, the Petition describes how Kambatla and Lippert-Budenske share principles addressing allocating cluster computing resources to jobs that typically include multiple tasks and the problem of resource fragmentation and severe under-utilization of computing resources in a cluster. Petition, 59-61. These principles apply to the heterogeneous computing

platforms taught by Lippert and Budenske regardless of whether Kambatla teaches a platform including different processor types. EX1030, ¶57.

C. Claim 9

The POR asserts that Ground 2’s challenge to claim 9 fails because allegedly it does not remedy the claim 1. This argument fails for the same reasons explained for claim of Ground 1 above.

The POR also asserts that Kambatla fails to disclose “addressing distribution of sub-tasks to different processor types.” POR, 62. But the Petition fully addresses how Kambatla would combine with Lippert-Budenske to do so. *See* Petition, 64 (“[A] POSITA would have been motivated to implement the dynamic assignment of sub-tasks to the compute nodes and booster nodes of a heterogeneous computing system, as taught by Lippert-Budenske, with the application manager of Kambatla in order to efficiently enable and manage the execution of multiple such applications on the plurality of computing resources.”). Further, as explained in Section III.B., *supra*, Kambatla teaches processors of different types and whether Kambatla does so is irrelevant, regardless. EX1030, ¶58.

D. A POSITA Would Have Been Motivated To Combine Kambatla With Lippert-Budenske

As explained in Section III.B., *supra*, Kambatla teaches processors of different types and, regardless, whether Kambatla does so is irrelevant. A POSITA

would consider Kambatla's teachings to be relevant to the heterogenous platforms of Lippert and Budenske and would therefore combine those teachings with Lippert-Budenske with a reasonable expectation of success. EX1030, ¶59

**IV. PARALLEL DISTRICT COURT
CLAIM CONSTRUCTION POSITIONS**

The Board has designated as precedential the Director's order in *Revvo Techs., Inc. v. Cerebrum Sensor Techs, Inc.*, addressing situations where a petitioner adopts a different claim construction position before a district court and the Board. IPR2025-00632, Paper 20 (Director Nov. 3, 2025). Here, Petitioner has offered clarifying constructions in district court for the terms "first computing iteration" and "further computing iteration." *See* EX1031, 24-25 (proposing "an initial execution of the plurality of sub-tasks" and "a repeated execution of the plurality of sub-tasks," respectively). Those constructions are consistent, however, with the plain meaning and how Petitioner has mapped the claims in this IPR. They were proffered in district court simply to confirm for the jury that "the plurality of subtasks" refer to the same plurality of subtasks.

Respectfully submitted,

Dated: December 23, 2025

By: /Andrew M. Mason/
Andrew M. Mason (Reg. No. 64,034)
andrew.mason@klarquist.com

Counsel for Petitioner

**CERTIFICATE OF COMPLIANCE WITH
TYPE-VOLUME LIMITATION PURSUANT TO 37 C.F.R. § 42.24**

This brief complies with the type-volume limitation of 37 C.F.R. § 42.24(a)(1)(i).

The brief contains **5,581** words, excluding the parts of the brief exempted by 37 C.F.R. § 42.24 (a).

The brief has been prepared in a proportionally spaced typeface using Microsoft Word for O365 in a 14-point Times New Roman font.

Dated: December 23, 2025

By: /Andrew M. Mason/

Andrew M. Mason (Reg. No. 64,034)
andrew.mason@klarquist.com
KLARQUIST SPARKMAN, LLP
One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, Oregon 97204
Tel: 503-595-5300
Fax: 503-595-5301

Counsel for Petitioner

CERTIFICATE OF SERVICE
IN COMPLIANCE WITH 37 C.F.R. § 42.6(E)(4)

The undersigned certifies that on December 23, 2025, a complete copy of the **PETITIONER'S REPLY TO THE PATENT OWNER RESPONSE** and **EXHIBITS 1028-1034** were served on counsel for the Patent Owner via electronic mail as follows:

Michael H. Heim
Christopher L. Limbacher
Heim, Payne & Chorush, LLP
1111 Bagby Street, Suite 2100
Houston, TX 77022
Email: mheim@hpcllp.com
climbacher@hpcllp.com

By: /Andrew M. Mason/
Andrew M. Mason (Reg. No. 64,034)
andrew.mason@klarquist.com
KLARQUIST SPARKMAN, LLP
One World Trade Center, Suite 1600
121 S.W. Salmon Street
Portland, Oregon 97204
Tel: 503-595-5300
Fax: 503-595-5301

Counsel for Petitioner