Filed: February 10, 2021

Filed on behalf of
Patent Owner Advanced Cluster Systems, Inc.
By: Jon W. Gurka
Ted M. Cannon
Cheryl T. Burgess
KNOBBE, MARTENS, OLSON & BEAR, LLP
2040 Main Street, Fourteenth Floor
Irvine, CA 92614
Tel.: (949) 760-0404
Fax: (949) 760-9502
Email: BoxZTANNL.017LP2@knobbe.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NVIDIA CORPORATION,
Petitioners,

v.

ADVANCED CLUSTER SYSTEMS, INC.
Patent Owner

Case No. IPR2021-00019
U.S. Patent 10,333,768

PATENT OWNER PRELIMINARY RESPONSE

TABLE OF CONTENTS

Page No.

TABLE OF CONTENTS
(*cont'd*)

Page No.

TABLE OF CONTENTS
(*cont'd*)

Page No.

TABLE OF AUTHORITIES

Page No(s).

TABLE OF AUTHORITIES
(*cont'd*)

Page No(s).

OTHER AUTHORITIES

EXHIBIT LIST

Exhibit No.	**Description**
2001	Declaration of Jaswinder Pal Singh, Ph.D.
2002	*Curriculum Vitae* of Jaswinder Pal Singh, Ph.D.
2003	Exhibit Number Not Used
2004	Exhibit Number Not Used
2005	Exhibit Number Not Used
2006	Exhibit Number Not Used
2007	Declaration of Vineer Bhansali, Ph.D.
2008	Declaration of John Bancroft
2009	Dauger Research, *Supercompute Mathematica* (2009) ("SEM Poster")
2010	Dauger Research, *Pingpong MPI Benchmark – SEM v. "grid"* (2009) ("SEM PingPong Benchmark")
2011	ACS and Dauger Research, *Supercomputing Engine for Mathematica* ("SEM Manual")
2012	*Supercomputing Engine for Mathematica* (submitted in 2007) ("SEM White Paper")
2013	ACS, *SET™: A Software Framework for Parallel Programming** (Ver. 021313-2) ("SET White Paper")
2014	ACS, *SET™ v1.0: Supercomputing Engine Technology™ Reference Manual* ("SET Manual")
2015	Dean Dauger, *Supercomputing Engine Technology: HPC for the Missing Middle* (Dec. 2011) ("SET Presentation")
2016	ACS, *SET™ SOFTWARE* (Ver. 121127-2) ("SET Datasheet")

Exhibit No.	Description
2017	*Parallel Bars*, The Economist (June 4, 2011)
2018	Excerpt from Dean Dauger, *Supercomputing Engine Technology: HPC for the Missing Middle* (Dec. 2011) ("SET Presentation II")
2019	DOE, *Reviews of (2a) SET Plasma Simulation Utilizing Interactive Supercomputing Implementation* (saved on Jan. 1, 2015)
2020	SET Datasheet (Ver. 121127-1)
2021	DOE, *Reviews of (2b) SET Expanded Validation to Demonstrate SET (Supercomputing Engine Technology) Advantages when inexperienced programmers are utilizing it to parallelize codes* (saved on Jan. 1, 2015)
2022	DOE, *Reviews of (2b) SET (Supercomputing Engine Technology) for Microsoft Windows* (saved on Jan. 1, 2015)
2023	Yuko Matsuda, *Supercomputing Engine for Mathematica: an independent report* (Nov. 17, 2008)
2024	Excerpt from Zvi Tannenbaum, *Helping the Software Industry Transition to Multicore Parallelism* (Feb. 2011)
2025	Excerpt from Zvi Tannenbaum, *SET™ Windows Project Narrative: Topic 2b*.
2026	Excerpt from U.S. Department of Energy, Small Business Innovation Research (SBIR) and Small Business Technology Transfer (STTR) Program, *Topics: FY 2015 Phase I Release 1* (Ver. 1 July 14, 2014)
2027	Declaration of Dean Dauger, Ph.D.
2028	Objective Evidence of Non-Obviousness Nexus Claim Chart for Claim 1 of the '768 Patent Re Supercomputing Engine for Mathematica ("SEM")

Exhibit No.	**Description**
2029	Objective Evidence of Non-Obviousness Nexus Claim Chart for Claim 1 of the ’768 Patent Re Supercomputing Engine Technology (“SET”)

Patent Owner Advanced Cluster Systems, Inc. ("ACS") submits this Preliminary Response in opposition to the Petition to institute an *inter partes* review of U.S. Patent No. 10,333,768 ("the ’768 patent").

I. INTRODUCTION AND SUMMARY OF ARGUMENT

The Petition should be denied at least because it fails to cite any prior art permissible in an IPR that discloses fundamental features of the claimed invention.

The ’768 patent describes and claims a fundamental improvement over previous efforts to allow multiple nodes to work together to perform computations in parallel. Specifically, the inventors designed a mechanism to allow the nodes of a computer cluster to communicate tasks and data with one another in a peer-to-peer architecture. According to the background section of the patent, one previous attempt to allow multiple nodes to work together to perform computations was "a form of grid computing known as ‘distributed computing.’" Ex. 1001 at 1:44-62. The form of grid computing described in the patent relied on a "master node that manages a plurality of slave nodes or computational nodes." *Id.* at 1:51-53. The "master kernel . . . handles all input, output, and scheduling of the other kernels (the computational kernels or slave kernels). Computational kernels receive commands and data only from the node running the master kernel." *Id.* at 1:55-59. The nodes "generally do not communicate with one another as peers." *Id.* at 1:46-47. The patent distinguishes this form of grid computing from a "computer cluster having peer-to-peer node architecture." *Id.* at 12:33-40. Ex. 2001 ¶ 36.[1]

The '768 patent expressly claims the "peer-to-peer node architecture" disclosed in the specification. Specifically, claim 1 recites a computer cluster with multiple nodes and "a mechanism for the nodes to communicate results of mathematical expression evaluation with each other using a peer-to-peer architecture." Claim 1 further recites that "one or more of the nodes are configured to . . . after accepting user instructions, communicate at least some of the user instructions using the mechanism for the nodes to communicate with each other." Ex. 2001 ¶ 37.

[1] Herein, a single citation to the expert declaration of Dr. Jaswinder Pal Singh is provided at the end of each paragraph that is supported by Dr. Singh's testimony. Dr. Singh is a longstanding professor of Computer Science at Princeton University who has a Ph.D. in Electrical Engineering from Stanford University and extensive knowledge of and experience in the field of the invention. As in most patentability challenges, the predominant issues in this case are technical issues for which expert analysis is relevant. With respect to the technical issues addressed herein, Patent Owner has largely adopted Dr. Singh's expert analysis.

The specification discloses a particular mechanism for the nodes to communicate using a "peer-to-peer architecture." Specifically, multiple "cluster node modules" establish connections from each cluster node module to every other cluster node module and exchange messages in a process that "provides the peer-to-peer behavior of the cluster node modules." Ex. 1001 at 23:51-52, 24:39-40, 24:52-53, 25:1-2, 25:9-10, 25:23-24, 25:37-38. Dependent claim 4 recites the "cluster node modules." Ex. 2001 ¶ 38.

A POSITA would understand, in view of the specification, that "peer-to-peer behavior" is an essential feature of both the "peer-to-peer architecture" limitation of claim 1 and the narrower "cluster node modules" limitation of claim 4. The "peer-to-peer behavior" includes at least that each node can communicate tasks and data with other nodes without the tasks and data being required to go through a central server or master node. Ex. 2001 ¶ 39.

Accordingly, consistent with the inventors' fundamental architecture of the invention, the Board should construe the "peer-to-peer architecture" limitation of claim 1 to require at least "an architecture in which each node can communicate tasks and data with other nodes without the tasks and data being required to go through a central server or master node." Similarly, the Board should construe the slightly narrower "cluster node module" limitation of claim 4 to mean "a module that cooperates with other cluster node modules to establish intercommunication among nodes in a computer cluster and to exchange messages such that each node can communicate tasks and data with other nodes without the tasks and data being required to go through a central server or master node."

The prior art cited in the Petition does not disclose either the "peer-to-peer architecture" limitation of claim 1 or the "cluster node module" limitation of claim 4. Petitioner attempts to rely on a software product called "Distributed Maple." However, it is immediately apparent that although the programmers of Distributed Maple wrote a journal article ("Schreiner 1") discussing Distributed Maple in detail, no document about Distributed Maple discloses the "peer-to-peer architecture" or "cluster node module" limitations.

Because no Distributed Maple document discloses the fundamental features claimed in the '768 patent, Petitioner attempts to cobble together ***13*** documents (Exs. 1008-1020), including the Schreiner 1 journal article, a purported reference manual, a document about task logging, and even purported source code and "ReadMe" files included with the Distributed Maple software package, and to misinterpret those documents as allegedly collectively disclosing every claim limitation. The Petition fails to establish that at least seven of the documents (collectively labeled the "Distributed Maple Code") are even prior art, as the Petition presents inadequate evidence that those documents were accessible to the relevant public at the relevant time. Thus, the Board should disregard the Distributed Maple Code and may deny the Petition on that basis alone.

Further, tacitly conceding that even the combination of those 13 documents does not disclose all the claim limitations, Petitioner relies on 2020 testimony of a programmer of Distributed Maple, Dr. Wolfgang Schreiner, that the Distributed Maple software package had features that Schreiner 1 and the other documents omitted years earlier. Petitioner's attempt to weave together numerous disparate documents and 2020 testimony to create a narrative that the Distributed Maple software package allegedly had the features of the claimed invention is a transparent attempt to improperly present a "public use" validity challenge, in violation of the IPR statute. The Board should reject Petitioner's improper public use challenge.

Even if the 13 Distributed Maple references are accepted as prior art and cobbled together, they do not, individually or collectively, disclose the "peer-to-peer architecture" or "cluster node module" limitations, which, as properly construed, require each node to be able to "communicate tasks and data with other nodes without the tasks and data being required to go through a central server or master node." In fact, the prior art expressly discloses that the "schedulers" that the Petition relies upon distribute tasks through a central server or master node. Specifically, the prior art discloses: "All remote schedulers send new tasks to the root node scheduler which distributes them among all machines." Ex. 1008 at 316. A POSITA would understand the disclosed "root node scheduler"—which is necessarily involved in at least the distribution of tasks—is a "central server" or "master node." Therefore, the cited prior art does not disclose the claimed "peer-to-peer architecture" or "cluster node modules" able to "communicate tasks and data with other nodes without the tasks and data being required to go through a central server or master node."

At least because no permissible prior art discloses these two fundamental claim limitations, the Petition should be denied. The Petition should also be denied for additional reasons, as explained below.

II. CLAIM CONSTRUCTION

A. "a mechanism for the nodes to communicate . . . with each other using a peer-to-peer architecture" (all challenged claims)

Claim 1 recites "***a mechanism for the nodes to communicate*** results of mathematical expression evaluation ***with each other using a peer-to-peer architecture***" and "wherein one or more of the nodes are configured to . . . communicate at least some of the user instructions using ***the mechanism for the nodes to communicate with each other***" (emphases added). With respect to these two limitations, there are two claim construction issues relevant to the Board's decision whether to institute an IPR: (1) determining that both limitations refer to the same "mechanism for the nodes to communicate . . . with each other using a peer-to-peer architecture" and (2) determining the meaning of "peer-to-peer architecture." Ex. 2001 ¶ 40.

1. The "wherein" clause refers back to the same "mechanism for the nodes to communicate . . . with each other using a peer-to-peer architecture" introduced earlier in the claim.

A POSITA would understand that the "wherein" clause uses the standard claim-drafting technique of using a shorthand phrase introduced by the definite article "the"—specifically, "the mechanism for the nodes to communicate with each other"—to refer to a limitation recited earlier in the claim. A POSITA would also understand that the ***only*** earlier limitation of claim 1 that recites a "mechanism" that provides antecedent basis for "the mechanism" of the "wherein" clause, is "a mechanism for the nodes to communicate . . . with each other using a peer-to-peer architecture." Accordingly, the Board should determine that the "wherein" clause refers back to the same "mechanism for the nodes to communicate . . . with each other using a peer-to-peer architecture" recited earlier in the claim. This means that "results of mathematical expression evaluation" and "at least some of the user instructions" must be communicated using the claimed "mechanism for the nodes to communicate . . . with each other using a peer-to-peer architecture."[2] Ex. 2001 ¶ 41.

---

[2] No constructions of "results of mathematical expression evaluation" and "at least some of the user instructions" are needed to decide whether to institute

2. "peer-to-peer architecture"

The claimed communication mechanism must use a "peer-to-peer architecture." A POSITA would understand that, in the context of cluster computing, the adjective "peer-to-peer" ordinarily means that each node can communicate tasks and data with other nodes without the tasks and data being required to go through a central server or master node. A POSITA would also look to the specification to confirm this understanding.

As explained below, a POSITA would conclude that at least communicating tasks and data is a primary function of the "peer-to-peer architecture" of the '768 patent. Therefore, "peer-to-peer architecture," in the context of the '768 patent, should be construed to require at least "an architecture in which each node can communicate tasks and data with other nodes without the tasks and data being required to go through a central server or master node." Ex. 2001 ¶ 43.

A POSITA would understand that the "peer-to-peer architecture" required by claim 1 is one feature of the interconnected "cluster node modules" disclosed by the

---

an IPR. However, Patent Owner reserves the right to propose constructions for these phrases in other administrative or judicial proceedings, or in the Patent Owner Response in this proceeding if an IPR is instituted.

’768 patent and recited by claim 4.[3] Accordingly, the ’768 patent’s disclosure of the “peer-to-peer architecture” enabled by the “cluster node modules” is instructive of what “peer-to-peer architecture” means in the context of the ’768 patent. Figure 2 illustrates that each cluster node module is connected to every other cluster node module, as shown by the yellow highlighting in the annotated figure below.

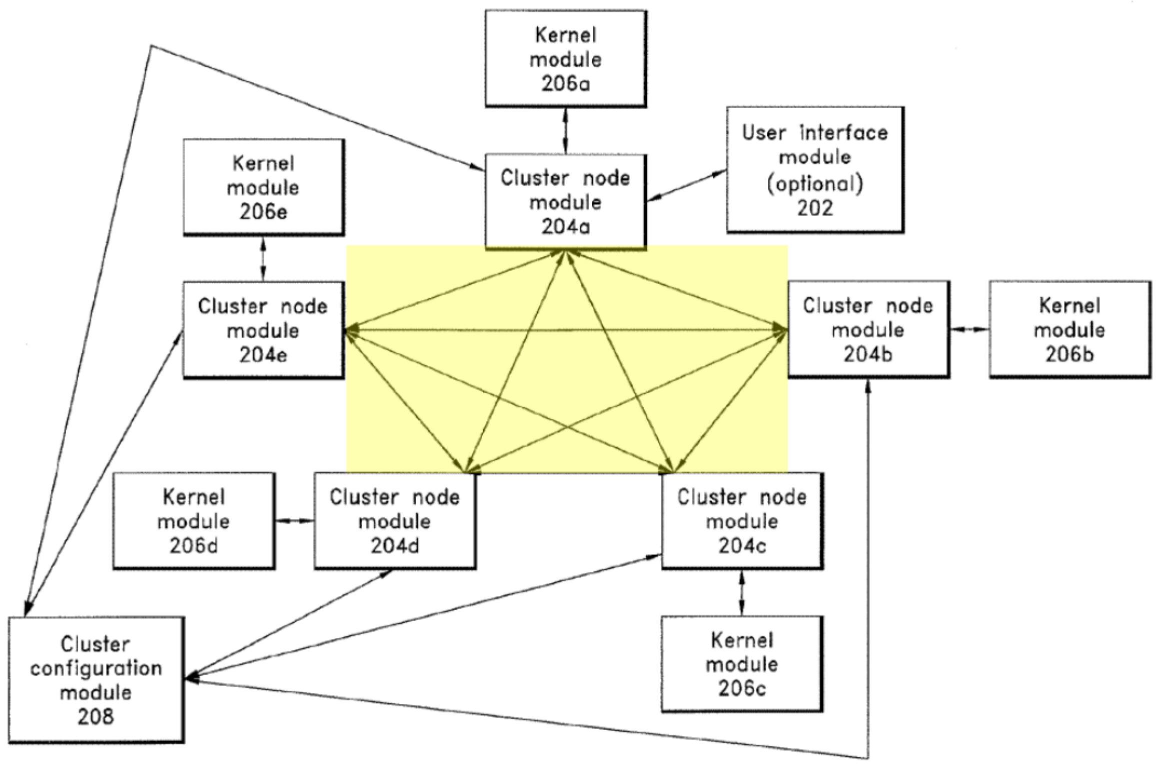


FIG. 2

[3] Claim 1 is broader than claim 4 because, while it requires a “peer-to-peer architecture,” it does not require that architecture to be implemented using the “cluster node modules” recited by claim 4.

Ex. 1001, Fig. 2 (highlighting added). The specification confirms that "each cluster node module 204a-e is connected to all other cluster node modules" and "[t]he cluster node modules 204a-e establish communication with one another" through "direct connections" between each cluster node module. Ex. 1001 at 23:13-14, 23:51-56. Ex. 2001 ¶ 44.

As illustrated and described, the cluster node modules provide a direct connection from each node to every other node. In addition, the specification discloses that a process for passing messages among the cluster node modules "provides the peer-to-peer behavior of the cluster node modules," allowing them to "interact on a pair-wise or collective basis." Ex. 1001 at 25:22-41. Ex. 2001 ¶ 45.

In view of the specification, a POSITA would understand that the message-passing process allows each cluster node module to communicate tasks and data with other nodes without the tasks and data being required to go through a central server or master node. Specifically, the specification discloses that "[c]ommunications can occur between any two or more cluster node modules" and that "[e]ach of the cluster node modules 204a-e is in communication with respective kernel modules 206a-e," thereby enabling "MPI calls and advanced cluster commands" to be "used to parallelize program code received from an optional user interface module 208 and ***distribute tasks*** among the kernel modules 206a-e." Ex. 1001 at 6:9-25 (emphasis added). Ex. 2001 ¶ 46.

In addition, the specification distinguishes a peer-to-peer architecture from the master-slave architecture typically used for "a form of grid computing known as 'distributed computing.'" Ex. 1001 at 1:44-62; *see also id.* at 12:30-33 (gridMathematica connects kernels "in a master-slave relationship rather than a peer-to-peer relationship"). The specification describes the following characteristics of the master-slave architecture of grid computing:

> Grid computers include at least one node known as a master node that manages a plurality of slave nodes or computational nodes. In gridMathematica, each of a plurality of kernels runs on a single node. ***One kernel is designated the master kernel, which handles all input, output, and scheduling of the other kernels (the computational kernels or slave kernels). Computational kernels receive commands and data only from the node running the master kernel.***

Id. at 1:51-59 (emphasis added). By distinguishing the peer-to-peer architecture of the invention from the master-slave architecture, the specification implicitly indicates that the peer-to-peer architecture does not have the identified characteristics of the master-slave architecture. Accordingly, by contrast to the master-slave architecture, each node in the disclosed peer-to-peer architecture is able to handle "scheduling" and communicating (including distributing, sending, and receiving) "commands and data" without requiring a central server or master node. A POSITA would understand that the communication of tasks and data falls within these "scheduling" and communicating "commands and data" functions. Ex. 2001 ¶ 47.

Accordingly, a POSITA would understand that the communication of tasks and data are primary functions that the nodes of the disclosed peer-to-peer architecture of the '768 patent can handle without requiring a central server or master node. The Board should construe "peer-to-peer architecture," in the context of the '768 patent, to require at least "an architecture in which each node can communicate tasks and data with other nodes without the tasks and data being required to go through a central server or master node." Ex. 2001 ¶ 48.

B. "cluster node module" (claim 4)

Dependent claim 4 recites "wherein each of the nodes comprises one or more cluster node modules." The phrase "cluster node module" was not a common or well-known technical phrase having any specific meaning in the relevant art at the time of the invention. Indeed, even today, more than 14 years after the earliest effective filing date of the '768 patent, a search for patents or patent application publications that use the phrase "cluster node module" returns only patents and applications filed by the inventors of the '768 patent. Accordingly, there was no ordinary meaning of "cluster node module" in the relevant field at the relevant time. Further, while the individual terms "cluster," "node," and "module," were known in the relevant field, a POSITA would understand that no ordinary meaning for the combined phrase "cluster node module" could reliably be composed by attempting to combine the meanings of the individual terms. Ex. 2001 ¶ 49.

In view of the claim language and specification, a POSITA would understand that the inventors coined the phrase "cluster node module" to encapsulate essential features of the modules that interconnect the nodes in a preferred embodiment of the invention.[4] Because "cluster node module" is a coined phrase, reliance on the specification is necessary to ascertain its meaning. *3M Innovative Props. Co. v. Tredegar Corp.*, 725 F.3d 1315, 1321 (Fed. Cir. 2013) ("Idiosyncratic language, highly technical terms, or terms coined by the inventor are best understood by reference to the specification.") As explained below, in view of the specification, the Board should construe "cluster node module" to mean "a module that cooperates with other cluster node modules to establish intercommunication among nodes in a computer cluster and to exchange messages such that each node can communicate tasks and data with other nodes without the tasks and data being required to go through a central server or master node." Ex. 2001 ¶ 50.

The specification discloses several ***optional*** components of the cluster node modules. For example:

---

[4] Several claims of the '768 patent, including claim 1, recite "a plurality of nodes" but not "cluster node modules."

> ***FIG. 3 shows one embodiment*** of a cluster node module 204 implementing MPI calls and advanced MPI functions. ***In the embodiment shown in FIG. 3***, cluster node module 204 includes MPI module 302, advanced functions module 304, received message queue 306, and message receiving queue 308.

Ex. 1001 at 12:41-46 (emphases added). Because the components depicted in Figure 3 are part of "one embodiment," a POSITA would understand that they are optional components of the cluster node modules. Dependent claims of U.S. Patent No. 8,082,289 (which is related to the '768 patent) specifically reciting that the cluster node modules comprise an MPI module, advanced functions module, received message queue, and message receiving queue also demonstrate that these components are optional parts of the cluster node modules. IPR2020-01608, Ex. 1001, at claim 8 (received message queue), claim 9 (message receiving queue), claim 13 (advanced functions module), claim 26 (MPI module). Accordingly, a POSITA would understand that the basic "cluster node modules" do not require an MPI module, advanced functions module, received message queue, and message receiving queue. Thus, if the Petition suggests that a "cluster node module" necessarily comprises these components (*see* Pet. at 52), the Petition is incorrect.[5] Ex. 2001 ¶ 51.

Rather than interpreting "cluster node module" to include optional components such as those depicted by Figure 3, a POSITA would look to the specification to limit the "cluster node module" to its essential features. The specification unambiguously discloses that cluster node modules are configured to establish intercommunication with one another, communicate messages among themselves and with kernels, and, through such message-passing, "provide[] the peer-to-peer behavior of the cluster node modules." Ex. 1001 at 23:51-52, 24:39-40, 24:52-53, 25:1-2, 25:9-10, 25:23-24, 25:37-38. Significantly, the specification does not say these disclosed features apply just to "one embodiment" of the cluster node modules or otherwise suggest they are optional. Therefore, a POSITA would understand that the capability to establish intercommunication among all cluster node modules and to exchange messages to provide "peer-to-peer behavior" are essential features of the cluster node modules. Ex. 2001 ¶ 52.

---

[5] The Petition does not commit to ***any*** definitive claim construction of "cluster node module." The Petition merely states that "a 'cluster node module' includes code ***relating to*** . . . and ***can*** also include" the components depicted by Figure 3. Pet. at 52.

A POSITA would further examine both the ordinary meaning of "peer-to-peer" and the specification to determine what is meant by the cluster node modules' "peer-to-peer behavior." As explained above with respect to the "peer-to-peer architecture" limitation, a POSITA would conclude that "peer-to-peer behavior" includes at least that each node can communicate tasks and data with other nodes without the tasks and data being required to go through a central server or master node. Ex. 2001 ¶ 53.

Accordingly, the Board should construe "cluster node module" to mean "a module that cooperates with other cluster node modules to establish intercommunication among nodes in a computer cluster and to exchange messages such that each node can communicate tasks and data with other nodes without the tasks and data being required to go through a central server or master node."[6] Ex. 2001 ¶ 54.

[6] A POSITA would recognize there is some overlap between the "cluster node modules" limitation of claim 4 and the "peer-to-peer architecture" limitation of claim 1, but the "cluster node modules" limitation is narrower.

III. THE PETITION DOES NOT ESTABLISH A REASONABLE LIKELIHOOD THAT ANY CHALLENGED CLAIM IS UNPATENTABLE

A. The Petition does not establish that the Distributed Maple Code references are prior art printed publications.

The single ground of the Petition relies on a collection of ***seven*** references that the Petition calls "Distributed Maple Code." Pet. at 14 (the claims "are rendered obvious by Schreiner 1 in view of Schreiner 2, Schreiner 3, the ***Distributed Maple Code***, the Maple Guide, the SPARC IV Article, and the AMD Article."), 8-9 (explaining that Distributed Maple Code includes Exhibits 1012-1018). Thus, the single ground is a ***13***-reference obviousness ground. This ground is fundamentally flawed because Petitioner has not established that the Distributed Maple Code references upon which the ground relies are prior art printed publications. The Board may deny the Petition on that basis alone.

The Petition asserts that the Distributed Maple Code references (Exhibits 1012-1018) are source code and related "ReadMe" and other instructional files for the Distributed Maple software. Pet. at 8-9. Petitioner relies on Schreiner's testimony that he posted the Distributed Maple Code in 2003 on the public website of the Research Institute for Symbolic Computation ("RISC"), where the references allegedly could be downloaded through the webpage shown in Exhibit 1024. Ex. 1006 ¶¶ 30-36. But Schreiner's uncorroborated testimony of his own alleged publication of the claimed invention is not reliable evidence that he posted the Distributed Maple Code, in the form submitted in this IPR, at the relevant time. *See, e.g., Finnigan Corp. v. ITC,* 180 F.3d 1354, 1366 (Fed. Cir. 1999) (corroboration is required of a witness's testimony about his own allegedly invalidating activities).

Petitioner failed to corroborate Schreiner's testimony. Schreiner tacitly acknowledges that Distributed Maple Code has changed over time by testifying that there were several different versions of the source code. Ex. 1006 ¶ 30. Petitioner relies entirely upon Schreiner's memory from more than seven years ago to suggest the version of Distributed Maple Code filed in this IPR was publicly accessible back then. No documentary evidence corroborates Schreiner's recollection. While Exhibit 1008 lists the URL of the RISC website and Exhibit 1024 includes hyperlinks, no evidence shows what content, if any, the hyperlinks pointed to. Significantly, the purportedly archived version of Exhibit 1024 attached to the Rosenberg declaration does not include an archived version of the Distributed Maple Code itself. Thus, there is no corroborated evidence the Distributed Maple Code as filed was, in 2003, publicly accessible from the RISC website or anywhere else.

Moreover, even if there were evidence the Exhibit 1024 hyperlinks pointed to the Distributed Maple Code, no evidence but Schreiner's conclusory assertion shows that persons interested and ordinarily skilled in the relevant subject matter could locate the Distributed Maple Code through the exercise of reasonable diligence.

There is no evidence that any other recognized website in the relevant field linked to the Distributed Maple Code, that anyone discussed the Distributed Maple Code or where to find it at relevant conferences, that advertising or other materials directed anyone to the Distributed Maple Code, that any index or catalog of web content directed anyone to the Distributed Maple Code, or that the Distributed Maple Code would be listed with reasonable prominence in search results for relevant search terms. While Schreiner says his software was "accessible through Google searches" (Ex. 1006 ¶ 21), he has not established that the software would have been listed reasonably prominently in response to a search for the terms that a reasonably diligent searcher interested and ordinarily skilled in the relevant field would have employed. The most the evidence submitted by Petitioner shows is that Schreiner himself, or possibly others who helped create the Distributed Maple Code or already knew of its existence, may have been able to locate whatever version was posted at that time. Because that is insufficient to meet Petitioner's burden to show public accessibility of the Distributed Maple Code, Petitioner cannot rely on those references to meet its burden. And because the single ground relies on the Distributed Maple Code, the Board may deny the Petition on that basis alone.

B. The Petition does not establish claims 1, 4-10, 18-22, 24-25, 30-31, or 33-34 are unpatentable.

1. The Petition does not show that the prior art discloses the "peer-to-peer architecture" limitation.

Claim 1 recites "a mechanism for the nodes to communicate results of mathematical expression evaluation with each other using a ***peer-to-peer architecture***" (emphasis added). Petitioner fails to meet its initial burden with respect to claim 1 and dependent claims 4-10, 18-22, 24-25, 30-31, and 33-34 at least because the Petition does not show that the prior art discloses this "peer-to-peer architecture" limitation. Ex. 2001 ¶ 55.

As explained above in Section II(A)(2), the Board should construe "peer-to-peer architecture" to require at least "an architecture in which each node can communicate tasks and data with other nodes without the tasks and data being required to go through a central server or master node." The alleged prior art does not disclose the "peer-to-peer architecture" limitation at least because it expressly requires at least all tasks to go through a central server or master node. Ex. 2001 ¶ 55.

Specifically, Schreiner 1 discloses: "All remote schedulers send new tasks to the root node scheduler which distributes them among all machines." Ex. 1008 at 316. It further discloses that "[s]ince the root is in charge of task scheduling, the root sees every task created in the session." *Id.* at 323. A POSITA would understand that the "root node" of Schreiner 1 is a "master node." Accordingly, the prior art's requirement that at least tasks go through a master node means that the prior art does not disclose a "peer-to-peer architecture." Ex. 2001 ¶ 56.

The Petition relies on Schreiner 1's disclosure that a node needing "to send a message to one of its peers . . . can thus establish a direct connection for message transfers" and Schreiner 3's nearly identical disclosure that "[w]hen a client wants to send a message to another client through a peer connection, but such a connection has not existed between these two clients yet, the client creates it." Pet. at 29-30. But these disclosures do not negate Schreiner 1's express disclosure that its architecture requires all tasks to go through a master node. A POSITA would understand that the creation of a "peer connection" does not make Schreiner's architecture "peer-to-peer" when Schreiner 1 and Schreiner 3 expressly disallow communication of tasks using a peer-to-peer architecture by requiring all tasks to go through a master node. Ex. 2001 ¶ 57.

Nothing in the prior art even suggests that tasks can be communicated using direct node-to-node communication without requiring the tasks to go through the "single root node." Indeed, Schreiner 1's express disclosure that "[a]ll remote schedulers send new tasks to the root node scheduler which distributes them among all machines" (Ex. 1008 at 316) and "the root is in charge of task scheduling" (*id.* at 323) undermines any possible contention that the disclosed architecture enables direct node-to-node communication of tasks without the tasks being required to go through the "single root node." Ex. 2001 ¶ 58.

Petitioner's arguments about the communication of result messages fail to show that the prior art satisfies the "peer-to-peer architecture" limitation at least because they do not negate the prior art's express teaching that tasks are required to go through a master node. Further, the Petition even fails to show that the prior art communicates result messages without requiring the involvement of a master node. Petitioner attempts to imply there is a link between Schreiner 1's sentence, on page 319, about an "idle kernel" returning "a result message" with the sentence on page 311 that "additional socket connections between remote scheduler instances are created on demand" by quoting those sentences directly adjacent to each other in the Petition. Pet. at 30. But those sentences are neither adjacent nor logically linked in Schreiner 1. The sentences are eight pages apart, in different sections, and Schreiner 1 does not suggest they are related to each other. The page 311 quote is in the "Software architecture" section and relates to the "software architecture" of Figure 1. Ex. 1008 at 311. The page 319 quote, by contrast, is in the "System and execution model" subsection of the "Fault tolerance" section and relates to the "execution model" of Figure 5. *Id.* at 319. Significantly, the "execution model depicted in Fig. 5," as shown below, does ***not*** depict any direct node-to-node communications that do not require data to go through the root node:

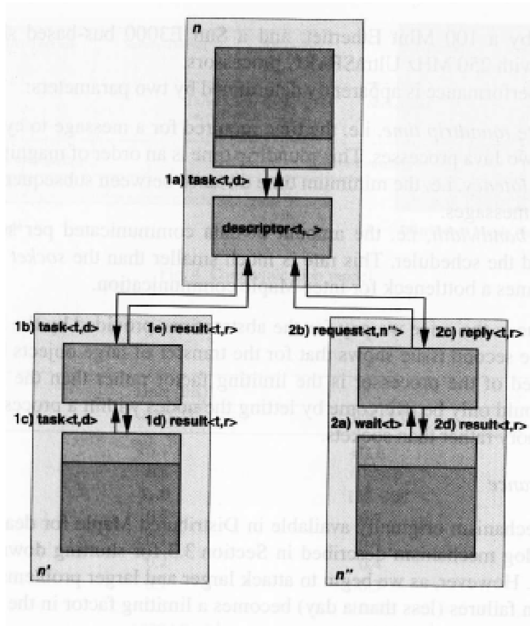


Fig. 5. Execution model.

Id. at 320. Indeed, ***every*** connection depicted on Figure 5 includes the root node. Ex. 2001 ¶ 59.

The Petition also relies on Figure 1 of Schreiner 3, arguing that the "6 result message" box depicts "client n" sending "client 1" a "result message." Pet. at 31. However, the Petition ignores the "7 store message" box, which shows that "client n" sends the same "result" to the "root." Ex. 1010 at 5 ("When the result is computed the root receives it in a *store message* and saves it to the hard disk (see Figure 1)."), 9 ("If a node has computed the result of a task, it sends it in a *result message* to the node which created this task and also sends a duplicate in a *store message* to the root."). Accordingly, Schreiner 3 sends results to a master node. Further, Schreiner 3's communication of results does not overcome the prior art's express disclosure that tasks are required to go through a master node. Ex. 2001 ¶ 60.

Therefore, the Petition does not prove that the prior art discloses a "peer-to-peer architecture" in which "each node can communicate tasks and data with other nodes without the tasks and data being required to go through a central server or master node." Therefore, the Petition fails to meet its initial burden with respect to claim 1 and its dependent claims. Ex. 2001 ¶ 60.

The Petition does not allege that it would have been obvious to modify Schreiner 1 such that the communication of tasks does not require the root node. The Board should neither rewrite the Petition nor allow Petitioner to rewrite the Petition to include such a proposed modification. Moreover, even if the Petition had made such an obviousness allegation, it would not have been obvious to a POSITA to modify Schreiner 1 such that the communication of tasks does not require the root node because that modification would have destroyed the ability of users to use the fault tolerance mechanism disclosed in Schreiner 1. Ex. 2001 ¶ 61.

2. The Petition does not show that the prior art discloses communication of "user instructions" using the "peer-to-peer architecture."

As explained above, the "wherein" clause of claim 1 requires the same "peer-to-peer architecture" used to communicate "results" to also communicate "at least some of the user instructions." The Petition does not show that the prior art meets this limitation. The Petition relies on the prior art's communication of ***tasks*** as allegedly meeting the limitation, but as explained above, the prior art expressly discloses that its communication of tasks relies on requiring the tasks to go through a root node, meaning that it does not use a peer-to-peer architecture. Ex. 2001 ¶ 62.

The Petition specifically alleges that, in the prior art, "the root node accepts . . . Distributed Maple instructions" and then the "root node communicates these instructions using the node-to-node messaging mechanism." Pet. at 49-50 (citing Ex. 1008 (Schreiner 1) at 309-314). The Petition cites five pages for that allegation, failing to identify with particularity any disclosure that the instructions are communicated using a peer-to-peer architecture. *Id.* Upon examination of the cited pages, it is evident that Schreiner 1 never says that those instructions are communicated using a "peer-to-peer architecture" in which the instructions are not required to go through a central server or master node. Ex. 1008 at 309-314. Ex. 2001 ¶ 63.

In fact, the cited pages of Schreiner 1 indicate that instructions are sent as ***tasks***. Ex. 1008 at 309-314. For example, the Petition acknowledges that "the user's instruction dist[start] is communicated to other nodes as ***task*** messages." Pet. at 50 (emphasis added). The Petition further asserts that dist.maple source code confirms that these instructions are communicated as ***task*** messages. *Id.* However, as explained above, Schreiner 1 expressly discloses that tasks, even if viewed as task messages, are distributed by the root node: "All remote schedulers send new tasks to the root node scheduler which distributes them among all machines." Ex. 1008 at 316; *see also id.* at 323 ("the root is in charge of task scheduling"). Accordingly, because the communication of all tasks must go through a master node, the prior art does not disclose using a "peer-to-peer architecture" for such communication. Therefore, Petitioner failed to meet its initial burden with respect to claim 1 and its dependent claims. Ex. 2001 ¶ 64.

The Petition does not allege that it would have been obvious to modify Schreiner 1 such that the distribution of tasks does not require the root node. The Board should neither rewrite the Petition nor allow Petitioner to rewrite the Petition to include such a proposed modification. Moreover, even if the Petition had made such an obviousness allegation, it would not have been obvious to a POSITA to modify Schreiner 1 such that the distribution of tasks does not require the root node because that modification would have destroyed the ability of users to use the fault tolerance mechanism disclosed in Schreiner 1. Ex. 2001 ¶ 65.

3. The Petition does not show that the prior art discloses "wherein the third node comprises a third hardware processor with a plurality of processing cores, wherein the third node is configured to receive the result of the first mathematical expression evaluation from the second node, execute at least a second mathematical expression evaluation using the received result, and communicate the result of the second mathematical expression evaluation to the first node."

This limitation recites a precise order of operations involving three specific nodes: (1) the third node receives, from the second node, a result of a calculation that was performed by the second node in a different claim limitation; (2) the third node performs a different calculation based on the received result; and (3) the third node sends the new result to the first node. The Petition alleges that the alleged prior art publications "disclose this element." Pet. at 43. Ex. 2001 ¶ 66.

Importantly, the Petition does not allege that this limitation would have been obvious; it alleges that the prior art discloses the limitation. The Board may not rewrite the Petition to change the Petition's allegation of actual disclosure to an obviousness allegation. *SAS Inst., Inc. v. Iancu*, 138 S. Ct. 1348, 1356 (2018) ("Nothing suggests the Director enjoys a license to depart from the petition and institute a different inter partes review of his own design.")

The Petition fails to prove its allegation that the prior art discloses this detailed limitation. To prove that allegation, the Petition would need to show where the prior art teaches the precise order of operations, involving three specific nodes, specified by the claim. The Petition does not do that. Instead, the Petition refers to the prior art's general disclosure that multiple nodes may be involved in evaluating mathematical expressions and then speculates that the prior art could work in a manner that would meet the claim limitation. Pet. at 44-47. Ex. 2001 ¶ 67.

The Petition first relies on "the example of Figure 5" of Schreiner 1. *Id.* at 45. It is clear on the face of Petitioner's argument that Figure 5 does not actually disclose every detail about the claim limitation and that Petitioner is merely speculating about how Figure 5 allegedly could work. The Petition asserts that a "POSITA would understand that the third node then ***could*** execute a second mathematical expression using that result, and in fact, that is ***likely why*** the third node requested the result in the first place." *Id.* at 46 (emphases added). The Petition further speculates that "the third node ***could*** send the result of the second mathematical expression to the first (root) node." *Id.* (emphasis added). Ex. 2001 ¶ 68.

The Petition also relies on an "example . . . illustrated in Schreiner3 Fig. 1." *Id.* at 47. But the cited example from Schreiner 3 is about a mechanism for "the ***logging*** of task return values" (Ex. 1010 at 1) and neither Schreiner 3 nor the Petition explains how Figure 1 relates to the evaluation of mathematical expressions by multiple nodes. Accordingly, Petitioner's interpretation of Figure 1 is self-serving and speculative. Moreover, the Petition itself makes clear that Figure 1 does not actually disclose every detail about the claim limitation and that Petitioner is merely speculating about how Figure 1 allegedly could work. The Petition speculates that "once [client 1] receives the result message from node 'client n,' it ***may*** use it to perform at least a second mathematical expression evaluation." Pet. at 47. Petitioner offers no evidence or explanation supporting this speculation. *Id.* Ex. 2001 ¶ 69.

Because the Petition merely speculates about what prior art nodes "could" or "may" do, the Petition fails to prove Petitioner's assertion that the prior art discloses the claim limitation. Ex. 2001 ¶ 70. And because the Petition lacks any allegation that it would have been obvious to modify the prior art to satisfy the claim limitation, Petitioner cannot now fall back on such an obviousness allegation. Petitioner simply failed to meet its initial burden.

C. The Petition does not establish claims 4, 6-10, 30, or 31 are unpatentable.

Claims 4, 6-10, 30, and 31 indirectly depend on claim 1 and, thus, are patentable for the same reasons set forth above that claim 1 is patentable. In addition, these claims are patentable because the prior art does not disclose the limitation of claim 4 reciting that each node "comprises one or more cluster node modules." As explained above, because "cluster node module" is a coined phrase encapsulating the essential features of the cluster node modules disclosed in the specification, the phrase should be construed to mean "a module that cooperates with other cluster node modules to establish intercommunication among nodes in a computer cluster and to exchange messages such that each node can communicate tasks and data with other nodes without the tasks and data being required to go through a central server or master node." These essential features of the cluster node modules are fundamental parts of the claimed invention of claim 4. Ex. 2001 ¶ 71.

The Petition alleges that the "dist.Scheduler and dist.maple components" of the prior art are "cluster node modules." Pet. at 52-55. However, the Petition does not allege or prove that those components allow each node to "communicate tasks and data with other nodes without the tasks and data being required to go through a central server or master node." *Id.* In fact, Schreiner 1 expressly discloses that at least distributing tasks requires a master node: "All remote schedulers send new tasks to the root node scheduler which distributes them among all machines." Ex. 1008 at 316. Schreiner 1 further confirms "the root is in charge of task scheduling" and "the root sees every task created in the session." *Id.* at 323. A POSITA would understand that the "root node" of Schreiner 1 is a "master node." Accordingly, because the purported cluster node modules of the prior art do not "communicate tasks and data with other nodes without the tasks and data being required to go through a central server or master node," they do not satisfy the "cluster node module" limitation as properly construed. Petitioner failed to meet its initial burden with respect to claim 4 and its dependent claims. Ex. 2001 ¶ 72.

The Petition does not allege that it would have been obvious to modify Schreiner 1 such that the distribution of tasks does not require the root node. The Board should neither rewrite the Petition nor allow Petitioner to rewrite the Petition to include such a proposed modification. Moreover, even if the Petition had made such an obviousness allegation, it would not have been obvious to a POSITA to modify Schreiner 1 such that the distribution of tasks does not require the root node because that modification would have destroyed the ability of users to use the fault tolerance mechanism disclosed in Schreiner 1. Ex. 2001 ¶ 73.

D. The Petition improperly relies on alleged public use evidence.

Petitioner also fails to meet its burden because the Petition relies on Wolfgang Schreiner's testimony of an alleged public use of Distributed Maple software to attempt to fill in gaps in the alleged prior art printed publications cited by Petitioner. Schreiner alleges that he and others publicly used and "set up" a "Distributed Maple cluster" such that "each node executed the same software . . . and therefore had the same capabilities." Ex. 1006 ¶ 40. Schreiner's testimony is an improper attempt to fill in gaps in the disclosure of alleged printed publications with testimony about alleged public uses of Distributed Maple by Schreiner and others. Schreiner testifies that "Exhibit 1008 discusses some ***actual cluster configurations we used during the development and testing*** of Distributed Maple." *Id.* ¶ 37 (emphasis added). He says the Abstract "refers to situations where ***we installed*** Distributed Maple." *Id.* ¶ 39 (emphasis added). Accordingly, Schreiner's testimony is, at most, evidence of an alleged public use, not the disclosure of alleged printed publications.

The Petition substantively relies on Schreiner's testimony with respect to several claim limitations to fill in gaps in the disclosure of the alleged printed publications. With respect to the limitation "a plurality of nodes, wherein each of the plurality of nodes comprises a hardware processor," the Petition cites Schreiner's testimony that "each node contained a processor." Ex. 1006 ¶ 47 (cited by Pet. at 23-24). With respect to the limitations "wherein each of the nodes is configured to access a non-transitory computer-readable medium comprising program code" and "configured to access a first memory comprising program code for a user interface and program code for a first single-node kernel," the Petition cites Schreiner's testimony about how "a Distributed Maple cluster was set up." Pet. at 26, 33 (both citing Ex. 1006 ¶ 40). With respect to the limitation "a mechanism for the nodes to communicate results of mathematical expression evaluation with each other using a peer-to-peer architecture," the Petition cites Schreiner's testimony identifying three Java modules he allegedly wrote. Pet. at 28 (citing Ex. 1006 ¶ 14). With respect to the limitation "a plurality of processor cores," the Petition cites Schreiner's testimony that he "installed and operated Distributed Maple in a cluster where one or more of the nodes included multiple processors prior to 2005." Pet. at 40 (citing Ex. 1006 ¶ 61). With respect to the claim 10 limitation "wherein each cluster node module accepts instructions from the user interface and interprets one or more of the instructions," the Petition cites Schreiner's testimony that the "code that implemented" Distributed Maple commands was "written in the Maple language" and "executed in an interpreter environment." Pet. at 59 (citing Ex. 1006 ¶ 52).

Accordingly, with respect to numerous limitations, the Petition relies on Schreiner's testimony about how the Distributed Maple software allegedly created by Schreiner and others operated. This testimony is, at most, evidence of an alleged public use, not the disclosure of alleged printed publications.

35 U.S.C. § 311(b) prohibits Petitioner's reliance on Schreiner's alleged public use of Distributed Maple to fill in gaps in the disclosure of alleged printed publications. Section 311(b) permits an IPR petitioner to challenge a patent "only on a ground that could be raised under section 102 or 103 and ***only on the basis of prior art consisting of patents or printed publications***." 35 U.S.C. § 311(b) (emphasis added). Accordingly, because Schreiner's testimony of alleged public use of Distributed Maple is not a "patent or printed publication," Petitioner cannot properly rely on that testimony to supply claim limitations missing from the printed prior art.

This is not a case, as in *Koninklijke Philips v. Google*, 948 F.3d 1330, 1337 (Fed. Cir. 2020) or under the USPTO's August 18, 2020 guidance, in which the Board may properly rely on evidence outside of patents and printed publications to establish the "general knowledge" of a POSITA. The Petition does not even assert that Schreiner's public use testimony establishes such general knowledge. Instead, the Petition relies on the testimony to recharacterize the printed prior art to incorrectly assert that it teaches missing claim limitations. That use of alleged public use testimony is not permitted by *Koninklijke* or the USPTO's guidance.

That Petitioner found it necessary to attempt to rely on public use testimony—despite the impropriety of such reliance under 35 U.S.C. § 311(b)—shows that the alleged printed prior art, by itself, does not disclose all claim limitations. The Board should reject Petitioner's attempt to go beyond the "patents and printed publications" upon which an IPR petition must be based, and find that Petitioner failed to meet its initial burden to show a reasonable likelihood it would prevail with respect to at least one claim.

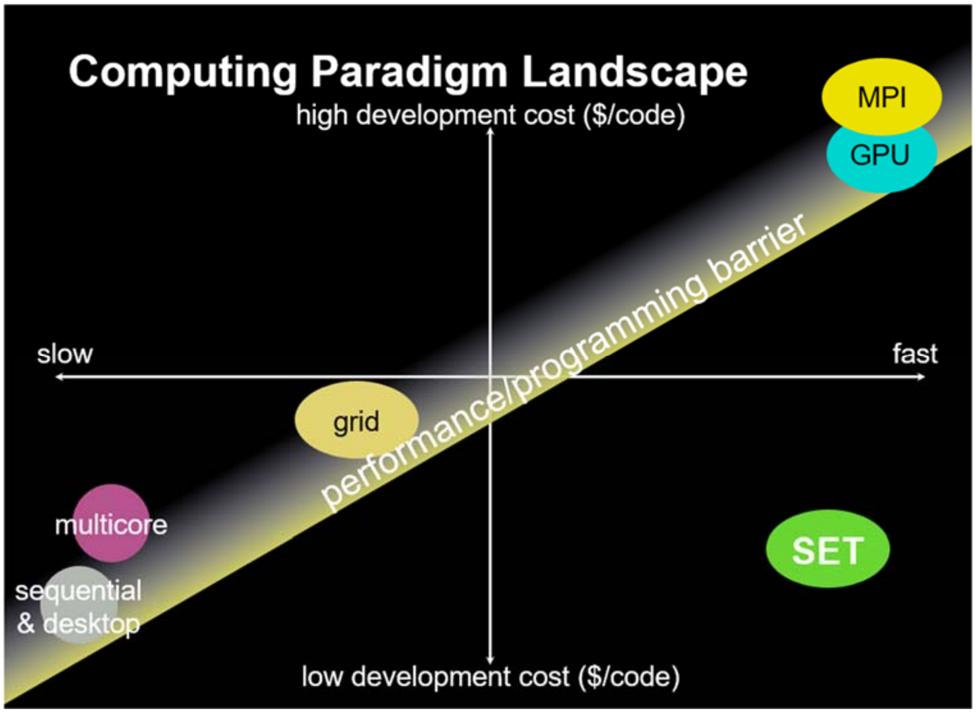
E. Objective evidence confirms the non-obviousness of the challenged claims.

Objective evidence strongly supports the non-obviousness of the challenged claims, including meeting a long-felt, unmet need; the claimed invention's surprising results; failure of others to meet this need; industry praise; initial skepticism; and copying of the claimed invention. *See, generally*, Ex. 2017; Ex. 2018 at 27-28; Ex. 2019 at 2; Ex. 2020. When present, objective evidence of non-obviousness must be considered. *Apple Inc. v. Samsung Elecs. Co.*, 839 F.3d 1034, 1048 (Fed. Cir. 2016), *cert. denied*, 138 S. Ct. 420 (2017).

1. Patent Owner's SEM™ and SET™ products met a long-felt but previously unmet need.

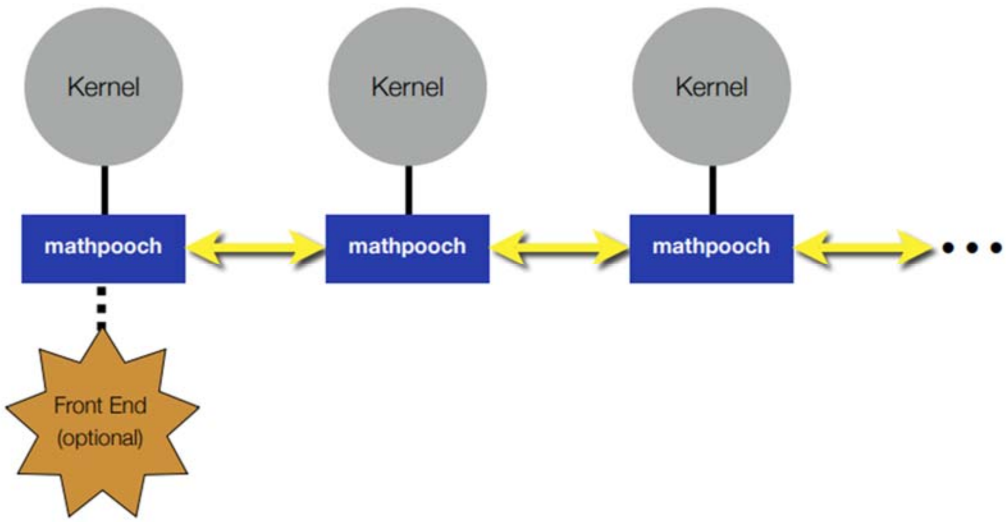
Parallel computation on cluster computers increases the speed at which computations can be performed. Indeed, some particularly complex computations require parallel processing because they simply use too much memory or other computational resources for a single processor to effectively handle. Parallel programming using traditional parallel-computing architectures was notoriously difficult, time-consuming, and expensive. Ex. 2027 ¶¶ 13-18 (citing Exs. 2014, 2024-2025) (collecting quotes); Ex. 2007 ¶ 8; Ex. 2008 ¶ 21. To achieve the highest performance advantages of cluster computing requires asynchronous, peer-to-peer message passing capability between the nodes, such as by using a message-passing paradigm like MPI. Ex. 2027 ¶ 13. This MPI-style parallelized code is the highest performing, but also traditionally the most difficult to implement. *Id.*

Parallel code was typically generated by converting serial code. Ex. 2008 ¶ 25. A programmer would take serial code and then "break" it into pieces for execution on each of the nodes in a parallel computer. *Id.* The programmer then added to each piece the message passing to accomplish inter-nodal communication. Using traditional parallel-computing architectures, only an experienced programmer with specialized parallel programming expertise could develop this type of parallelized code. Ex. 2027 ¶¶ 13-18; Ex. 2007 ¶ 8; Ex. 2008 ¶ 21. And while this process was difficult and time-consuming even for experienced parallel programmers—it was practically impossible for ordinary programmers. Ex. 2027 ¶¶ 13-18; s*ee, generally* Ex. 2017. The relationship between traditional parallel computing performance and the difficulty of the parallel programming is depicted below:

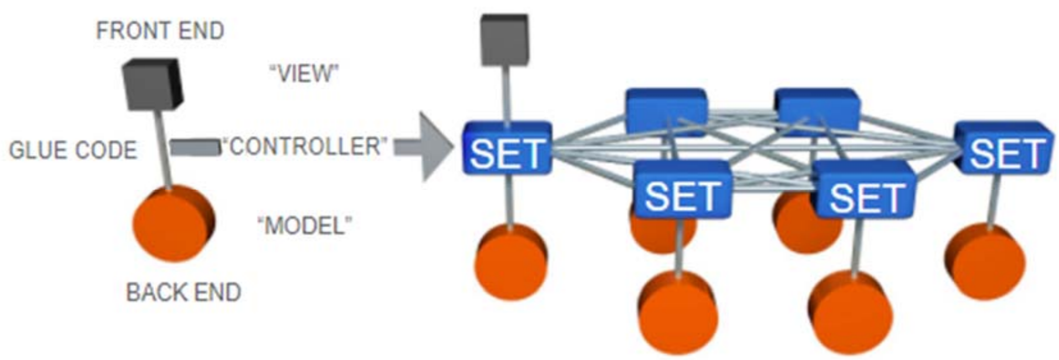


Ex. 2015 at 36. ACS dubbed the inability to achieve results comparable to the highest possible parallel computing performance with a low level of difficulty "the performance/programming barrier." *Id*.; *see also* Ex. 2027 ¶ 19. This posed an "insurmountable barrier for many organizations" that otherwise would have taken advantage of cluster computing. Ex. 2027 ¶ 17 (citing Ex. 2025). Thus, there was a long-felt but unmet need for a way to unlock the performance advantages of cluster computing without requiring specialized expertise or excessive time, effort, and cost (*i.e.*, a need to break the performance/programming barrier). Ex. 2027 ¶¶ 10-19; Ex. 2007 ¶¶ 8, 11; Ex. 2008 ¶¶ 16-23.

Patent Owner developed a cluster-computing architecture, used in its SEM™ and SET™'s products, that met this long-felt, unmet need. Ex. 2027 ¶¶ 20-30. The new architecture interposed a communication layer between the front end user interface and the kernels running on each node of the cluster, or back end. Below is a diagram from one of the first SEM™ manuals that illustrates this architecture:



Ex. 2011 at 4. Below is a diagram from one SET™ datasheet that illustrates this architecture:



Ex. 2016 at 1.

SEM™ and SET™'s cluster-computing architecture enables parallel execution of Mathematica and more general applications, respectively, with a high level of performance but without extensive specialized programming expertise, or the excessive investment of time and effort demanded by traditional parallel-computing architectures. Ex. 2027 ¶¶ 20-30; Ex. 2007 ¶¶ 10, 12 (discussing SEM™); Ex. 2008 ¶¶ 25-28 (discussing SET™). The unique SEM™ and SET™ cluster-computing architecture embodied by the challenged claims is the reason that SEM™ and SET™ were able to meet the long-felt but previously unmet need. Ex. 2027 ¶¶ 24, 27-29; *see also* Ex. 2001 ¶¶ 83-88; *infra* Section III(E)(6). This fact weighs heavily in favor of non-obviousness. *WBIP, LLC v. Kohler Co.*, 829 F.3d 1317, 1332 (Fed. Cir. 2016) ("Evidence of a long felt but unresolved need tends to show non-obviousness because it is reasonable to infer that the need would have not persisted had the solution been obvious.").

2. SEM™ and SET™ achieved surprising results.

SEM™ and SET™'s superior performance and ease of use were unexpected. Ex. 2027 ¶¶ 38-41. As discussed above, traditional parallel-computing architectures presented a performance/programming barrier. *See supra* Section III(E)(1). But the SEM™ and SET™ cluster-computing architecture enabled parallel performance approaching the highest speeds possible using traditional parallel-computing architectures in an unexpectedly easy, fast, and inexpensive manner.

As one example, SEM™ performed better than gridMathematica in terms of raw performance using a parallel computing benchmark. SEM™ and gridMathematica exhibited comparable performance when both applications passed small messages. Ex. 2027 ¶ 39 (citing Ex. 2010). SEM™ substantially outperformed gridMathematica for larger messages, even when gridMathematica used more nodes. *Id.* (citing Ex. 2010). This performance advantage was unexpected, but even more surprising was that SEM did not require specialized expertise to obtain the advantage. Ex. 2027 ¶¶ 39, 41; Ex. 2007 ¶¶ 9-10, 12.

Similarly, the application of SET™'s architecture to large, complex applications facilitated parallelization of those applications in far less time and with less effort than would have been required using traditional parallel-computing architectures. For example, it might take a skilled parallel programmer 12 to 18 months to rewrite the serial code for a typical large mainstream application as parallel code using traditional parallel-computing architectures. Ex. 2027 ¶ 40 (citing Ex. 2016 at 3). But using SET™'s architecture, ACS was able to parallelize Wolfram Research's Mathematica in 1 engineer-month, Apple's HD QuickTime Exporter in 1 engineer-month, and Equalis's Scilab in 2.5 engineer-months. *Id.* (citing Ex. 2016 at 3).

That SEM™ and SET™ achieved such surprising results because of their unique architecture embodied by the claims weighs heavily in favor of non-obviousness. *Circuit Check Inc. v. QXQ Inc.*, 795 F.3d 1331, 1337 (Fed. Cir. 2015)

(finding that the fact a skilled artisan would have been surprised supported the jury's finding that objective evidence of non-obviousness existed.)

3. SEM™ and SET™ successfully addressed the need, where others had failed.

SEM™ and SET™ succeeded where others failed. Ex. 2027 ¶¶ 36-37; Ex. 2007 ¶¶ 29-31; *see also* Ex. 2007 ¶ 11. Before SEM™ and SET™, no one had succeeded at breaking the performance/programming barrier. *See supra* Section III(E)(1). Others had tried to do so by developing automatic parallelizers or universal compilers that would take serial object code as input and output object parallel code. Ex. 2027 ¶ 37; Ex. 2008 ¶ 30. However, none of these attempts achieved performance comparable to what could be achieved using traditional parallel-computing architectures. Ex. 2027 ¶¶ 37, 53 (citing Ex. 2021 at 2); Ex. 2008 ¶¶ 30. This strongly supports non-obviousness. *Uniroyal, Inc. v. Rudkin-Wiley Corp.*, 837 F.2d 1044, 1054 (Fed. Cir. 1988) ("[F]ailure of others to provide a feasible solution to a long standing problem is probative of nonobviousness.").

4. SEM™ and SET™ received industry praise.

Those familiar with SET™ and SEM™ praised them. Ex. 2027 ¶¶ 42-47. For example, PIMCO's Managing Director of Quantitative Modeling and Risk Analytics, Vineer Bhansali, Ph.D., stated that he "found SEM[TM] to be ***very efficient*** in terms of ***stability*** and use for financial engineers who do not have the time to optimize load balancing issues but want to focus on modeling." Ex. 2007 ¶ 12; Ex. 2027 ¶¶ 42-43 (citing Ex. 2018 at 3) (emphasis added).

Former professor at the Tokyo Institute of Technology, Yuko Matsuda, stated "I CAN endorse SEM[TM] with pride. I would emphasize the ***flexibility*** and ***scalability*** of SEM[TM]. And SEM[TM] can make writing Mathematica programs even more ***flexibl***e than before." Ex. 2027 ¶ 44 (citing Ex. 2018 at 4) (emphasis added). Professor Matsuda went on to write a white paper discussing SEM™, in which he praised SEM™ stating, "[f]or Mathematica users, SEM[TM] stands in an advantageous position compared to the Parallel Computing Toolkit (does not use MPI and adopts a Wait/Queue protocol) by Wolfram." *Id.* ¶ 45 (citing Ex. 2023 at 2).

This praise supports non-obviousness. *WBIP*, 829 F.3d at 1334 (industry praise of "a product which embodies the patent claims weighs against an assertion that the same claim would have been obvious"); *see also Institut Pasteur & Universite Pierre Et Marie Curie v. Focarino*, 738 F.3d 1337, 1347 (Fed. Cir. 2013).

5. SEM™ and SET™ were met by initial skepticism.

Because so many had tried but failed to break the performance/programming barrier, many were skeptical that the SEM™ architecture could truly enable the high performance of cluster computing without requiring specialized expertise or excessive time and effort. Ex. 2007 ¶ 8; Ex. 2008 ¶ 32.

Experts were equally skeptical of SET™. For example, the Department of Energy ("DOE") considered three SET™-based grant applications submitted by Patent Owner. The DOE's industry expert reviewers expressed skepticism that SET™ would work. Ex. 2027 ¶¶ 48-54 (citing Exs. 2019, 2021-22). One reviewer stated "[t]he applicant . . . appears to offer an unverified solution to mak[e] parallel computing 'easy'. This reviewer . . . is ***highly skeptical of the results claimed in the proposal***. Without quantitative proof that SET[TM] provides the results, ***I cannot believe that the solution is viable***." Ex. 2027 ¶ 52 (Ex. 2019 at 2) (emphasis added); *see also id.* ¶ 53 (citing Ex. 2019 at 3-4) (misinterpreting SET™ to be an auto parallelizer).

This initial skepticism highlights the non-obviousness of the claimed invention. *WBIP*, 829 F.3d at 1335-36 ("If industry participants or skilled artisans are skeptical about whether or how a problem could be solved or the workability of the claimed solution, it favors non-obviousness."). The initial skepticism was entirely unwarranted given the technical success of SEM™ and SET™.

6. SEM™ and SET™ embody the challenged claims.

There is sufficient nexus between the objective evidence related to SEM™ and SET™ and the challenged claims to accord substantial weight to the objective evidence of non-obviousness. Patent Owner's SEM™ and SET™ products practice at least the challenged independent claims. *See generally* Exs. 2009-2016. Dr. Singh and Dr. Dauger's testimony provide a more detailed analysis of how SEM™ and SET™ practice the challenged claims. Ex. 2001 ¶¶ 74-79; Ex. 2028; Ex. 2029; *see also* Ex. 2027 ¶¶ 31-35.

For example, with respect to claim 1, Dr. Singh explains that the contemporaneous documents show that a cluster running SEM™ or SET™ embodies the fundamental claimed architectural improvements over the prior art. The following figure depicts aspects of a SET™ cluster.

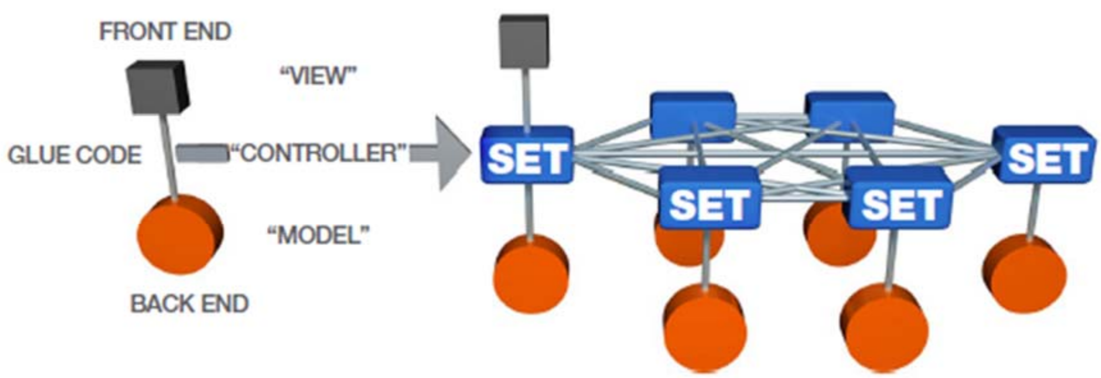


Figure 1. MVC (Left) and how SET is integrated into MVC (Right)

Ex. 2016 at 1. Dr. Singh explains that the figure and accompanying text of the documents show that the blue "SET" rectangles are "a mechanism for the nodes to communicate . . . using a peer-to-peer architecture," the gray "FRONT END" square is a "user interface," and the orange "BACK END" circles are "kernels." Ex. 2001 ¶ 77; Ex. 2028 at 3-5.

The documents also show that the blue "SET" rectangles satisfy the "peer-to-peer architecture" construction set forth above. Every "SET" block is connected to every other "SET" block such that the nodes cooperate to establish intercommunication between the nodes and use MPI to "handle[] low-level message passing internally." Ex. 2013 at 2; Figure 1. Further, the "SET" blocks handle "MPI low-level tasks such as communication and synchronization between processes, data partitioning and distribution, mapping of processes onto processors, and input/output of data structures" and provide "high-level calls for common parallel tasks as Collective communication, Divide and conquer execution and data generation, guard-cell management, element management, and more." Ex. 2016 at 1. There is no indication in the SET documents that tasks and data are required to go through a central server or master node. Dr. Singh concluded that each of the "SET" blocks "can communicate tasks and data with other nodes without the tasks and data being required to go through a central server or master node," and, thus, includes the "peer-to-peer architecture" required by the claim. Ex. 2001 ¶¶ 77, 79; Ex. 2029 (in particular at 3-5); *see also* Ex. 2027 ¶¶ 26, 33-35. A similar analysis shows that SEM™ also embodies the challenged independent claim. Ex. 2001 ¶¶ 76, 79; Ex. 2027.

Dr. Singh also explains that the documents show that a cluster running SEM™ or SET™ embodies all claim limitations of the challenged independent claim, including:

- A computer cluster with a plurality of nodes (*e.g.*, nodes of SEM or SET cluster);
- each of the plurality of nodes comprises a hardware processor;
- one or more of the nodes are configured to receive a command to start a cluster initialization process for the computer cluster (*e.g.*, SETInitialize);
- each of the nodes is configured to access a non-transitory computer-readable medium (*e.g.*, memory) comprising program code for a single-node kernel that, when executed, is capable of causing the hardware processor to evaluate mathematical expressions (*e.g.*, SEM's Mathematica kernels or SET's parallelized application back end code);
- a mechanism for the nodes to communicate results of mathematical expression evaluation with each other using a peer-to-peer architecture (*e.g.*, SEM module or SET module);
- a first node comprising a first hardware processor configured to access a first memory comprising program code for a user interface (*e.g.*, a front end) and program code for a first single-node kernel, the first single-node kernel configured to interpret user instructions and distribute calls to at least one of a plurality of other nodes for execution (*e.g.*, SEM's Mathematica kernels or SET's parallelized application back end code);

- a second node comprising a second hardware processor with a plurality of processing cores (*e.g.*, a multicore processor), wherein the second node is configured to receive calls from the first node, execute at least a first mathematical expression evaluation, and communicate a result of the first mathematical expression evaluation to a third node (*e.g.*, when executing a parallel Fourier transform);
- a third node comprising a third hardware processor with a plurality of processing cores cores (*e.g.*, a multicore processor), wherein the third node is configured to receive the result of the first mathematical expression evaluation from the second node, execute at least a second mathematical expression evaluation using the received result, and communicate the result of the second mathematical expression evaluation to the first node (*e.g.*, when executing a parallel Fourier transform);
- the first node is configured to return the result of the second mathematical expression evaluation to the user interface (*e.g.*, when the parallel fourier transform is complete);
- one or more of the nodes are configured to: accept user instructions; after accepting user instructions (e.g., from the front end), communicate at least some of the user instructions using the mechanism for the nodes to communicate with each other (*e.g.*, SEM module or SET module); and after

communicating at least some of the user instructions using the mechanism, communicate at least some of the user instructions to one or more single-node kernels (*e.g.*, SEM's Mathematica kernels or SET's parallelized application back end code).

Ex. 2001 ¶¶ 74-79; Ex. 2028; Ex. 2029; *see also* Ex. 2027 ¶¶ 31-35.

Each of the objective indicia of non-obviousness results from the SEM™ and SET™ architecture embodied by the challenged claims. SEM™ and SET™'s claimed architecture allowed those products to meet the long-felt but previously unmet need by breaking the performance/programming barrier and enabling high-performance cluster computing without the specialized expertise or excessive time or effort required by traditional parallel-computing architectures. Ex. 2027 ¶¶ 21-31. Other parallel-computing architectures failed to meet the long-felt, unmet need precisely because they lacked SEM™ and SET™'s claimed architecture. *Id.* ¶ 37. The exceptional performance and other surprising results achieved by SEM™ and SET™ were also due to the claimed architecture. *Id.* ¶¶ 39-42. SEM™ and SET™'s claimed architecture is also directly responsible for the industry praise for the products and the products' ability to prove initial skepticism wrong. *Id.* ¶¶ 47-48.

7. There is evidence that Petitioner copied the claimed invention.

At a meeting conducted under a non-disclosure agreement in November 2012, Patent Owner explained to Petitioner the unique cluster-computing architecture of the claimed invention embodied in the SET™ product. *Id.* ¶ 56. After the meeting, Patent Owner sent Petitioner, in confidence, an email attaching a specially tailored data sheet showing how Patent Owner's patented technology could complement Petitioner's general-purpose GPUs ("GPGPUs") and provide a communications infrastructure for direct all-to-all communications between each GPU. *Id.* ¶¶ 57-59. The datasheet included the following figure showing how SET™'s architecture could complement Petitioner's Tesla GPUs.

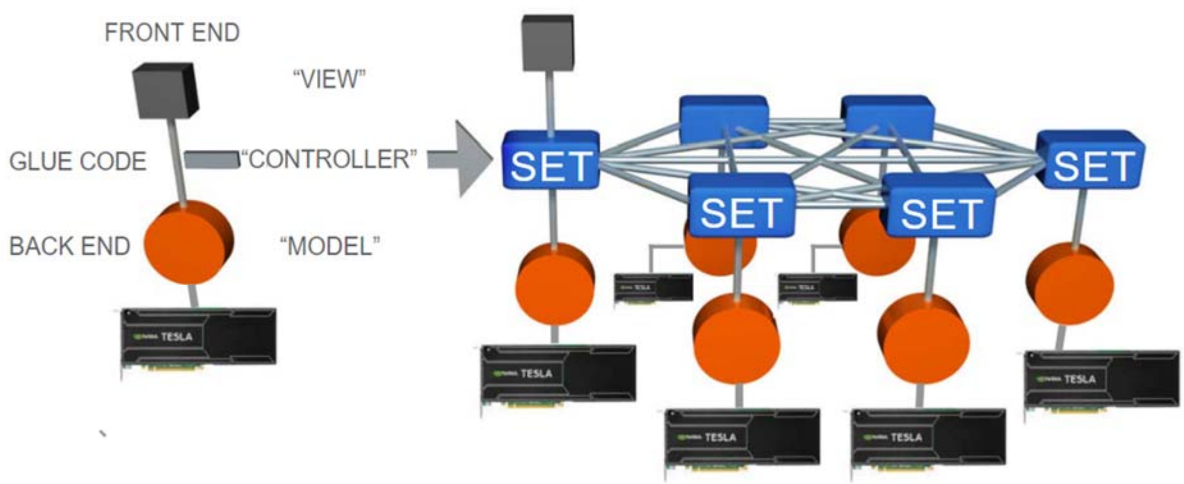


Figure 3. SET is a Perfect Companion to GPGPUs

Ex. 2020 at 4. The datasheet was marked with U.S. Patent Nos. 8,082,289 and 8,140,612. Ex. 2027 ¶ 58; Ex. 2020 at 4.

Petitioner then copied the claimed invention by incorporating the claimed architecture into Petitioner's GPGPUs in the manner described by the datasheet. Petitioner eventually named its GPU interconnect architecture, which uses the claimed architecture, NVLink™. Petitioner then released a new version of CUDA[7] that, for the first time, added support for NVLink™. Ex. 2027 ¶ 59.

Accordingly, Petitioner's copying of the claimed invention weighs heavily in favor of non-obviousness. *Windsurfing Int'l, Inc. v. AMF, Inc.*, 782 F.2d 995, 1000 (Fed. Cir. 1986) ("[C]opying the claimed invention, rather than one within the public domain, is indicative of non-obviousness.").

F. The Board should deny the Petition on the merits.

For the foregoing reasons, Petitioner failed to meet its initial burden with respect to every challenged claim. Therefore, the Board should deny the Petition on the merits because Petitioner did not show "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).

IV. THE PETITION DOES NOT COMPLY WITH THE IPR STATUTE OR REGULATIONS

A. The Petition does not set forth an adequate claim construction.

35 U.S.C. § 312(a)(4) provides that an IPR petition "may be considered ***only if***—"the petition provides such other information as the Director may require by

---

[7] CUDA is Petitioner's parallel computing platform and programming model that enables using one or more GPGPUs for general purpose computing.

regulation." (Emphasis added). Pursuant to that statute, the Director has required, by regulation, that "the petition ***must*** set forth . . . [h]ow the challenged claim is to be ***construed***" and "[h]ow the ***construed claim*** is unpatentable." 37 C.F.R. §§ 42.104(b)(3) and 42.104(b)(4) (emphases added).

The Petition does not comply with the requirement to construe the claims because it does not construe ***any*** claim terms, even though construction is necessary to resolve the Petition. The Petition merely says, "Petitioner believes that no express construction of any term is needed to resolve the challenges herein." Pet. at 14. Petitioner's purported belief that no claim construction is needed is incorrect. It is readily apparent from the intrinsic record of the '768 patent ***alone***—without any need to wait for Patent Owner to take any claim construction position—that constructions of at least the following limitations are necessary to determine whether Petitioner meets its initial burden for institution or its ultimate burden to show that the claims are unpatentable:

- "***a mechanism for the nodes to communicate*** results of mathematical expression evaluation ***with each other using a peer-to-peer architecture***" (all challenged claims); and
- "cluster node module" (claim 4).

The statement, in the November 2019 Consolidated Trial Practice Guide, that "a petitioner may include a statement that the claim terms require no express construction," is not a blanket authorization for Petitioner to omit a claim construction, or to require Patent Owner to go first on claim construction, in every case. Petitioner's assertion that no express construction is needed must be reasonable in view of its challenges to the claims.

Here, Petitioner's assertion that no express construction is needed was not reasonable. Petitioner asserts that the prior art discloses the claim 1 limitation requiring communication between the nodes using a "peer-to-peer architecture" without offering any construction of "peer-to-peer architecture." As explained above, a POSITA would understand that proper construction of the "peer-to-peer architecture" limitation requires analysis of the specification to ascertain the functions the architecture must be able to perform without requiring a central server or master node. Accordingly, it is readily apparent that such an analysis of the specification is essential to determine whether the Petition establishes that the prior art discloses the claimed "peer-to-peer architecture." Therefore, because Petitioner did not do such an analysis of the specification to properly construe "peer-to-peer architecture," it failed to identify "[h]ow the challenged claim is to be construed" and "[h]ow the construed claim is unpatentable," as required by 37 C.F.R. §§ 42.104(b)(3) & 42.104(b)(4).

Petitioner also asserts that the prior art discloses the claimed "cluster node modules" of claim 4 and its dependent claims. However, because "cluster node module" is a coined phrase without an ordinary meaning to a POSITA, it is necessary to rely on the specification to ascertain its meaning. *See 3M Innovative Props.*, 725 F.3d at 1321 ("Idiosyncratic language, highly technical terms, or terms coined by the inventor are best understood by reference to the specification.") Petitioner failed to propose any construction for this coined phrase, and, thus, failed to identify "[h]ow the challenged claim is to be construed" and "[h]ow the construed claim is unpatentable," as required by 37 C.F.R. §§ 42.104(b)(3) & 42.104(b)(4).

Because Petitioner failed to provide claim constructions that are necessary to evaluate the Petition, the Petition is statutorily deficient, and should be denied.

B. The Petition improperly relies on an alleged public use that can be adjudicated properly only in district court.

As explained above in Section III(D), the Petition also violates 35 U.S.C. § 311(b) because it relies on Wolfgang Schreiner's testimony of an alleged public use of Distributed Maple software to attempt to fill in gaps in the alleged prior art printed publications cited by Petitioner. Section 311(b) states that an IPR petitioner may challenge a patent claim "only on a ground that could be raised under section 102 or 103 and ***only on the basis of prior art consisting of patents or printed publications***." Here, the "prior art consisting of patents or printed publications" lacks disclosure of essential claim limitations necessary to Petitioner's *prima facie* case, so the Petition expressly relies on the testimony of a fact witness about an alleged public use, which is not “prior art consisting of patents or printed publications,” to fill in the omitted limitations. Moreover, as explained above in Section III(D), this is not a case, like *Koninklijke Philips v. Google*, 948 F.3d 1330, 1337 (Fed. Cir. 2020) or under the USPTO’s August 18, 2020 guidance, in which the Board may properly rely on evidence outside of patents and printed publications to establish the “general knowledge” of a POSITA. Accordingly, the Petition is deficient under 35 U.S.C. § 311(b), and should be denied.

Assessing the propriety of the Petition’s reliance on an alleged public use of Distributed Maple software should not involve a comparison of how much the Petition relies on statutorily permissible prior art printed publications with how much the Petition relies on a statutorily impermissible alleged public use. Any ground that relies on an alleged public use of Distributed Maple software to any degree is a statutorily impermissible ground that the Board should deny under 35 U.S.C. § 311(b). Further, the Board should not ignore Petitioner’s improper reliance on an alleged public use, thereby modifying the grounds upon which Petitioner chose to rely. *SAS Inst.*, 138 S. Ct. at 1356. Moreover, if the Board finds that some grounds of the Petition as written improperly rely on an alleged public use, but other grounds do not, the Board may not cure this defect by instituting the statutorily proper grounds but denying the statutorily improper ones. *Id.* at 1355 (the IPR statute “indicates a binary choice—either institute review or don’t.”) Under *SAS*, the only proper remedy for a Petition that includes ***any*** statutorily improper ground—as this Petition does—is to deny the whole Petition. *Id.*

Finally, even if the Board finds the Petition is not technically deficient under 35 U.S.C. § 311(b) because of its improper reliance on public use testimony, the Board has discretion to deny the Petition to avoid the injustice of allowing Petitioner to have two bites at the apple: (1) this challenge to the ’768 patent in this IPR and (2) a later public use challenge to the same patent in district court, relying, at least in part, on the same documents and testimony presented here. If this IPR is instituted and a final written decision issues, Patent Owner expects to ask the District of Delaware to apply estoppel to bar such a public use challenge based on Distributed Maple. However, district courts have not consistently applied estoppel to bar public use challenges that relied, at least in part, on alleged printed prior art raised in an IPR. *See, e.g., Wasica Fin. GmbH v. Schrader Int’l, Inc.,* 432 F. Supp. 3d 448, 454 (D. Del. 2020) (estoppel barred validity challenge based on a prior art system when printed prior art presented in an IPR was materially identical to the system); *SPEX Techs. Inc v. Kingston Tech. Corp.,* 2020 WL 4342254, at *15 (C.D. Cal. June 16, 2020) (estoppel does not bar a validity challenge based on a prior art system unless the challenge is a “patent or printed publication theory in disguise”). Thus, it is uncertain whether the court would apply estoppel to bar Petitioner from re-litigating essentially the same invalidity challenge in the form of a public use challenge.

When, as here, a patentability challenge presented to the Board is essentially equivalent to a public use invalidity defense that can only be adjudicated in district court, the Board should defer to the district court, by denying the Petition, so that Petitioner can challenge the patent only once on essentially the same ground. Congress expressly excluded public use invalidity defenses from IPRs, leaving district courts with exclusive jurisdiction to adjudicate such defenses in the first instance. Here, Petitioner's challenge is, in essence, a public use defense: specifically, that Wolfgang Schreiner allegedly programmed and publicly used a software application called Distributed Maple, which allegedly had features satisfying or making obvious the challenged claim limitations. While Petitioner cites printed documents that allegedly disclose some of the features of Distributed Maple, it relies on Schreiner's testimony about additional features that Distributed Maple allegedly had, but which are not described in the printed documents. Congress chose district court, not the Board, as the appropriate venue to adjudicate public use challenges including testimony from alleged creators of public use software applications.

Petitioner's stipulation that it will not pursue the specific grounds identified in the Petition in the parallel district court proceeding does not prevent the injustice of allowing Petitioner to have one bite at the apple in this IPR and a second bite based on essentially the same ground in district court. Indeed, Petitioner's stipulation is illusory because Petitioner can argue (incorrectly) to the Board that the Petition challenges the claims based on prior art patents and printed publications and later argue to the district court that an invalidity challenge based on Schreiner's alleged public use of Distributed Maple is different from the specific grounds identified in the Petition. The Board should foreclose this unjust result, while preserving Petitioner's right to present its invalidity arguments based on Distributed Maple ***once***, in district court, by exercising its discretion to deny the Petition.[8]

C. The Petition exceeds the word limit.

37 C.F.R. § 42.24(a)(i) limits the Petition to 14,000 words. The Petition certifies its word count is 13,953. Pet. at 79. But the Petition used tricks to undercount words, including using non-standard formats such as "EX-1001,"

---

[8] This basis for discretionary denial does not implicate the *Fintiv* factors, which relate only to arguments "for discretionary denial under *NHK* ***due to an earlier trial date***" in a parallel district court proceeding. *Apple Inc. v. Fintiv, Inc.*, IPR2020-00019, Paper No. 11 at 5 (P.T.A.B. Mar. 20, 2020) (precedential). Here, the timing of the district court trial is irrelevant because it would be *per se* unjust to allow Petitioner to relitigate essentially the same validity challenge in district court regardless of when the district court trial occurs. The Board should exercise its discretion to deny the Petition to prevent that injustice.

"Schreiner1," and "¶216" and copying text as images on pages 22, 45, and 48 of the Petition. These instances add up to about 600 excess words. The Board has rejected briefs because they used similar formatting tricks to exceed page or word limits. *See, e.g., Starbucks Corp. v. Ameranth, Inc.*, CBM2015-00091, Paper 16 at 2-3 (P.T.A.B. Jan. 29, 2016) (rejecting a Patent Owner Response that used "at least 28 point" spacing rather than "double spacing").

V. CONCLUSION

For the foregoing reasons, the Board should deny the Petition.

Respectfully submitted,

KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: February 10, 2021

By: /Ted M. Cannon/

Jon W. Gurka, Reg. No. 44,139
Ted M. Cannon, Reg. No. 55,036
Cheryl T. Burgess, Reg. No. 55,030

Attorneys for Patent Owner
Advanced Cluster Systems, Inc.

CERTIFICATE OF TYPE-VOLUME LIMITATIONS UNDER 37 C.F.R. § 42.24

Pursuant to 37 C.F.R. § 42.24(d), Counsel for Patent Owner Advanced Cluster Systems, Inc. hereby certifies that this document complies with the type-volume limitation of 37 C.F.R. § 42.24(a)(1)(i). According to Microsoft Word for Office 365 word count, this document contains 11,942 words, including any statement of material facts to be admitted or denied in support, and excluding the table of contents, table of authorities, mandatory notices under § 42.8, exhibit list, certificate of service or word count, or appendix of exhibits or claim listing.

Respectfully submitted,

KNOBBE, MARTENS, OLSON & BEAR, LLP

Dated: February 10, 2021

By: /Ted M. Cannon/
Jon W. Gurka, Reg. No. 44,139
Ted M. Cannon, Reg. No. 55,036
Cheryl T. Burgess, Reg. No. 55,030

Attorneys for Patent Owner
Advanced Cluster Systems, Inc.

CERTIFICATE OF SERVICE

I hereby certify that a true and correct copy of **PATENT OWNER PRELIMINARY RESPONSE** and accompanying **EXHIBITS 2001 – 2002 and 2007 -2029** are being served on February 10, 2021, via email pursuant to 37 C.F.R. § 42.6(e), to counsel for Petitioners as addressed below:

Brent Yamashita (Lead Counsel)
Jonathan Hicks
DLA Piper LLP (US)
2000 University Ave.
East Palo Alto, CA 94303
Phone: 650-833-2348

Email: NVIDIA-ACS-IPR@us.dlapiper.com

Dated: February 10, 2021

By: /Ted M. Cannon/
Jon W. Gurka, Reg. No. 44,139
Ted M. Cannon, Reg. No. 55,036
Cheryl T. Burgess, Reg. No. 55,030

Attorneys for Patent Owner
Advanced Cluster Systems, Inc.

33919109