Trials@uspto.gov
571-272-7822

Paper 9
Date: May 5, 2021

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NVIDIA CORP.,
Petitioner,

v.

ADVANCED CLUSTER SYSTEMS, INC.,
Patent Owner.

IPR2021-00075
Patent 8,140,612 B2

Before KARL D. EASTHOM, ARTHUR M. PESLAK, and SEAN P. O'HANLON, *Administrative Patent Judges*.

EASTHOM, *Administrative Patent Judge*.

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

NVIDIA Corp. ("Petitioner") filed a Petition (Paper 1, "Pet.") requesting an *inter partes* review of claims 1, 4–8, 10, 12, 13, 15, and 29 (the "challenged claims") of U.S. Patent No. 8,140,612 B2 (Ex. 1001, the "'612 patent"). Advanced Cluster Systems, Inc. ("Patent Owner") filed a Preliminary Response (Paper 5, "Prelim. Resp."). Pursuant to the Board's authorization (Paper 6), the parties filed additional briefing. Paper 7 ("Pet. Reply"); Paper 8 ("PO Sur-reply").

The Board has authority to determine whether to institute an *inter partes* review ("IPR"). *See* 35 U.S.C. § 314(b); 37 C.F.R. § 42.4(a) (2020). Under 35 U.S.C. § 314(a), authorization of an *inter partes* review requires the information in the Petition and the Preliminary Response to "show[] that there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition." For the reasons that follow, we institute an *inter partes* review as to the challenged claims of the '612 patent on all grounds of unpatentability presented.

I. BACKGROUND

A. Real Parties-in-Interest

Petitioner identifies itself as the real party-in-interest. Pet. 2. Patent Owner identifies itself as the real party-in-interest. Paper 3, 1.

B. Related Proceedings

The parties indicate that the '612 patent is the subject of *Advanced Cluster Systems, Inc. v. NVIDIA Corp.*, No. 19-cv-02032 (D. Del. filed Oct. 28, 2019). Pet. 3; Paper 3, 1. The parties collectively identify the following *inter partes* review petitions filed against patents related to the '612 patent: IPR2020-01608, IPR2021-00019, IPR2021-00020, and IPR2021-00108. Pet. 3;

Paper 3, 1.

C. The ’612 patent

The ’612 patent, titled “Cluster Computing Support for Application Programs,” “relates to the field of cluster computing generally and to systems and methods for adding cluster computing functionality to a computer program, in particular.” Ex. 1001, code (54), 1:17–20. Embodiments of the ’612 patent include enabling a software package to benefit from a plurality of nodes in a cluster. *Id.* at 2:13–17. For example, “[o]ne embodiment adapts a software module designed to run on a single node, such as, for example, the Mathematica kernel, to support cluster computing, even when the software module is not designed to provide such support.” *Id.* at 2:24–27. “One embodiment provides parallelization for an application program, even if no access to the program's source code is available.” *Id.* at 2:28–30. “One embodiment provides access to [] high-performance computing through a Mathematica Front End, a command line interface, one or more high-level commands, or a programming language such as C or FORTRAN.” *Id.* at 2:20–23.

Figure 1 of the '612 patent follows:

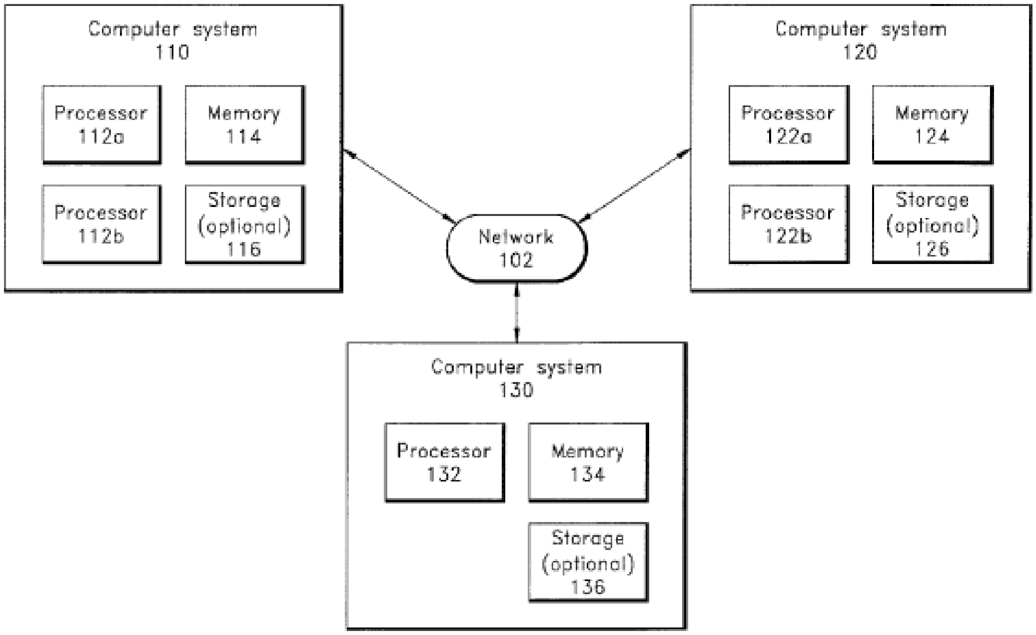

FIG. 1

Figure 1 "is a block diagram of an embodiment of a computer cluster 100 wherein computer systems 110, 120, 130 communicate with one another via a communications network 102." Ex. 1001, 4:63–66. Each computer system includes at least one processor 112a, 112b, 122a, 122b, 132, memory 114, 124, 134, and, optionally, storage 116, 126, 136. *See id.* at 4:66–5:6. Each processor includes an independent processing core, or "node," that is capable of single-threaded execution. *See id.* at 4:45–48, 5:6–11.

Figure 2 of the ’612 patent, showing relationships among software modules in computer cluster 100 (Ex. 1001, 5:16–18), follows:

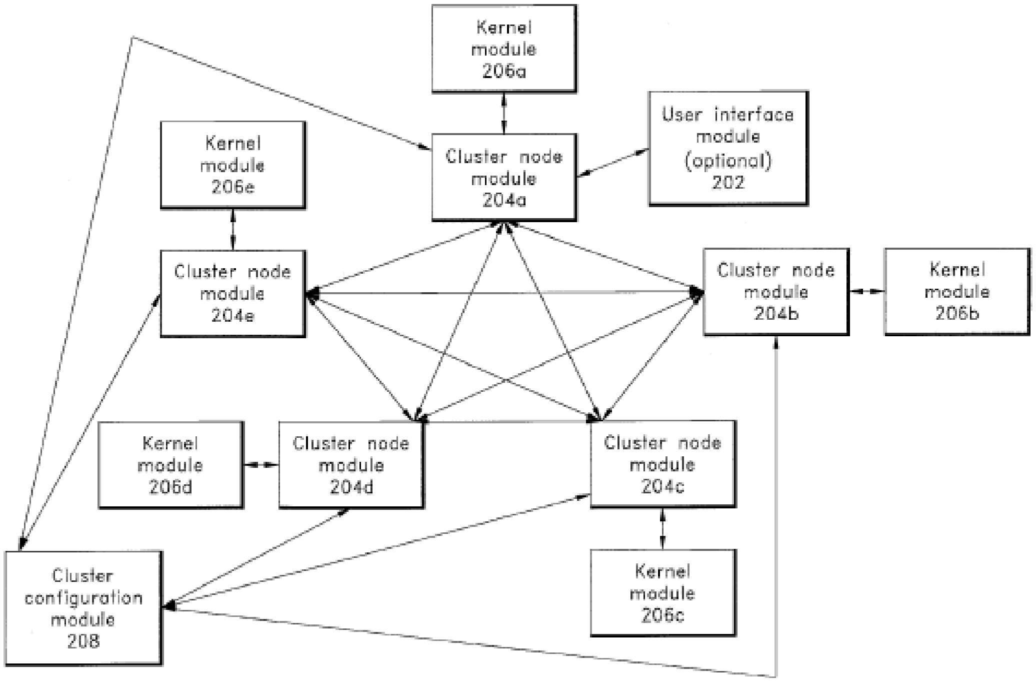

FIG. 2

Figure 2 above illustrates user interface module 202 in communication with cluster node module 204a, which in turn, is in communication with its kernel module 206a, cluster configuration module 208, and other cluster node modules 204b, 204c, 204d, and 204e, with each cluster node module respectively in communication with kernel modules 206b, 206c, 206d, and 206e. *See* Ex. 1001, 5:34–36, 5:40–45.

In the embodiment of Figure 2, each kernel module communicates with a single cluster node module. *See* Ex. 1001, 5:32–37. Cluster node modules are software modules that “can include” at least a portion of the message-passing interface (MPI) application programming interface (API) to interact with an application, such as Mathematica. *See id.* at 11:36–51. In the Figure 2 embodiment, each cluster node module communicates with each of the other cluster node modules. *Id.* at 5:40–47. Also, as indicated above, one of the cluster node modules (module 204a) communicates with user interface module 202. *Id.* at 11:6–15. That cluster node module receives commands from the user interface and submits the commands to all of the other cluster node modules. *Id.* at 23:20–24, 24:37–46. Each cluster node module communicates the command to its respective kernel module, such as for example, Mathematica kernels. *Id.* at 11:46–51, 23:24–26, 24:46–48. Each kernel module processes the command and returns a result to its respective cluster node module. *Id.* at 23:27–31, 24:61–65. The cluster node module can report the result to the other cluster node modules. *Id.* at 23:31–34, 24:65–25:1. This "interact[ion] on a pair-wise or collective basis" allows code running within multiple, simultaneously running kernel modules to perform calculations, processing, or other work on a larger scale and faster than one kernel acting alone. *Id.* at 23:55–63.

An "interpreter," "sometimes called a kernel," "executes instructions provided to the program by a user, a script, or another source" and "can manage at least some hardware resources of a computer system and/or can manage communications between those resources and software (for example, the provided instructions, which can include a high-level programming language)." Ex. 1001, 1:38–44.

D. Illustrative Claim 1

The Petition challenges claims 1, 4–8, 10, 12, 13, 15, and 29. Pet. 1. Claim 1, the sole independent claim, illustrates the challenged claims at issue:

> 1. [a.] A computer cluster comprising:
>
> [b] [i.] a plurality of nodes, wherein each node is configured to access a computer-readable medium comprising [ii.] program code for a single-node kernel module configured to interpret user instructions;
>
> [c.] [i.] a plurality of cluster node modules, wherein [ii] each cluster node module is stored in a computer-readable medium and [iii.] configured to communicate with a single-node kernel and with one or more other cluster node modules, [iv.] to accept instructions from a user interface or a script, and to interpret at least some of the user instructions such that the plurality of cluster node modules communicate with one another in order to act as a cluster, and
>
> [d.] a communications network to connect the nodes;
>
> [e.] wherein the plurality of cluster node modules are configured to cooperate to interpret, translate, or interpret and translate commands for execution by a plurality of single-node kernel modules, and
>
> [f.] wherein at least one of the plurality of cluster node modules returns a result to the user interface.

Ex. 1001, 28:23–43 (bracketed information added by Board to conform to Petitioner's nomenclature).

E. The Asserted Challenges to Patentability

The Petition relies on the following references:

Wolfgang Schreiner et al., *Distributed Maple: Parallel Computer Algebra in Networked Environments*, 35 Journal of Symbolic Computation 305 (2003) (Ex. 1008) ("Schreiner1");

Wolfgang Schreiner, *Distributed Maple – User and Reference Manual (V 1.1.12)* (2001) (Ex. 1009) ("Schreiner2");

Károly Bósa and Wolfgang Schreiner, *Task Logging, Rescheduling and Peer Checking in Distributed Maple* (2002) (Ex. 1010) ("Schreiner3");

K. M. Heal et al., *Maple V Learning Guide* (J. S. Devitt ed., 1998) (Ex. 1011) ("Maple Guide");

"Distributed Maple Code" references (Pet. xii):

> Dist.Maple5: "[S]ource code for Distributed Maple" (Ex. 1012);
>
> Maple Function Source Code: "Source code for parallel versions of Maple functions in Distributed Maple from the 'distsoft' directory" (Ex. 1013);
>
> CASA Function Source Code: "Source code for parallel versions of CASA functions in Distributed Maple from the 'distsoft' directory" (Ex. 1014);
>
> Install1 File: "'Install' file for Distributed Maple" (Ex. 1015);
>
> ReadMe1 File: "'ReadMe' file for Distributed Maple" (Ex. 1016);
>
> Install2 File: "'Install' file for source code in 'distsoft' directory" (Ex. 1017);
>
> ReadMe2 File: "'ReadMe' file for source code in 'distsoft' directory" (Ex. 1018);

H.M. Deitel, P.J. Deitel, D.R. Choffnes, Operating Systems (3rd Ed. 2004) (Ex. 1019) ("Deitel");

Yu Chen, Qian Fang, Zhihui Du, Sanli Li, "TH-MPI: OS Kernel Integrated Fault Tolerant MPI," Recent Advances in Parallel Virtual Machine and Message Passing Interface, Proc. 8th European PVM/MPI Users' Group Meeting Santorini/Thera, Greece, Sept. 23–26, 2001 ("Chen") (Ex. 1020).

Petitioner refers to Exhibits 1008–1010 and 1012–1018 collectively as "Distributed Maple Publications." Pet. 7–8.

Petitioner asserts the following grounds of unpatentability:

Claim(s) Challenged	35 U.S.C. §	Reference(s)/Basis
1, 6, 10, 12, 13, 15, 29	103(a)[1]	Schreiner1, Schreiner2, Schreiner3, Maple Guide, Distributed Maple Code
4, 5, 7, 8	103(a)	Schreiner1, Schreiner2, Schreiner3, Maple Guide, Distributed Maple Code, Deitel
4, 5, 7, 8	103(a)	Schreiner1, Schreiner2, Schreiner3, Maple Guide, Distributed Maple Code, Deitel, Chen

See Pet. 10, 59, 72. Petitioner submits declarations of Henry Tufo, Ph.D. (Ex. 1005) and Wolfgang Schreiner, Ph.D. (Ex. 1006) in support of its contentions. Patent Owner submits declarations of Jaswinder Pal Singh, Ph.D. (Ex. 2001), Vineer Bhansali, Ph.D. (Ex. 2007), John Bancroft (Ex. 2008), and Dean Dauger, Ph.D. (Ex. 2033), in support of its Preliminary Response.

II. ANALYSIS

Petitioner challenges claims 1, 4–8, 10, 12, 13, 15, and 29 as obvious based on the grounds listed above. Pet. 1. Patent Owner disagrees. Prelim. Resp. 1–8.

[1] The filing date of the application resulting in the ’612 patent precedes the effective date of Leahy-Smith America Invents Act (“AIA”), Pub. L. No. 112–29, 125 Stat. 284 (2011). Thus, reference is to the pre-AIA version of section 103.

A. Legal Standards

35 U.S.C. § 103(a) renders a claim unpatentable if the differences between the claimed subject matter and the prior art are such that the subject matter, as a whole, would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. *See KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007).

Tribunals resolve obviousness based on underlying factual determinations, including (1) the scope and content of the prior art; (2) any differences between the claimed subject matter and the prior art; (3) the level of skill in the art; and (4) where in evidence, so-called secondary considerations. *See Graham v. John Deere Co.*, 383 U.S. 1, 17–18 (1966). Prior art references must be "considered together with the knowledge of one of ordinary skill in the pertinent art." *In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994) (citing *In re Samour*, 571 F.2d 559, 562 (CCPA 1978)).

B. Level of Ordinary Skill in the Art

Petitioner relies on Dr. Tufo, who testifies that a person having ordinary skill in the art at the time of the invention ("POSITA") would have had "a Bachelor's degree in computer science, electrical engineering, or an equivalent field, and two years of academic or industry experience in parallel and/or distributed computing." Ex. 1005 ¶ 40; Pet. 13 (citing Ex. 1005 ¶¶ 38–41).

On one hand, the Preliminary Response does not present a proposed level of ordinary skill at this preliminary stage. On the other hand, Dr. Singh, Patent Owner's declarant, disagrees with Dr. Tofu's proposed level of ordinary skill. Ex. 2001 ¶¶ 33–34.[2] The Preliminary Response does not cite Dr. Singh's proposal. For purposes of this Decision on Institution, we adopt Petitioner's proposed level of ordinary skill in the art, which comports with the teachings of the '612 patent and the asserted prior art.

C. Claim Construction

In an *inter partes* review, the Board construes each claim "in accordance with the ordinary and customary meaning of such claim as understood by one of ordinary skill in the art and the prosecution history pertaining to the patent." 37 C.F.R. § 42.100(b) (2020). Under the same standard applied by district courts, claim terms have their plain and ordinary meaning as would have been understood by a person of ordinary skill in the art at the time of the invention and in the context of the entire patent disclosure. *Phillips v. AWH Corp.*, 415 F.3d 1303, 1313 (Fed. Cir. 2005) (en banc). "There are only two exceptions to this general rule: 1) when a patentee sets out a definition and acts as his own lexicographer, or 2) when the patentee disavows the full scope of a claim term either in the specification or during prosecution." *Thorner v. Sony Comput. Entm't Am. LLC*, 669 F.3d 1362, 1365 (Fed. Cir. 2012).

---

[2] Dr. Singh contends that "'parallel computing' is a category that includes 'cluster computing' but 'distributed computing' is different from both 'cluster computing' and 'parallel computing.'" *Id.* ¶ 33. Accordingly, Dr. Singh testifies that a "POSITA would have had a Bachelor's degree in computer science, electrical engineering, or an equivalent field, and two years of academic or industry experience in ***cluster computing***." *Id.* ¶ 34. However, Dr. Singh states that "my conclusions would not change even assuming that Dr. Tufo's identification of the level of ordinary skill in the art were correct." *Id.* For purposes of institution, Dr. Singh's testimony reveals that any differences in the proposed level of ordinary skill raised by Dr. Singh are not material.

Claim 1 recites the following "cluster node module" limitation, which raises the sole claim construction under dispute here:

> a plurality of cluster node modules, wherein each cluster node module is stored in a computer readable medium and configured to communicate with a single-node kernel and with one or more other cluster node modules, to accept instructions from a user interface or a script, and to interpret at least some of the user instructions such that the plurality of cluster node modules communicate with one another in order to act as a cluster.

Generally, "Petitioner believes that no express construction of any term is needed to resolve the challenges herein but reserves the right to present express constructions in response to any argument by P[atent] O[wner]." Pet. 13.[3] Notwithstanding Petitioner's statement, Patent Owner argues that "Petitioner . . . 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)." Prelim. Resp. 50. To the contrary, Petitioner sufficiently identifies how it reads the claims onto Schreiner1's system, as discussed below. *See infra* § II.D.6. Also, it is

---

[3] Neither party argues that term "module" as recited in claim 1 (i.e., "cluster node module" or "kernel module") is a means-plus-function term even though "'module' is a well-known nonce word that can operate as a substitute for 'means' in the context of § 112, para. 6." *See Williamson v. Citrix Online, LLC*, 792 F.3d 1339, 1350 (Fed. Cir. 2015) (en banc) ("use of the word 'means' creates a presumption that [35 U.S.C.] § 112, ¶ 6 applies"); *see* 35 U.S.C.] § 112, ¶ 6 ("An element in a claim for a combination may be expressed as a means . . . for performing a specified function without the recital of structure, material, or acts in support thereof, and such claim shall be construed to cover the corresponding structure, material, or acts described in the specification and equivalents thereof."). Because neither party argues that "module" is a means-plus-function term, we decline to construe it as such for institution purposes.

evident from Patent Owner's claim construction arguments (addressed in this section and further below (*id.*)) that Patent Owner disagrees with, and therefore understands, the central claim construction dispute underlying Petitioner's implicit claim construction of "cluster node module."

Specifically, Patent Owner presents two separate requirements for the "cluster node module" as recited in claim 1. First, Patent Owner contends "the claims require the cluster node modules to collectively accept instructions from the user interface *without the instructions first passing through any kernel*. After accepting instructions, the cluster node modules may optionally send the instructions to kernels." Prelim. Resp. 1 (emphasis added).

Second, Patent Owner contends

> 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*."

Prelim. Resp. 15–16 (emphasis added).

Both of Patent Owner's contentions seek to import negative limitations into claim 1. Further, regarding the first requirement, Patent Owner argues claim 1 "excludes communication of instructions from the user interface to a kernel, which then forwards the instructions to a cluster node module." Prelim. Resp. 9. Patent Owner sets forth its argument in the following diagram:

Correct order:	User Interface → Cluster Node Module → (Kernel)
Incorrect order:	User Interface → Kernel → Cluster Node Module

The diagram above represents Patent Owner's claim construction wherein Patent Owner labels as an "[i]ncorrect order" a construction of claim 1 that allows a kernel to pass an instruction from a user interface to a cluster node module. *See* Prelim. Resp. 9 (citing Ex. 2001 ¶ 41).

Contrary to Patent Owner's arguments, the plain language of claim 1 does not preclude instructions from passing from a user interface via an intervening kernel to a cluster node module and does not require any of the other negative limitations argued. Rather, claim 1 recites "each cluster node module is . . . configured to communicate with a single-node kernel and with one or more other cluster node modules, to accept instructions from a user interface or a script, and to interpret at least some of the user instructions such that the plurality of cluster node modules communicate with one another in order to act as a cluster." This language requires "each cluster node module . . . to communicate with a single-node kernel" without specifying the kernel's position relative to a user interface and cluster node module, and also without specifying anything about a central server or master mode. The language also requires "each cluster node module . . . to accept instructions from a user interface or a script, and to interpret at least some of the user instructions." As indicated, this language does not prevent the instructions from passing through a kernel, central server, or master node situated between the user interface or script and each cluster node module. The language of the claim 1 also does not specify the origin of "a script" or preclude the script from coming from a kernel, central server, or master mode.

Contrary to Patent Owner's other arguments, the specification does not limit the claims in the manner argued. *See* Prelim. Resp. 10–11 (citing Ex. 1001, 21:24–26, Fig. 2). Patent Owner relies on Figure 2 and other selected passages to incorporate limitations into claim 1 (*see id.*), but the specification specifically states that "[t]he drawings and the associated descriptions are provided to illustrate embodiments and *not to limit the scope of the disclosure*." Ex. 1001, 3:63–65 (emphasis added). Moreover, following guidance from the Federal Circuit, generally "[w]e do not read limitations from the embodiments in the specification into the claims." *Hill-Rom Servs., Inc. v. Stryker Corp.*, 755 F.3d 1367, 1371 (Fed. Cir. 2014) (citing *Liebel-Flarsheim Co. v. Medrad, Inc.*, 358 F.3d 898, 904 (Fed. Cir. 2004)).

Similarly, the specification describes several embodiments under a "SUMMARY" of the invention section in general terms "[w]ithout limiting the scope of the invention." *See* Ex. 1001, 2:7–8. One passage mimics the broad language of claim 1, which plainly does not provide the negative limitations argued by Patent Owner:

> Each cluster node module is configured to communicate with a single node kernel and with one or more other cluster node modules, to accept instructions from the user interface, and to interpret at least some of the user instructions such that the plurality of cluster node modules communicate with one another in order to act as a cluster.

Ex. 1001, 3:22–28.

This generic passage allows each cluster node module "to accept instructions from the user interface," without requiring it to accept them directly from the user interface. The passage says nothing about precluding passage of a message through a master node, central server, or kernel.

Patent Owner also relies on Figure 2 as "show[ing] that the user interface module 202 is connected to the cluster node module 204a ***only*** and the kernel module 206a is connected to the cluster node module 204a ***only***." Prelim. Resp. 11. Patent Owner similarly contends that "[i]n embodiments that include cluster node modules, there is no connection between the user interface module 202 and the kernel module 206a or any of the other kernel modules." *Id.* at 11–12.

As noted above, however, embodiments in the specification generally do not limit the claim terms. *See Hill-Rom Servs.*, 755 F.3d at 1371. Also, the specification contradicts Patent Owner's arguments on this preliminary record. For example, it states "[a] kernel module 206 typically includes program code for interpreting high-level code, commands, and/or instructions *supplied by a user or a script* into low-level code, such as, for example, machine language or assembly language." Ex. 1001, 21:33–36 (emphasis added). Similarly, the specification specifically states that "[t]he operating system or its components *can connect the front end 202 and kernels 206 to one another in the same manner as a cluster node module 204* would or by a variation of one of the techniques described previously." Ex. 1001, 24:22–26 (emphasis added). In other words, contrary to Patent Owner's arguments, on this preliminary record, the specification supports connecting a user interface directly to any kernel.

In line with the above teachings, the specification also describes user interface module 202 communicating with kernel module 202 for "some embodiments" as follows:

> In *some embodiments*, computer cluster 100 includes a user interface module 202, such as, for example a Mathematica Front End or a command line interface, *that includes program code for a kernel module 206 to provide graphical output, accept graphical input, and provide other methods of user communication that a graphical user interface or a command-line interface provides*. *To support a user interface module 202, the behavior of a cluster node module 204a is altered in some embodiments.* Rather than sending output to and accepting input from the user directly, the user interface module 202 activates the cluster node module 204a to which it is connected and specifies parameters to form a connection, such as a MathLink connection, between the cluster node module 204a and the user interface module 202. The user interface module's activation of the cluster node module 204a can initiate the execution of instructions to activate the remaining cluster node modules 204b–e on the cluster and to complete the sequence to start all kernel modules 206a–e on the cluster. *Packets from the user interface module 202, normally intended for a kernel module 206a, are accepted by the cluster node module 204a as a user command*. Output from the kernel module 206a associated with the cluster node module 204a can be forwarded back to the user interface module 202 for display to a user. Any of the cluster node modules 204a–e can be configured to communicate with a user interface module 202.

Ex. 1001, 21:6–31 (emphasis added).

Patent Owner argues that portions of this passage support its construction. Prelim. Resp. 10 (citing Ex. 1001, 21:24–26). However, in context, the passage, read in its entirety, does not support Patent Owner's narrow claim construction on this preliminary record. Rather, the first emphasized portion as quoted above explicitly describes communication between user interface module 202 and kernel module 206. It also describes "alter[ing]" "the *behavior* of cluster node module 204a . . . [but only] *in some embodiments*" so that the cluster node module "can initiate the execution of instructions to activate the remaining cluster node modules 204b–e on the cluster and to complete the sequence to start all kernel modules 206a–e on the cluster." *See id.* (alterations and emphasis by Board). In other words, to the extent altering this "behavior" involving cluster node modules somehow limits normal behavior with respect to normal kernel communications, it only occurs for "some embodiments"—i.e., a subset of "some embodiments" introduced at the beginning of the passage.

In addition, the passage verifies that packets *normally* pass to kernel module 206a (at least for some contemplated embodiments). *See* Ex. 1001, 21:6–31. Therefore, it is only in a subset of the embodiments that packet messages pass from user interface module 202 first, then through cluster node module 204, and finally to kernel module 206a. *See id.* Accordingly, nothing in this passage limits the claims to a direct connection between a user interface and a cluster node module, by precluding an intervening kernel, master node, or server. In addition, Patent Owner's claim construction attempts to allow some forms of indirect communication between cluster nodes, by only attempting to preclude an intervening "*central* server" or "*master* mode," leaving the claim construction open to interpretation without requisite support from the specification. Similarly, Patent Owner explicitly states that its claim construction does not require a direct connection between a user interface and a cluster node module, because it allows for intervening "devices" or "components." *See* Prelim. Resp. 14 n.4 ("To be clear, this construction does not require instructions to be transmitted directly from the user interface to a cluster node module without passing through other devices, such as routers, switches, or other components.").

In addition, cluster node module 204a appears to act as a "master node" or "central server" when it is connected to the user interface module, because messages from other kernels (nodes) must pass through cluster node module 204a on their way to the kernel associated with cluster node module 204a and/or to the user interface node.[4] *See, e.g.*, Fig. 2, Ex. 1001, 6:17–21 ("Results of evaluations performed by kernel modules 206a-e are communicated back to the first cluster node module 204a via the cluster node modules 204a-e, which communicates them to the user interface module 208."), 11:33–36 ("In one embodiment, the cluster node modules 204a-e provide a way for many kernel modules 206a-e such as, for example, Mathematica kernels, running on a computer cluster 100 to communicate with one another."), 22:48–50 ("The cluster node module creates an illusion that a kernel module is communicating directly with the other kernel modules."). Cluster node module 204a also acts as a "central server," because it instigates connections to the remaining cluster node modules, according to the column 21 passage discussed and reproduced above. It also controls the other cluster node modules in a "procedure to shut down the system." *See id.* at 24:37–53.

The specification also indicates that a load balancing embodiment includes a "root processor" that assigns tasks to each of the cluster nodes.

---

[4] According to the specification, "[t]he term 'node' refers to a processing unit or subunit that is capable of single-threaded execution of code." Ex. 1001, 4:46–48. The specification also describes "computers, microprocessors, and/or processor cores ('nodes')." *Id.* at 1:24–25.

See Ex. 1001, 19:60–20:4. On this preliminary record, this further shows that the claims do not require a cluster node module "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*" as asserted by Patent Owner.

Finally, on this preliminary record, the prosecution history also does not support the negative limitations argued by Patent Owner. *See* Ex. 1002, 1 ("Applicant hereby rescinds and retracts such disclaimer" arising out of "any prior amendments or characterizations of the scope of any claim or referenced art.").

Based on the foregoing discussion, requisite disclaimer, disavowal, or lexicography does not appear to exist on this preliminary record to import Patent Owner's proposed negative limitations into the plain language of independent claim 1. *See Omega Eng'g, Inc. v. Raytek Corp.*, 334 F.3d 1314, 1321–23 (Fed. Cir. 2003) (holding that a claim limitation to "strike the periphery . . . for visibly outlining" an energy zone with a laser does not justify a negative limitation of "not striking the center or interior portion" of the energy zone absent an "express disclaimer or independent lexicography in the written description that would justify adding that negative limitation").

No other terms require an express construction. Only those terms that are in controversy need be construed, and only to the extent necessary to resolve the controversy. *Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017) (citing *Vivid Techs., Inc. v. Am. Sci. & Eng'g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999)).

D. Alleged Obviousness of Claims 1, 6, 10, 12, 13, 15, and 29

Petitioner relies on the combined teachings of Schreiner1, Schreiner2, Schreiner3, Maple Guide, and Distributed Maple Code to allege obviousness of claims 1, 6, 10, 12, 13, 15, and 29. *See* Pet. 14–59.

1. Schreiner1 (Ex. 1008)

Schreiner1, entitled "Distributed Maple: parallel computer algebra in networked environments," bears a copyright date of 2003. Ex. 1008, 3. Schreiner1 is a journal article authored by Dr. Schreiner, Christian Mittermaier, and Karoly Bosa. *Id.* at 3. Schreiner1 "gives a comprehensive overview on the design and the use of 'Distributed Maple,' an environment for parallel computer algebra on multiprocessors and heterogeneous computer clusters." *Id.* at 3. Schreiner1 explains that Distributed Maple was developed on the basis of the computer algebra system Maple (*id.*) and that Distributed Maple is built on top of the Maple kernel and does not require any kernel extensions. *Id.* at 4. Schreiner1 states that Distributed Maple is "so portable that applications can be executed in many different environments" and "so general that it can be applied to schedule tasks of other computer algebra systems (e.g., Mathematica)." *Id.* Schreiner1 describes Distributed Maple as providing "a programming model which is based on functional/logic/dataflow parallelism" that "allows the creation of a large number of implicitly scheduled tasks with automatic resolution of data dependencies and of globally shared data structures with implicit synchronization." *Id.*

Schreiner1 describes using Distributed Maple "to develop the first parallel versions for a number of non-trivial applications from algebraic geometry (parallel curve and surface plotting and parallel neighborhood analysis).” Ex. 1008, 5. Schreiner1 discloses that “[t]he user interacts with Distributed Maple via a conventional Maple frontend (text or graphical).” *Id.* at 7. Schreiner1 explains that “[t]he core of Distributed Maple is a scheduler program which is completely independent and even unaware of Maple” and “can in fact embed and schedule tasks from any kind of computation kernels that implement a specific communication protocol.” *Id.* at 8.

Schreiner1 discloses that a Distributed Maple session comprises two main components: a scheduler and a Maple interface. Ex. 1008, 8. Figure 1 of Schreiner1, reproduced below, depicts these components and corresponding software architecture for a Distributed Maple session. *Id.* at 9.

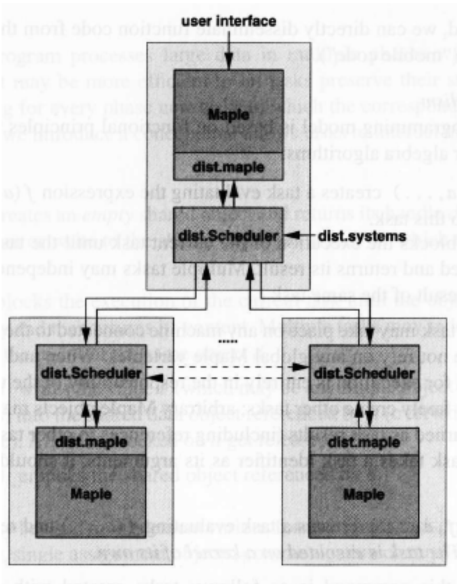

Fig. 1. Software architecture.

As shown in Figure 1, a Distributed Maple session “comprises a set of nodes each of which holds a pair of processes: a *kernel* and a *scheduler*.” Ex. 1008, 17. “Initially, a single task runs on the root kernel; this task may subsequently create new tasks which are distributed via the schedulers to other kernels and may in turn create new tasks." *Id.* With reference to Figure 1, Schreiner1 explains that "every scheduler instance accepts tasks from the attached computation kernel and schedules these tasks among all machines connected to the session." *Id.* at 9. Schreiner1 further explains that "[t]he Maple kernel is a single-threaded process which communicates by a simple communication protocol with the schedule on the same node" and "[a]ll capabilities for parallel and distributed program execution are embedded in this scheduler." *Id.* at 12.

In general, "[t]he system embeds kernels of the computer algebra system Maple as computational engines into a networked coordination layer implemented in the programming language Java." Ex. 1008, 3 (Abstract). Maple is a computer algebra system similar to that of Mathematica. *See id.* at 4. The Distributed Maple system "connects external computation kernels on various machines and schedules concurrent tasks for execution on them." *Id.*

Using "a comparatively high-level programming model, one may write parallel Maple programs that show good speedups in medium-scaled environments." Ex. 1008, 3 (Abstract). Schreiner1 states "[b]oth the Distributed Maple system itself and the library of parallel version of . . . Maple algorithms are in stable versions freely available under the GUN Library General Public License at http://www.risc.uni-linz.ac.at/software/distmaple." *Id.* at 5.

2. Schreiner2 (Ex. 1009)

Schreiner2, titled "Distributed Maple – User and Reference Manual (V 1.1.12)," by Dr. Schreiner, published on July 6, 2001. Ex. 1009, 1.

Shcreiner2 describes the same Maple and Distributed Maple system as Schreiner1 in further detail, including programming and system details. *See, e.g.*, *id.* at 1, 4, 13–19. More particularly, Schreiner2 "describes the use of a system for writing distributed Maple applications and sketches its implementation." *Id.* at 4. Schreiner3 states that its "goal is to provide an environment that makes it easy to implement parallel algorithms in Maple and that can be easily installed in any environment in order to facilitate the distribution of these implementations." *Id.* at 4. As in Schreiner1, Schreiner2 states "[t]he system . . . can be freely downloaded from http://www.risc.uni-linz.ac.at/software/distmaple." *Id.* at 1.

3. Schreiner3 (Ex. 1010)

Schreiner3, titled "Task Logging, Rescheduling and Peer Checking in Distributed Maple," by Dr. Schreiner and others, published on March 18, 2002. Exhibit 1010, 1. Schreiner3 describes the same Maple and Distributed Maple system as Schreiner1 in further detail, including system details and fault analysis. *See id.* at 1–14. Schreiner3 cites Schreiner2. *Id.* at 44 (also citing http://www.risc.uni-linz.ac.at/software/distmaple).

Figure 1 of Schreiner3 illustrates an Execution Model of the Distributed Maple system:

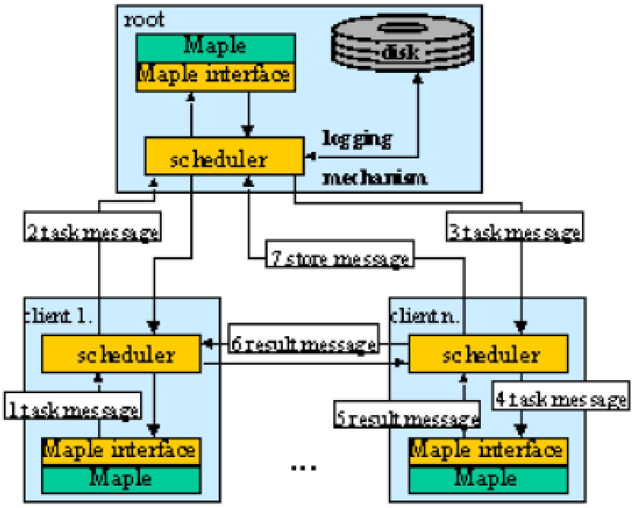

Figure 1: Execution Model

Figure 1 depicts the passing of messages between nodes in a Distributed Maple system, and a logging mechanism in the root node. Ex. 1010, 5.

Schreiner3 explains that "[t]he logging mechanism in Distributed Maple is a fault tolerance mechanism for saving the results of intermediate tasks and the values of shared objects during the computation" and allows the system "to restore the results of computed tasks in a later session, if the current session crashes." Ex. 1010, 3.

4. Maple Guide (Ex. 1011)

Maple Guide, entitled "Maple V Learning Guide, Release 5," and published by Waterloo Maple, Inc., bears a copyright date of 1998. Ex. 1011, 5. Maple Guide explains that "Maple V is a *Symbolic Computation System* or *Computer Algebra System*" and that "[b]oth phrases refer to Maple V's ability to manipulate information in a symbolic or algebraic manner." *Id.* at 11. Maple Guide is an introductory guide that describes how to use Maple's graphical interface and the Maple programming language. *Id.* at 13.

5. Distributed Maple Code (Ex. 1012–Ex. 1018)

Distributed Maple Code is a collection of the following computer code files embodying a distribution of Distributed Maple for installation on a computer: dist.maple5 file (Ex. 1012), source code for parallel versions of Maple functions in the distsoft directory (Ex. 1013), source code for parallel versions of CASA functions in the distsoft directory (Ex. 1014), an "Install" file for Distributed Maple (Ex. 1015), a "ReadMe" file for Distributed Maple (Ex. 1016), an "Install" file for the source code in the distsoft directory (Ex. 1017), and a "ReadMe" file for the source code in the distsoft directory (Ex. 1018).

6. Claims 1, 6, 10, 12, 13, 15, and 29

a. Analysis of Claim 1

Petitioner relies on the combined teachings of Schreiner1, Schreiner2, Schreiner3, Maple Guide, and Distributed Maple Code (source code), as supported by the testimony of Dr. Tufo and Dr. Schreiner, to allege obviousness of claims 1, 6, 10, 12, 13, 15, and 29. *See* Pet. 18–54.

As motivation to combine the "Distributed Maple Publications," Petitioner contends that they share the same author, Dr. Schreiner, and all relate to the same software project, called "Distributed Maple." Pet. 14. Petitioner essentially contends that a person of ordinary skill would have consulted the references to learn details about the system, including fault tolerances and capabilities, in order to combine desired features for running the software modules and system. *See* Pet. 14–16.

Regarding the Maple Guide (not authored by Dr. Schreiner), according to Petitioner, "Schreiner1 teaches that Distributed Maple includes Maple software modules and refers readers to www.maplesoft.com, a website operated by Waterloo Maple, the company that authored and sold the Maple software, for further details." Pet. 15 (citing Ex. 1008, 44; Ex. 1006 ¶ 49).[5] Petitioner explains that

> Waterloo Maple published the Maple Guide, and a POSITA would have been motivated to read the Maple Guide to learn more about Maple. The teaching in the Distributed Maple Publications that Distributed Maple utilized Maple, including its kernel and libraries, provides a POSITA with a strong, express motivation to combine the features described in the Distributed Maple Publications with the features of Maple, as described in the Maple Guide.

Id. at 16 (citing Ex. 1005 ¶ 47).

Petitioner also contends that "the references . . . were publicly available on the same webpage – http://www.risc.unilinz. ac.at/ software/distmaple – which was cited by Schreiner1 and date-stamped and archived by the Internet Archive, and is submitted as Exhibits 1024 and 1025." Pet. 15 (citing Ex. 1008, 5–6; Ex. 1009, Abstract, 4; Ex. 1005 ¶¶ 45–46; Ex. 1006 ¶¶ 24–25; Ex. 1024; Ex. 1025). As noted above, Schreiner1 states that "[b]oth the Distributed Maple system itself and the library of parallel version of . . . Maple algorithms are in stable versions freely

---

[5] Describing Distributed Maple as using "a conventional Maple frontend," Schreiner1 states that "'Maple' is a registered of 'Waterloo Maple Inc.'" Ex. 1008, 7. Schreiner1 also cites http://www.maplesoft.com under a listing of reference sources, listing "Maple, W., Maple 6, 2001" as one such reference source. *Id.* at 44.

available under the GUN Library General Public License at http://www.risc.uni-linz.ac.at/software/distmaple." Ex. 1008, 5.

Petitioner maintains that regardless of the motivation as summarized above, "Schreiner1 expressly teaches nearly all of the claim limitations by itself, and further motivations to combine for specific features are detailed below in connection with particular claim limitations." Pet. 16 (citing Ex. 1005 ¶ 48).

Claim's 1 preamble recites "[a] computer cluster comprising." *See* Pet. 22 (designating the preamble "a."). Petitioner contends that even if the preamble limits the claim, "Schreiner1 discloses it." *Id.* Petitioner relies on Schreiner1's abstract, which states "[w]e describe the design and use of Distributed Maple, an environment for executing **parallel computer algebra programs on multiprocessors and heterogeneous clusters**." *Id.* (Ex. 1008, Abstract). Petitioner also contends Schreiner1's Figure 1 depicts a cluster and Schreiner1 otherwise describes "parallel operations on various clusters." *Id.* (citing Ex. 1008, Fig. 1, 22–42; Ex. 1005 ¶ 60).

Claim 1 also recites the following limitations:

> [b.] "a plurality of nodes, wherein each node is configured to access a computer-readable medium comprising program code for a single-node kernel module configured to interpret user instructions;"
>
> > [i.] "a plurality of nodes, wherein each node is configured to access a computer-readable medium"
> >
> > [ii.] "comprising program code for a single-node kernel module configured to interpret user instructions"

To address these limitations, Petitioner annotates Schreiner1's Figure 1 as follows (Pet. 23):

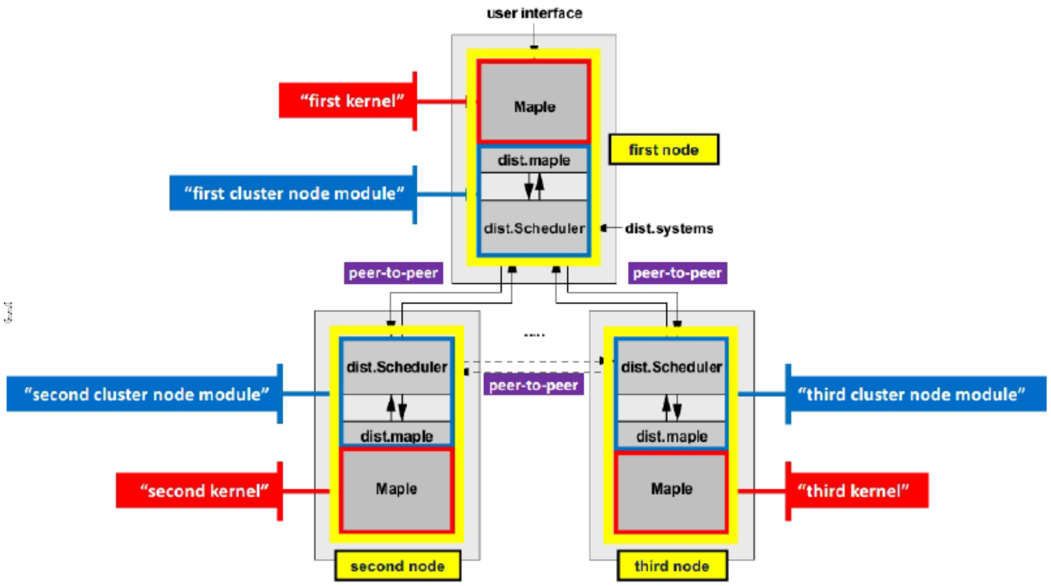

Fig. 1. Software architecture.

According to Petitioner's annotations above in Figure 1, Figure 1 shows Maple kernels (first kernel, second kernel, and third kernel) at each node (first node, second node, and third node). Pet. 23 (citing Ex. 1008, 9; Ex. 1005 ¶ 61). Further addressing limitations b and b.i, Petitioner states that "Maple is installed in storage and loaded in memory when the program is run by one or more processors." *Id.* (citing Ex. 1011, 5, 92). Petitioner also contends that [t]he Maple "kernel consists of highly optimized C code." *Id.* (quoting Ex. 1011, 88). Petitioner also contends that "Distributed Maple" is a software program loaded into memory for execution by one or more processors. *Id.* at 24 (citing Ex. 1008, 5, 22–42; Ex. 1009, 9; Ex. 1015; Ex. 1016; Ex. 1005 ¶ 64). Petitioner also points to "shared memory" as disclosed in Schreiner1 as a computer-readable memory for running Maple and holding the Maple kernels and Distributed Maple code. *Id.* (citing Ex. 1008, 9, 12, 25; Ex. 1009, 9; Ex. 1005 ¶ 65).

Addressing limitation b.ii, Petitioner contends that Schreiner teaches use of Maple as a single-node kernel on each node of the Distributed Maple cluster. Pet. 25 (citing Ex. 1008, 8–9). Petitioner also contends that each Maple kernel interprets user instructions. *Id.* For example, Petitioner contends that that "Distributed Maple 'embeds kernels of the computer algebra system Maple as computational engines' and employs 'a comparatively high-level programming model.'" *Id.* (quoting Ex. 1008, Abstract; citing *id.* at 30; 1005 ¶ 68.).

To further support its showing, Petitioner quotes the Maple Guide:

> The kernel is the base of Maple's system. It contains fundamental and primitive commands: ***the Maple language interpreter (which converts the commands you type into machine instructions your computer processor can understand)***, algorithms for numerical calculation, and routines to display results and perform other input and output operations. . . . The Maple kernel implements the most frequently used routines for integer and rational arithmetic and simple polynomial calculations.

Pet. 25–26 (quoting Ex. 1011, 88) (emphasis by Petitioner). Petitioner also explains that "Schreiner2 teaches that high-level commands are translated by the kernel into lower-level code for execution by the processors." *Id.* at 26 (citing Ex. 1009, 7, 15, 30). According to Petitioner, "Schreiner1 . . . discloses using Distributed Maple with a Mathematica kernel, exactly like in the embodiments of the '612 patent." *Id.* (citing Ex. 1008, 4; Ex. 1005 ¶ 71).

Claim 1 also recites the following limitations:

> [c.] [i.] a plurality of cluster node modules, wherein [ii] each cluster node module is stored in a computer-readable medium and [iii.] configured to communicate with a single-node kernel and with one or more other cluster node modules, [iv.] to accept instructions from a user interface or a script, and to interpret at least some of the user instructions such that the plurality of cluster node modules communicate with one another in order to act as a cluster, and

Generally addressing the cluster node modules and single-node kernel limitations, Petitioner annotates Schreiner1's Figure 1 (as reproduced above) to depict first, second, and third cluster node modules (blue) communicating in a peer-to-peer fashion with each other and communicating with single-node Maple kernels (red). *See* Pet. 28 (annotating Schreiner's Figure 1). Petitioner explains that Schreiner's "Figure 1 depicts two parts of Distributed Maple's [blue] first cluster node module: dist.maple and dist.Scheduler." *Id.* (citing Ex. 1008, Fig. 1; Ex. 1005 ¶ 75).

According to Petitioner, "[t]he dist.Scheduler and dist.maple modules work together [as a (blue) cluster node module] to provide communication capabilities: dist.Scheduler 'coordinates node interaction,' and dist.maple 'implements the interface between the [red] kernel [Maple] and the scheduler.'" Pet. 28 (quoting Ex. 1008, 8). To support its showing, Petitioner quotes the following passage from Schreiner1: "The Maple kernel is a single-threaded process which communicates by a simple communication protocol with the scheduler on the same node. All capabilities for parallel and distributed program execution are embedded in this scheduler." *Id.* (quoting Ex. 1008, 12).

Petitioner also quotes from Schreiner2 to describe the scheduler and Maple interface:

> **Scheduler** A scheduler program manages the node interaction and schedules tasks among nodes. This program is implemented by a Java class library with main class *dist.Scheduler* and is independent of Maple. The initial scheduler process reads all application-specific information from the configuration file *dist.systems* and may then start instances of the scheduler on other machines with which it communicates via sockets.
>
> **Maple Interface** The package *dist.maple* running on each Maple kernel implements the interface between Maple and the scheduler. Communication between both components is based on pipes (named pipes for the Maple kernel connected to the user interface and standard input/output streams for the backend kernels). During a session, additional socket connections between remote scheduler instances are dynamically established on demand.

Pet. 29 (quoting Ex. 1009, 24; citing Ex. 1005 ¶ 76).

Addressing limitation c.ii, Petitioner relies on its showing above with respect to limitation b.i, contending that the "[t]he Distributed Maple software was installed in a storage device and loaded into memory during operation." Pet. 30 (citing Ex. 1005 ¶ 78).

Addressing limitation c.iii, Petitioner explains that the scheduler sends some of the Distributed Maple commands to all kernels. Pet. 31 (citing Ex. 1008, 9–10, 14; Ex. 1009, 15). As an example, Petitioner quotes Schreiner1 as follows: "After a session has been established, every scheduler instance accepts tasks from the attached computation kernel and schedules these tasks among all machines connected to the session." Pet. 31 (quoting Ex. 1008, 9). "All remote schedulers send new tasks to the root node scheduler which distributes them among all machines." *Id.* (quoting Ex. 1008, 14). "[T]he scheduler accepts tasks from the Maple process and schedules these tasks among any node connected to the session." *Id.* (quoting Ex. 1008, 14).

Relying on its annotated version of Schreiner's Figure 1 reproduced above), Petitioner provides evidence that each cluster node module sends commands to other cluster node modules such that they communicate with each other in a peer-to-peer fashion. *See* Pet. 32–35 (citing Ex. 1008, 17–18, Figs. 1 (arrows showing communications between dist.Scheduler), 5; Ex. 1010, 5, Fig. 5).

As one example, Petitioner explains as follows:

> Schreiner1 teaches that the scheduler components of the cluster node modules implement peer-to-peer communication comprising direct message passing between peer nodes: "[A]ll nodes know of each other, i.e. a node knows the address of a machine and the number of a port on which (a thread of) the remote scheduler is listening for connection requests. When a node needs to send a message to one of its peers, it can thus establish a direct connection for message transfers."

Pet. 33 (quoting Ex. 1008, 13; citing Ex. 1008, 17; Ex. 1005 ¶ 86).

As noted above, claim 1 also recites limitation c.iv, "a plurality of cluster node modules . . . configured . . . to accept instructions from the user interface or a script, and to interpret at least some of the user instructions such that the plurality of cluster node modules communicate with one another in order to act as a cluster." In addition to the user interface identified in Schreiner1's Figure 1 (Ex. 1008, 9), Petitioner quotes Schreiner1 as disclosing a user interface. Pet. 37 ("The user interacts with Distributed Maple via a conventional Maple frontend (text or graphical), i.e. she operates within the familiar Maple environment for writing and executing parallel programs." (quoting Ex. 1008, 7–8; citing *id.* at 3–4)).

Regarding the limitation of the cluster node modules "accept[ing] instructions" from the user interface, according to Petitioner, "[t]he root kernel receives instructions from this user interface and passes them to its cluster node module." *Id.* Petitioner explains as follows:

> Fig. 1 . . . shows a command from "user interface" to the root kernel. If the command is a Distributed Maple instruction (e.g., dist[all], dist[start], dist[wait], etc.), the dist.maple component accepts the instruction and passes it to the scheduler component, as shown by the arrow directed from dist.maple to the scheduler component of the root cluster node module.

Pet. 37.

Relying further on Schreiner1, Petitioner provides the example of a "dist[all]" command sent from a user interface and interpreted by cluster node modules such that they act as a cluster:

> The instructions from the user interface are interpreted by the cluster node modules "such that the plurality of cluster node modules communicate with one another in order to act as a cluster," as the claim requires. As an example, Schreiner1 describes a user entering the Distributed Maple "dist[all]" instruction into the user interface: "dist[all](*command*) lets the Maple statement *command* be executed on every Maple kernel connected to the distributed session." EX-1008, [9]; EX-1009, 30. A POSITA understands that, in order for this Distributed Maple instruction and its embedded Maple statement ("*command"*) to be executed on every kernel, the instruction is sent to, and accepted by, every cluster node module, which then interprets it in order to execute it. Once each node evaluates the Maple statement, it communicates the result back to the node that issued the dist[all] instruction. Accordingly, the cluster node modules interpret the dist[all] command such that the nodes communicate to evaluate a Maple statement in parallel, thereby satisfying this claim element. EX-1005, ¶93.

Pet. 38–39.

Petitioner provides another example as disclosed in Scheiner1. *See* Pet. 39–40 (relying on a "dist[start] instruction" entered from a "root node user interface," and "interpreted by the root dist.maple component as a *task*, which the root scheduler can pass to other cluster nodes for execution." (citing Ex. 1008, 8, 10–12, 14)).

Patent Owner does not dispute in a clear fashion that Schreiner1 teaches limitation [c.iv] under our preliminary claim construction set forth above (*supra* § II.C). *See* Prelim. Resp. 27 ("At best, the Petition suggests that the prior art processes instructions in the following order: user interface → kernel → cluster node module."). Rather, Patent Owner relies on its claim construction argument, asserting that claim 1 precludes passing an instruction indirectly from a user interface, via a kernel, to a cluster node module. *See* Prelim. Resp. 27 ("This command processing order [in Schreiner1] is fundamentally different from the order required by claim 1: user interface → cluster node module → (kernels)."). As set forth above, the preliminary record does not support Patent Owner's narrow claim construction that effectively requires a direct connection between the user interface and cluster node modules. *See supra* § II.C.

Patent Owner also argues that in Schreiner1, the root node is a master node and it schedules tasks between other nodes. *See* Prelim. Resp. 29–30. Therefore, according to Patent Owner, Schreiner1 does not disclose a "cluster node module," because under Patent Owner's claim construction, a cluster node module must "communicate tasks and data with other nodes without the tasks and data being required to go through a central server or master node." *See id.*; *see also supra* § II.C. However, as determined above for purposes of institution, the record does not support Patent Owner's claim construction and does not require "communicating . . . without the tasks and data being required to go through a central server or master node." *See supra* § II.C.

Patent Owner also argues that Petitioner's "cited pages never suggest that the user interface shown in Figure 1 of Schreiner 1 sends any instruction to the dist.maple component." Prelim. Resp. 26 (citing Pet. 23; Ex. 2001 ¶ 60). Patent Owner argues that "the user interface communicates with the 'Maple' component—which the Petition calls the 'kernel'—*not* the dist.maple component." *Id.* at 27.

However, as indicated above, Schreiner's Figure 1 shows the user interface connected to the Maple kernel, which connects to dist.maple, and then to dist.Scheduler. *See* Ex. 1008, 9 (Fig. 1). So all communications from the user interface to dist.Scheduler (cluster node module) pass via the Maple kernel and dist.maple to dist.Scheduler. *See id.* Although as noted above, Patent Owner argues that Schreiner1's user interface does not send commands to dist.Scheduler, this appears to be another form of Patent Owner's claim construction argument. *See* Prelim. Resp. 26. In other words, Patent Owner does not argue clearly that dist.Scheduler does not receive commands indirectly from the user interface via a kernel. Rather, as noted above, Patent Owner focuses on its claim construction argument, stating that "[a]t best, the Petition suggests that the prior art processes instructions in the following order: user interface → kernel → cluster node module." *Id.* at 27. This is sufficient for purposes of institution under the claim construction set forth above. *See supra* § II.C.

To the extent Patent Owner argues that the dist.Scheduler does not even receive indirect instructions via a kernel from a user interface, Petitioner sufficiently shows otherwise for purposes of institution. As quoted above, Schreiner1 states that "[t]he user interacts with Distributed Maple via a conventional Maple frontend (text or graphical), i.e. she operates within the familiar Maple environment for writing and executing parallel programs." Pet. 37 (quoting Ex. 1008, 7–8; citing *id.* at 3–4)). Petitioner quotes Schreiner2 to show that the dist.maple component "run[s] on each Maple kernel" and "implements the interface between Maple and the scheduler." *Id.* at 29 (quoting Ex. 1009, 24). The dist.Scheduler "manages the node interaction" and "[t]he initial scheduler process reads all application-specific information." *Id.* (quoting Ex. 1009, 24). Further, "[c]ommunication between both components [i.e., the dist.Scheduler and dist.maple] is based on pipes (named pipes for the Maple kernel connected to the user interface and standard input/output streams for the backend kernels)." *Id.* (quoting Ex. 1009, 24). And "[t]he Maple kernel is a single-threaded process which communicates by a simple communication protocol with the scheduler on the same node. All capabilities for parallel and distributed program execution are embedded in this scheduler." *Id.* at 28 (quoting Ex. 1008, 12).

In light of these cited or quoted teachings in the Petition, Petitioner sufficiently explains for purposes of institution that "the root cluster node module comprising the scheduler and dist.maple components accepts these Distributed Maple instructions directly from the user interface of its Maple kernel (or from a script written in the Maple language) and interprets those instructions." Pet. 36–37 (citing Ex. 1008, 9–12; Ex. 1005 ¶ 90).

Further supporting the Petition, Schreiner1 states "tasks created with dist[start] are scheduled on a fixed number of Maple kernels." Ex. 1008, 10; *see* Pet. 36–37 (citing Ex. 1008, 9–12; Ex. 1005 ¶ 90). Also, "every scheduler instance accepts tasks from the attached computation kernel and schedules these tasks among all machines connected to the session."

Ex. 1008, 9 (describing Schreiner1's Fig. 1). Therefore, Schreiner1's descriptions, including Figure 1, reveal sufficiently for purposes of institution that the cluster node module scheduler at least receives task or application instructions from the user interface, because it reads all application instructions, and it also distributes tasks to multiple Maple kernels based on a dist[start] command from a user interface. *See also* Pet. 28–29 (citing Ex. 1008, 17–18 ("§ 3.5.1: providing an example of message passing between kernels, dist.maple components, and dist.Scheduler components"); Ex. 1009, 24).

The final limitations of claim 1 follow:

> [d.] a communications network to connect the nodes;
>
> [e.] wherein the plurality of cluster node modules are configured to cooperate to interpret, translate, or interpret and translate commands for execution by a plurality of single-node kernel modules, and
>
> [f.] wherein at least one of the plurality of cluster node modules returns a result to the user interface.

For limitation d, Petitioner quotes Schreiner1's disclosure of "networked environments" for the Distributed Maple software. *See* Pet. 42 (quoting Ex. 1008, 3). Petitioner also relies on other teachings in Schreiner1 describing computer nodes with processors connected in a network. *See id.* at 42–43 (citing Ex. 1008, 23, 25; Ex. 1005 ¶ 102).

For limitation e, Petitioner relies partly on its discussion of interpreting and translating Distributed Maple commands in connection with limitations b.ii, c.iii, and c.iv above. *See* Pet. 43–46. According to Petitioner, "the dist.maple and scheduler components interpret dist[all] and dist[start] commands for execution by the plurality of Maple kernels." *Id.* at

43. Petitioner also relies on the following load balancing disclosure for "cooperation" as claimed:

> Moreover, "[a]ll remote schedulers send new tasks to the root node scheduler which distributes them among all machines." EX-1008, [14]. The root scheduler (i.e., cluster node module) uses a load balancing scheme to distribute tasks. *Id.* Under this scheme, "a remote scheduler asks for new tasks whenever the number of received but not yet started tasks falls below a lower bound." *Id.* This is yet another example of cooperation – in order for a task to be executed, the cluster node modules cooperate to establish a load balancing scheme to distribute and interpret or translate task commands for execution. EX-1005, ¶107.

Id. at 46.

For limitation f, Petitioner reproduces "a screenshot of a session in which the user inputs several instructions (highlighted in green) and the cluster node modules then return the result to the user interface (highlighted in orange)." *See* Pet. 47 (citing Ex. 1008, 8; Ex. 1009, 5; Ex. 1005 ¶ 110). Petitioner also relies on a graphical output example for display in Schreiner1's user interface. *See id.* at 48 (reproducing Ex. 1008, Fig. 13).

b. Public Accessibility of the Distributed Maple Code

Patent Owner also generally argues, with respect to all of the claim limitations, that Petitioner failed to show that the Distributed Maple Code references were publically accessible prior to the date of the invention. *See* Prelim. Resp. 22–25 (noting that Distributed Maple Code includes Exhibits 1012–1018). However, Petitioner relies on the Distributed Maple Code references, which describe the source code for the Distributed Maple Code referenced in Schreiner1, Schreiner2, Schreiner3, and the Maple Guide, to *support* its showing based on the latter references. *See, e.g.*, Pet. 39 ("This understanding is further confirmed in the dist.maple source code files."). As noted above, Petitioner also argues "Schreiner1 expressly teaches nearly all of the claim limitations by itself." *Id.* at 16 (citing Ex. 1005 ¶ 48).

Patent Owner does not directly argue that the other references do not support institution sufficiently without the Distributed Maple Code (i.e., source code). In any event, addressing the Distributed Maple Code, Patent Owner contends that Petitioner relies on the uncorroborated testimony of Dr. Schreiner that he posted the Distributed Maple Code "in 2003 on the public website of Research Institute for Symbolic Computation ('RISC'), where the references allegedly could be downloaded through the webpage shown in Exhibit 1024." Prelim. Resp. 22–23 (arguing "corroboration is required of a witness's testimony about his own allegedly invalidating activities" (citing *Finnigan Corp. v. ITC*, 180 F.3d 1354, 1366 (Fed. Cir. 1999)). However, Petitioner also cites the website as published in Schreiner1, as Patent Owner acknowledges. Prelim. Resp. 23; Ex. 1008, 6. Nevertheless, Patent Owner contends that "[t]he most the evidence submitted by Petitioner shows is that [Dr.] 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." *Id.* at 24.

This line of argument downplays that Schreiner1, published in the Journal of Symbolic Computation in 2003, would have pointed interested artisans to the website listed therein in order to obtain the "Distributed Maple system itself," the main subject of Schreiner1 and described as "freely available." *See* Ex. 1008, 5. Also, the Internet Archive screenshot, Exhibit 1024, describes "Distributed Maple" and lists the same website as published in Schreiner1, and describes the website as "[m]aintained by: Wolfgang Schreiner, Last Modification: July 14, 2003." *See also* Ex. 1006

¶ 24 (Dr. Schreiner noting that the Internet Archive screenshot states "Last Modification: July 14, 2003" and testifying "[t]hat is consistent with my recollection of the time when I last modified this page" (citing Ex. 1024)); Ex. 1025 (similar Internet Archive screenshot evidence). This evidence corroborates Dr. Schreiner's testimony as to the timeframe he uploaded the source code to the RISC website.

Dr. Schreiner also testifies that "[i]t has been my practice to check, from time to time, whether my software was accessible through Google search results, and I did this prior to 2005 for these particular web pages and confirm that my Distributed Maple papers and software were accessible through Google searches." Ex. 1006 ¶ 21. Patent Owner argues that "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."[6] Prelim. Resp. 23 (emphasis added). However, as indicated above, Schreiner1, coauthored by Dr. Schreiner with others, and the Internet Archive documents, corroborate Dr. Schreiner's testimony about uploading the software on the RISC website.[7] During trial, Patent Owner will have the opportunity to cross-examine Dr. Schreiner, including regarding Google searches and his memory, assuming for the sake of argument that the ability to search the RISC website using Google is relevant to show public accessibility of the source code.

[6] Given the timeframe of 2003 to 2005 at issue here, Patent Owner probably meant "seventeen years ago" instead of "seven years ago."

[7] As noted above (§ II.D.1), two others co-authored Schreiner1 with Dr. Schreiner.

On this preliminary record, sufficient evidence exists to show that the Distributed Maple Code was publically available in 2003 and thereafter up to the date of the invention in 2006. *See* Ex. 1001, code (60) (listing the filing date of a provisional application as October 11, 2006). Patent Owner does not dispute that some form of the source code existed prior to the date of the invention. To the extent the source code may have changed over the relevant timeframe as Patent Owner argues (*see* Prelim. Resp. 23), the parties will have the opportunity to address the materiality of any such changes as they relate to specific claim limitations or evidence during trial.

Even if the source code was not publically available at the relevant time, Petitioner's reliance on it as extrinsic evidence solely to support its showing of how the Distributed Maple system operated at the time of the invention would be proper. *See In re Baxter Travenol Labs.*, 952 F.2d 388, 390 (Fed. Cir. 1991) (extrinsic evidence may be used to explain what a reference discloses); *Hospira v. Fresenius Kabi USA*, 946 F.3d 1322, 1329 (Fed. Cir. 2020) ("[E]xtrinsic evidence can be used to demonstrate what is 'necessarily present' in a prior art embodiment even if the extrinsic evidence is not itself prior art."). In any event, on this preliminary record for purposes of institution, after considering Petitioner's showing and Patent Owner's arguments and objective evidence of nonobviousness (as discussed further below), Petitioner sufficiently shows that Schreiner1, Schreiner2, Schreiner3, and the Maple Guide teach claim 1 with or without the supporting source code as disclosed in the Distributed Maple Code.[8]

---

[8] Patent Owner also contends that Petitioner's declarant impermissibly relies on "public use" information—i.e., information about commercial embodiments. *See* Prelim. Resp. 37–38; PO Sur-reply 10. Petitioner disagrees. Pet. Reply 9–10. For purposes of institution on this preliminary

c. Alleged Objective Evidence of Nonobviousness

Objective evidence of nonobviousness "may often be the most probative and cogent evidence in the record" and "may often establish that an invention appearing to have been obvious in light of the prior art was not." *Transocean Offshore Deepwater Drilling, Inc. v. Maersk Drilling USA, Inc.*, 699 F.3d 1340, 1349 (Fed. Cir. 2012) (citing *Stratoflex, Inc. v. Aeroquip Corp.*, 713 F.2d 1530, 1538 (Fed. Cir. 1983)).

Patent Owner argues that objective evidence regarding its SEM and SET products supports the nonobviousness of the challenged claims. Prelim. Resp. 33–50. Patent Owner alleges evidence of long-felt and unresolved need, failure by others, praise by others, skepticism of others, and copying. *Id.*

(i.) Nexus

Nexus is a legally and factually sufficient connection between the objective evidence and the claimed invention requiring a tribunal to consider the objective evidence in determining nonobviousness. *Demaco Corp. v. F. Von Langsdorff Licensing Ltd.*, 851 F.2d 1387, 1392 (Fed. Cir. 1988) ("Once a prima facie case of nexus is made the court must consider the evidence adduced on both sides of the question, with such weight as is

---

record, Petitioner sufficiently shows that the cited Exhibits support Dr. Tufo's testimony and at least teach claim 1 without improper reliance on public use information. *See id.* (mapping support for Dr. Tufo's testimony at Ex. 1005 ¶¶ 37–40, 70–76 to Ex. 1008–Ex. 1011). (To provide further support for its position, Petitioner also cites an alleged Exhibit supplied by Patent Owner (i.e., "EX-2005, 2"). *See* Pet. Reply 10. However, this Exhibit does not appear in the record so any argument premised on it provides no added support for Petitioner.)

warranted.”). “The patentee bears the burden of showing that a nexus exists . . . .” *WMS Gaming Inc. v. Int’l Game Tech.*, 184 F.3d 1339, 1359 (Fed. Cir. 1999).

Patent Owner contends that “traditional parallel-computing architectures was notoriously difficult, time-consuming, and expensive.” Prelim. Resp. 34 (citing Ex. 2033 ¶¶ 13–18; Ex. 2007 ¶ 8; Ex. 2008 ¶ 21). Patent Owner contends that programmers typically generated parallel code by breaking up and converting serial code “for execution on each of the nodes of a parallel computer.” *Id.* According to Patent Owner, “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).” *Id.* at 35 (citing Ex. 2033 ¶¶ 10–19; Ex. 2007 ¶¶ 8, 11; Ex. 2008 ¶¶ 16–23).

Patent Owner alleges it developed “a cluster-computing architecture, used in its SEM™ and SET™’s products, that met this long-felt, unmet need.” Prelim. Resp. 35–36. Patent Owner asserts that SEM™ and SET™ “enable[] parallel execution of Mathematica and more general applications, respectively . . . without extensive specialized programming expertise, or the excessive investment of time and effort demanded by traditional parallel computing architectures.” *Id.* at 36–37 (citing Ex. 2033 ¶¶ 20–30; Ex. 2007 ¶¶ 10, 12 (discussing SEM™); Ex. 2008 ¶¶ 25–28 (discussing SET™)).

As further described by Patent Owner, to meet this “long-felt, unmet need,” “[t]he 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.” Prelim. Resp. 35–36. Patent Owner contends that "[t]he 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." *Id.* at 37 (citing Ex. 2033 ¶¶ 24, 27–29; Ex. 2001 ¶¶ 83–88). Patent Owner argues this alleged claimed architecture obtained "superior performance" and "ease of use" that "were unexpected" and "surprising." *Id.* at 38. Similarly, Patent Owner argues that "SEM™ and SET™ succeeded where others failed" and received praise, and "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." *Id.* at 38–40.

Patent Owner argues that "[t]here is sufficient nexus between the objective evidence related to SEM™ and SET™ and the challenged claims" because the "SEM™ and SET™ products practice at least the challenged independent claims." Prelim. Resp. 41. Patent Owner argues that "[e]ach of the objective indicia of non-obviousness results from the SEM™ and SET™ architecture embodied by the challenged claims." *Id.* at 45. Patent Owner argues that "[o]ther parallel-computing architectures failed to meet the long-felt, unmet need precisely because they lacked SEM™ and SET™'s claimed architecture." *Id.* at 45 (citing Ex. 2033 ¶ 37). Patent Owner argues the "claimed architecture" is responsible for all of its asserted objective indicia of nonobviousness. *Id.* (citing Ex. 2033 ¶¶ 39–42, 46–47).

Nevertheless, even if the SEM™ and SET™ products fall within the broad scope of claim 1, for similar reasons to those underlying the claim construction as set forth above (§ II.C), the challenged claims do not require the "unique . . . architecture" of SEM™ and SET™. *See* Prelim. Resp. 37. In simple terms, the challenged claims are not architecture-specific. *See*

MeadWestVaco Corp. v. Rexam Beauty and Closures, Inc., 731 F.3d 1258, 1264 (Fed. Circ. 2013) (error to consider "*secondary considerations of non-obvious [that] involved only fragrance-specific uses*," when "*claims now at issue are not fragrance-specific*" (emphasis added)). The court in *MeadWestVaco* held that the district court erred because it "credited evidence advanced to show long-felt need and commercial success specific to the perfume industry," and the claims were not limited to fragrance-specific dispensers. *See id.* (reasoning that "'objective evidence of non-obviousness must be commensurate in scope with the claims which the evidence is offered to support'") (quoting *Asyst Techs., Inc. v. Emtrak, Inc.*, 544 F.3d 1310, 1316 (Fed. Cir. 2008) (internal quote citation omitted)); *see also Brown & Williamson Tobacco Corp. v. Philip Morris Inc.*, 229 F.3d 1120, 1130 (Fed. Cir. 2000) (stating the presumption that commercial success is due to the patented invention applies "if the marketed product embodies the claimed features, and is coextensive with them").

The Federal Circuit recently addressed nexus in two related Board cases, *Fox Factory, Inc. v. SRAM, LLC*, 944 F.3d 1366, 1373–78 (Fed. Cir. 2019) ("*Fox Factory I*") and *Fox Factory, Inc. v. SRAM, LLC*, 813 F. App'x 539, 542 (Fed. Cir. 2020) (*Fox Factory II*). In *Fox Factory II*, the court characterized the Board's holding underlying *Fox Factory I*, as follows:

> In [*Fox Factory I*], this court held that the Board misapplied the legal requirement, incumbent upon patent owners, of showing a nexus between evidence of secondary considerations and the obviousness of the claims of that patent—in particular, the requirement that the product from which the secondary considerations arose is "coextensive" with the claimed invention. *Fox Factory*, 944 F.3d at 1373–78. Contrary to the Board's view, we reaffirmed in that case that a product is not

> coextensive with a claimed invention simply because it falls within the scope of the claim.

Fox Factory II, 813 F. App'x at 542. In other words, no presumption of nexus exists for products that simply fall in the broad scope of the claims if the products are not coextensive with the claims.

As noted above, according to Patent Owner, "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"—i.e., a "unique . . . architecture." *See* Prelim. Resp. 35–37. As construed above, however, the claims do not require a communication layer between the user interface and kernels—the central feature relied upon by Patent Owner for its objective evidence of nonobviousness. *See supra* § II.C; Pet. Reply 6 (arguing that Dr. Dauger's testimony "rests on an incorrect claim construction"). Therefore, on this preliminary record, because the claims do not require this allegedly "unique . . . architecture," the claims are not coextensive (or reasonably commensurate in scope) with the products and other asserted evidence, so no nexus exists.

For the foregoing reasons and on this preliminary record, Patent Owner fails to meet its burden of establishing a nexus between the objective evidence regarding its SEM and SET products and the claims of the '612 patent. We, therefore, do not accord substantial weight to such evidence. *Fox Factory I*, 944 F.3d at 1373. For the sake of completeness, however, we address Patent Owner's remaining allegations relating to objective indicia of nonobviousness.

(ii) Long-Felt Need and Failure of Others

"The existence of a long-felt but unsolved need that is met by the claimed invention is . . . objective evidence of non-obviousness." *Millennium Pharms., Inc. v. Sandoz Inc.*, 862 F.3d 1356, 1369 (Fed. Cir. 2017) (citing *In re Cyclobenzaprine Hydrochloride Extended-Release Capsule Patent Litig.*, 676 F.3d 1063, 1081–83 (Fed. Cir. 2012)). "Long[-]felt need is closely related to the failure of others. Evidence is particularly probative of obviousness when it demonstrates both that a demand existed for the patented invention, and that others tried but failed to satisfy that demand." *Cyclobenzaprine*, 676 F.3d at 1082.

Patent Owner argues "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." Prelim. Resp. 35. Patent Owner argues that its SEM and SET products met this need. *Id.* at 35–37 (citing Ex. 2001 ¶¶ 83–88; Ex. 2007 ¶¶ 10, 12; Ex. 2008 ¶¶ 25–28; Ex. 2033 ¶¶ 10–30).

Petitioner argues that "[Dr.] Dauger provides no evidence or explanation for why interposing the communications software between the user interface and the kernel, as compared to connecting the communications software in some other way, would make any difference at all to the user." Pet. Reply 6. Petitioner argues that this testimony is also unpersuasive because it relies on an incorrect claim interpretation that requires the cluster node modules to accept instructions from the user interface without the instructions first passing through any kernel. *Id.* at 6–8.

Patent Owner argues that Petitioner's arguments regarding claim construction are outside the scope of our Order authorizing Petitioner to file its Reply. PO Sur-reply 6–9. To the contrary, as discussed in considering nexus above (§ II.D.6.c.ii), the scope of the claims is relevant to objective indicia of nonobviousness.

Regarding the failure of others, Patent Owner acknowledges that others developed automatic parallelizers and universal compilers that converted serial code to parallel code, but argues that none achieved performance comparable to traditional parallel-computing architectures. Prelim. Resp. 39; *see also* PO Sur-reply 3.

Petitioner argues that "P[atent] O[wner]'s argument is irrelevant because the claims neither recite 'automatic' or 'universal' parallelization nor require a specific level of optimization or advantageousness." Pet. Reply 2 (citing *ABT Sys., LLC v. Emerson Elec. Co.*, 797 F.3d 1350, 1362 (Fed. Cir. 2015)).

Patent Owner relies on the testimony of Dr. Dauger, Dr. Bhansali, and Mr. Bancroft to support its arguments about a long-felt, unmet need. *See* Prelim. Resp. 33–37. However, none of these declarants establishes that others unsuccessfully attempted to solve the problem. Dr. Dauger states that "[n]umerous others had tried to implement automatic parallelizers or universal compilers that would take serial object code as input and output object parallel code," but "[n]one of these efforts succeeded to produce accurate parallel code that was sufficiently optimized or advantageous enough to catch on." Ex. 2033 ¶ 37. These conclusory and uncorroborated statements fail to establish that others actually tried to solve the asserted problem. The testimony of the other declarants fares no better. Dr. Bhansali merely states that he was not aware of any other products like the SEM product. Ex. 2007 ¶ 11. Mr. Bancroft states that he "had heard rumo[]rs of people trying to develop a universal parallelizer that could be used to automatically parallelize serial code." Ex. 2008 ¶ 30. None of this testimony persuasively establishes that others actually tried and failed to solve the problem asserted by Patent Owner.

Accordingly, for the foregoing reasons and on this preliminary record, Patent Owner's evidence of long-felt need and failure of others is weak.

(iii) Unexpected Results

"If a patent challenger makes a prima facie showing of obviousness, the owner may rebut based on 'unexpected results' by demonstrating 'that the claimed invention exhibits some superior property or advantage that a person of ordinary skill in the relevant art would have found surprising or unexpected.'" *Procter & Gamble Co. v. Teva Pharms. USA, Inc.*, 566 F.3d 989, 994 (Fed. Cir. 2009) (quoting *In re Soni*, 54 F.3d 746, 750 (Fed. Cir. 1995)). "To be particularly probative, evidence of unexpected results must establish that there is a difference between the results obtained and those of the closest prior art, and that the difference would not have been expected by one of ordinary skill in the art at the time of the invention." *Bristol-Myers Squibb Co. v. Teva Pharms. USA, Inc.*, 752 F.3d 967, 977 (Fed. Cir. 2014); *see also Kao Corp. v. Unilever U.S., Inc.*, 441 F.3d 963, 970 (Fed. Cir. 2006) ("'[W]hen unexpected results are used as evidence of nonobviousness, the results must be shown to be unexpected compared with the closest prior art.'") (quoting *In re Baxter Travenol Labs.*, 952 F.2d at 392).

Patent Owner argues that the performance and ease of use of its SEM and SET products was unexpected. Prelim. Resp. 37–39 (citing Ex. 2033 ¶¶ 38–41). Patent Owner argues that its SEM product outperformed gridMathematica. *Id.* at 37–38. Patent Owner argues that it was able to parallelize Wolfram Research's Mathematica, Apple's HD QuickTime Exporter, and Equalis's Scilab using its SET product in a much shorter time period than expected. *Id.* at 38.

Petitioner argues that Patent Owner's reliance on the testimony of Dr. Dauger is unpersuasive because Dr. Dauger is a listed inventor of the '612 patent. Pet. Reply 1. Petitioner also argues that Dr. Dauger's "testimony is also irrelevant because it compares the claimed invention against gridMathematica, not the 'closest prior art.'" *Id.* (citing *In re Harris*, 409 F.3d 1339, 1344 (Fed. Cir. 2005); *Trs. of Columbia Univ. v. Illumina, Inc.*, 620 F. App'x 916, 922 (Fed. Cir. 2015)).

Patent Owner replies that "Dr. Dauger's testimony was submitted under oath and penalty of perjury" and the other evidence cited in its Preliminary Response support its contention that the SEM and SET products exhibited surprising results. PO Sur-reply 1–2. Patent Owner also argues that it "chose the closest prior art by comparing SEM™ with gridMathematica and SET™ with the conventional method of parallelizing applications." *Id.* at 2 (citing Ex. 2033 ¶ 40).

Patent Owner relies almost exclusively on the testimony of Dr. Dauger in asserting the surprising results of the SEM and SET products. *See* Prelim. Resp. 37–39 (citing Ex. 2033 ¶¶ 38–41). Dr. Dauger testifies the he was surprised that the SEM product performed better than gridMathematica and that Patent Owner was able to parallelize Mathematica "in one man-month." Ex. 2033 ¶¶ 38–41. Dr. Dauger is an inventor of the '612 patent, was Patent Owner's CTO, and is currently a consultant employed by Patent Owner. Ex. 1001, code (75); Ex. 2033 ¶ 1; Ex. 2015, 2; Ex. 2018, 2. On this preliminary ecord, Dr. Dauger's testimony about his personal surprise at the SEM and SET products that he helped create unpersuasive to establish unexpected results of these products. *See In re Cree*, 818 F.3d 694, 702 (Fed. Cir. 2016) (citing *Power-One v. Artesyn Techs., Inc.*, 599 F.3d 1343, 1352 (Fed. Cir. 2010) (concluding that "self-serving statements from researchers about their own work" do not have the same credibility as statements made by disinterested parties). Notably, Patent Owner provides no evidence to corroborate Dr. Dauger's assertions of unexpected results.

Additionally, Patent Owner provides no comparative testing against any prior art configuration, be it the closest or otherwise. Although Dr. Dauger testifies that some testing was performed (*see* Ex. 2033 ¶ 39), no documentation or data are provided from that testing to substantiate his assertions. This is the type of conclusory evidence that has been found insufficient. *See, e.g.*, *In re Lindner*, 457 F.2d 506, 508 (CCPA 1972) ("This court has said previously that mere lawyers' arguments unsupported by factual evidence are insufficient to establish unexpected results. . . . Likewise, mere conclusory statements in the specification and affidavits are entitled to little weight when the Patent Office questions the efficacy of those statements.").

Furthermore, Patent Owner does not persuasively argue that gridMathematica is the closest prior art. As noted by Petitioner, Patent Owner does not compare its products to the asserted references or the Distributed Maple system disclosed therein.

Accordingly, for the foregoing reasons and on this preliminary record, Patent Owner's evidence of unexpected results is weak.

(iv) Industry Praise

"Evidence that the industry praised a claimed invention or a product that embodies the patent claims weighs against an assertion that the same claimed invention would have been obvious. Industry participants, especially competitors, are not likely to praise an obvious advance over the known art." *Apple Inc. v. Samsung Elecs. Co.*, 839 F.3d 1034, 1053 (Fed. Cir. 2016) (en banc).

Relying on the asserted statements of Dr. Bhansali and Yuko Matsuda, Patent Owner argues that industry praise supports patentability of the '612 patent claims. Prelim. Resp. 39–40. According to Patent Owner, Dr. Bhansali found the SEM product to be efficient for load balancing issues and Mr. Matsuda endorsed the SEM product. *Id.*

Petitioner argues that "[Dr.] Bhansali focuses on 'load balancing,'" which "has no nexus because the patents do not assert that load balancing was novel or non-obvious." Pet. Reply 3–4 (citing *Kennametal, Inc. v. Ingersoll Cutting Tool Co.*, 780 F.3d 1376, 1385 (Fed. Cir. 2015)). Petitioner argues that Patent Owner's proposed construction of "cluster node module" excludes the only method of load balancing disclosed in the '612 patent. *Id.* at 4–5. Petitioner also argues that the references asserted in the Petition teach load balancing. *Id.* at 4.

Patent Owner argues that Petitioner's arguments regarding claim construction are outside the scope of our Order authorizing Petitioner to file its Reply. PO Sur-reply 1, 4–6. To the contrary, as discussed in considering nexus above (§ II.D.6.c.ii), the scope of the claims is relevant to objective indicia of nonobviousness.

In order for evidence of industry praise to be probative of nonobviousness, the evidence must relate specifically to features of the claimed invention. *See Apple*, 839 F.3d at 1053–55 (discussing "substantial evidence of praise in the industry that specifically related to features of the claimed invention"). As argued by Patent Owner, Dr. Bhansali testifies that he found the SEM product to be efficient for load balancing issues, by which he means "*the distribution of different parts of an algorithm or application across different nodes* and the overall process of parallelizing the algorithm or application." Ex. 2007 ¶ 12 (emphasis added). The '612 patent refers to load balancing in a similar manner. *See* Ex. 1001, 21:8–55. As correctly noted by Petitioner, the challenged claims do not recite load balancing or otherwise require distributing commands among the nodes in a particular manner. Therefore, on this preliminary record, Dr. Bhansali's testimony appears to focus on unclaimed features such that it is not probative of nonobviousness.

Regarding the asserted statements made by Mr. Matsuda, Patent Owner cites to the Dauger Declaration rather than any submission endorsed by Mr. Matsuda. Prelim. Resp. 40 (citing Ex. 2033 ¶¶ 44–45). Dr. Dauber cites to a slide deck which he appears to have prepared and the substance of which consists only of two quotations. Ex. 2033 ¶ 44 (citing Ex. 2018, 4); *see also* Ex. 2018, 2 (listing Dean E. Dauger, Ph.D. as the author). On this record, the uncorroborated statements of Dr. Dauber are unpersuasive to support the asserted statement of Mr. Matsuda.

Dr. Dauger also cites to Exhibit 2023, referring to it as a "white paper" written by Mr. Matsuda. Ex. 2033 ¶ 45 (citing Ex. 2023, 2). Initially, it is not clear what significance a "white paper" carries. In any event, in the sentence cited by Patent Owner, Mr. Matsuda merely states that the SEM product "stands in an advantageous position" compared with an undefined "Parallel Computing Toolkit" when used with Mathematica. Ex. 2023, 2. This is not the type of competitor praise that courts have found to be indicative of non-obviousness. *See, e.g.*, *Apple*, 839 F.3d at 1053–54 (discussing "numerous internal Samsung documents that both praised Apple's slide to unlock feature and indicated that Samsung should modify its own phones to incorporate Apple's slide to unlock feature").

Accordingly, for the foregoing reasons and on this preliminary record, Patent Owner's evidence of industry praise is weak.

(v) Skepticism

Evidence of industry skepticism weighs in favor of nonobviousness. *See United States v. Adams*, 383 U.S. 39, 52 (1966). "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 nonobviousness." *WBIP, LLC v. Kohler Co.*, 829 F.3d 1317, 1335 (Fed. Cir. 2016).

Patent Owner argues that experts expressed skepticism that the SEM and SET products would work. Prelim. Resp. 40–41. Regarding the SEM product, Patent Owner relies solely on the testimony of Dr. Bhansali and Mr. Bancroft. *Id.* at 41 (citing Ex. 2007 ¶ 8; Ex. 2008 ¶ 32). Regarding the SET product, Patent Owner relies on the opinion of one unidentified Department of Energy ("DOE") "reviewer." *Id.* at 40 (citing Ex. 2033 ¶ 52); *see also* PO Sur-reply 3–4.

Petitioner argues that rather than expressing skepticism that Patent Owner's products would work, the DOE reviewers quoted in Patent Owner's exhibits "expressed skepticism over the bold performance claims made by P[atent] O[wner]." Pet. Reply 2–3.

Regarding the SEM product, Patent Owner relies solely on statements of Dr. Bhansali and Mr. Bancroft regarding their personal experience with the SEM product. Prelim. Resp. 40 (citing Ex. 2007 ¶ 8; Ex. 2008 ¶ 32). Dr. Bhansali states that "[w]hen I first learned about SEM, I was uncertain whether the product would perform as promised." Ex. 2007 ¶ 8. Dr. Bhansali states that he "graduated from Cal. Tech. with a dual B.S.-M.S. in physics, and engineering and applied science" and "received [a] Ph.D. in theoretical physics from Harvard University." *Id.* ¶¶ 3–4. Neither Patent Owner nor Dr. Bhansali provide any detail about his industrial experience. Notably, Patent Owner did not file a CV for Dr. Bhansali in the record. Therefore, Patent Owner fails to establish that Dr. Bhansali is an artisan of ordinary skill with respect to parallel or distributed computing, and his opinion testimony carries little weight.

Although Mr. Bancroft states that he provided the SEM product to "many experienced parallel programmers" (Ex. 2008 ¶ 32), he provides no information about these allegedly experienced programmers. Additionally, neither Patent Owner nor Mr. Bancroft provide his CV, so it is difficult to assess his credibility to provide technical testimony. *See* Ex. 2008. Mr. Bancroft's experience appears to involve business development matters rather than technical engineering or computer science research and development. *See id.* ¶¶ 4–14. Moreover, Mr. Bancroft is on Patent Owner's Business Advisory Board. *Id.* ¶ 33. Therefore, it appears that Mr. Bancroft is not a disinterested party and may have economic or other interest in Patent Owner's success in this proceeding. Accordingly, on this preliminary record, Mr. Bancroft's opinion testimony carries little weight.

Regarding the SET product, Petitioner sufficiently shows that the DOE reviewers appear to indicate that the submissions they reviewed lacked sufficient detail for them to evaluate the performance assertions made in the submissions. *See* Pet. Reply 2–3. Patent Owner relies on the comments of "Reviewer 2" of Exhibit 2019. Prelim. Resp. 40 (citing Ex. 2033 ¶ 52; Ex. 2019, at 2). This reviewer states that "[t]he proposal . . . provides no quantitative or qualitative evidence of (efficient or not) use of compute[r] resou[r]ces by SET." Ex. 2019, 2. This reviewer also states that "[t]he applicants have not demonstrated quantitatively that their technology provides real results. . . . SET may prove to be the great success the applicants suggest, but there is no proof that it works on real CAD/CAM/CAE applications." *Id.* at 3. Continuing, this reviewer states that "[t]he applicants haven't clearly defined the nature of the plasma code, nor the effort in porting it to SET." *Id.* Thus, the statements of Reviewer 2 appear to stem from a lack of detail to assess the credibility of the assertions in the submissions.[9]

Other reviewers similarly identify a lack of detail in the submissions. Ex. 2019, 4 ("[T]he main concepts of this proposal have not been presented in any substantial detail." "There is no sound plan to showing that this SET-based approach can be commercially viable."); Ex. 2021, 1 ("[T]here is no plan to compare the performance of the applications compared to their theoretical performance."), 2 ("The auto parallelization tools have not

---

[9] The submissions to the DOE are not of record in this case.

provided high performance as they usually have too many generalizations to take advantage of a particular computing architecture. I do not have evidence that the SET tool is any different."), 4 ("The applicant has provided a general outline of the comparison test approach, but further details in the work plan are needed."; "While there is an overall projection of technical relevance, the lack of specificity in the proposed test situation presents a high level of uncertainty in achieving the more ambitious goals of this proposal."); Ex. 2022, 3 ("The performance of SET in Linux and Mac OS operating environments, the type of efficiency increases achieved, and the strength and limitations of the SET approach are not adequately described."), 4 ("The applicant has provided a general description of the technical problem and work plan, but specific details of the technical challenges to be encountered with the SET technology should be described in greater detail."). Thus, to the extent one reviewer expressed skepticism that the SET product would perform as claimed, this evidence is undercut by the overwhelming expression of a lack of detail provided in the submitted proposals that were reviewed.

Accordingly, for the foregoing reasons and on this preliminary record, Patent Owner's evidence of skepticism of others is weak.

(vi) Copying

"Copying may indeed be another form of flattering praise for inventive features." *Crocs, Inc. v. ITC*, 598 F.3d 1294, 1311 (Fed. Cir. 2010). Copying "requires evidence of efforts to replicate a specific product." *Wyers v. Master Lock Co.*, 616 F.3d 1231, 1246 (Fed. Cir. 2010). "This may be demonstrated either through internal documents; direct evidence such as disassembling a patented prototype, photographing its features, and using the photograph as a blueprint to build a virtually identical replica; or access to, and substantial similarity to, the patented product (as opposed to the patent)." *Iron Grip Barbell Co. v. USA Sports, Inc.*, 392 F.3d 1317, 1325 (Fed. Cir. 2004) (internal citations omitted). "We note, however, that a showing of copying is only equivocal evidence of nonobviousness in the absence of more compelling objective indicia of other secondary considerations." *Ecolochem, Inc. v. S. Cal. Edison Co.*, 227 F.3d 1361, 1380 (Fed. Cir. 2000); *see also In re GPAC*, 57 F.3d 1573, 1580 (Fed. Cir. 1995) ("[M]ore than the mere fact of copying by an accused infringer is needed to make that action significant to a determination of the obviousness issue." (quoting *Cable Elec. Prods. v. Genmark, Inc.*, 770 F.2d 1015, 1028 (Fed. Cir. 1985))); *Institut Pasteur & Université Pierre et Marie Curie v. Focarino*, 738 F.3d 1337, 1347–48 (Fed. Cir. 2013) ("Copying requires duplication of features of the patentee's work based on access to that work, lest all infringement be mistakenly treated as copying. . . . But the Board did not analyze whether Pasteur's showing of the similarities of its method to the content of the cited publications, *e.g.,* their use of the same specific GIIE endonuclease, indicated that the publications' authors had access to, and borrowed from, the Pasteur sources."); *Liqwd, Inc. v. L'Oreal USA, Inc.*, 941 F.3d 1133, 1136–39 (Fed. Cir. 2019) (mere access to information coupled with allegations of infringement are not sufficient evidence of copying, absent evidence of other circumstances, such as copying specifics of a disclosed or actual product, or altering a design after obtaining access to the information). "Of course, the proponent of objective evidence offered to show nonobviousness, such as copying, must show that a nexus exists between the evidence *and the claimed features of the invention*." *Liqwd*, 941 F.3d at 1138 (emphasis added).

Patent Owner asserts that it provided information to Petitioner regarding its SET product during a November 2012 meeting and in a subsequent "email attaching a specially tailored data sheet." Prelim. Resp. 45–47 (citing Ex. 2033 ¶ 56). Patent Owner argues that "Petitioner then copied the claimed invention by incorporating the claimed architecture into Petitioner's GPGPUs in the manner described by the datasheet" and "named its GPU interconnect architecture, which uses the claimed structure, NVLink™."[10] *Id.* at 46 (citing Ex. 2033 ¶ 59); *see also* PO Sur-reply 9–10.

Petitioner traverses Patent Owner's assertions of copying, calling it "a gross misrepresentation to the PTAB." Pet. Reply 8–9. Petitioner asserts that the inventors of the '612 patent sent an unsolicited email to one of its employees attaching a public SET datasheet that "did not have any suggestion of 'provid[ing] a communications infrastructure for direct all-to-all communications between each GPU.'" *Id.* at 8 (alteration in original) (citing Prelim. Resp. 56).

Patent Owner provides no evidence to support its assertion. Patent Owner does not provide any description of the NVLink product or compare this product to the claims of the '612 patent. On this record, Patent Owner's conclusory assertions are inadequate to establish copying of the claimed invention by Petitioner.

---

[10] Dr. Dauger refers to NVIDIA's "general-purpose GPU ('GPGPU') supercomputing" and describes "NVIDIA'S Tesla GPGPUs as hardware black boxes on which the back end executes." Ex. 2033 ¶¶ 56–57 (citing Ex. 2020, 4, Fig. 3).

Accordingly, for the foregoing reasons and on this preliminary record, Patent Owner's evidence of copying is weak.

d. Claims 6, 10, 12, 13, 15, and 29

Building on its showing with respect to claim 1, Petitioner presents a sufficient showing supported by the record with respect to dependent claims 6, 10, 12, 13, 15, and 29. *See* Pet. 48–59. Patent Owner does not address these dependent claims individually.

e. Summary

Based on the foregoing discussion, Petitioner sufficiently establishes for purposes of institution that the combination of Schreiner1, Schreiner2, Schreiner3, and the Maple Guide, with or without the Distributed Maple Code, renders claims 1, 6, 10, 12, 13, 15, and 29 obvious. Accordingly, Petitioner establishes a reasonable likelihood of prevailing with respect to claims 1, 6, 10, 12, 13, 15, and 29.

E. Alleged Obviousness of Claims 4, 5, 7, and 8 over Schreiner1, Schreiner2, Schreiner3, Maple Guide, Distributed Maple Code, and Deitel

1. Deitel (Ex. 1019)

Deitel generally teaches implementation of a network operating system ("OS"):

> Advances in telecommunications technology have profoundly affected OS's. A **network OS** enables its processes to access resources (e.g., files) that reside on other independent computers on a network. . . . In a networked environment, a process can execute on the computer on which it is created or on another computer on the network. In some network operating systems, users can specify exactly where their processes run; in others, the operating system, determines where processes are executed. For example, the system may determine that a process can be more efficiently executed on a computer experiencing a light load.

Ex. 1019, 38.

Deitel also describes benefits of partitioning memory space and issues associated therewith:

> We (as OS designers) view main memory in terms of memory organization. Do we place only a single process in main memory, or do we place several processes in memory at once (i.e., do we implement multiprogramming)? If main memory contains several processes simultaneously, do we give each the same amount of space, or do we divide main memory into portions (called partitions) of different sizes?

Ex. 1019, 379.

2. Alleged Obviousness for Claims 4, 5, 7, and 8

Petitioner contends claims 4, 5, 8, and 8, which depend directly or indirectly from claim 1, would have been obvious over the combination of Schreiner1, Schreiner2, Schreiner3, Maple Guide, Distributed Maple Code, and Deitel. *See* Pet. 59–72. Claim 4 requires an "OS" to "incorporate[] at least one of the plurality of cluster node modules." Claim 5 depends from claim 4 and requires the OS to include "a programming interface exposed to an application program, the programming interface being configured to allow the application program to access functionality supplied by the at least one cluster node module."

Petitioner relies on Deitel to suggest moving some of the Distributed Maple functions into an OS, and to provide an application interface to utilize some of these functions. *See* Pet. 62–68. Petitioner contends it would have been obvious to employ well-known application interfaces ("APIs") in an OS in order to aid a user via mouse clicks or windows in simplifying access to functions in Distributed Maple. *See id.*

Claims 7 and 8 generally and respectively require single-node kernels and cluster node modules to be "configured to access a partitioned space to perform processing of data." Relying on Deitel, Petitioner contends with respect to claim 7 that it would have been obvious to partition the memory as claimed, because "[i]t was common in the prior art to include partitions in memory for the OS space and the user space so that user applications could not interfere with the OS." *See* Pet. 69 ("A process can interfere with the OS's memory—either intentionally or inadvertently—by replacing some or all of its memory contents with other data. . . . Protection in a single-user contiguous memory allocation systems can be implemented with a single boundary register built into the processor." (quoting Ex. 1019, 387–3880); citing Ex. 1005 ¶ 151)). Petitioner sets forth a similar showing with respect to claim 8. *See id.* at 71–72.

Based on the foregoing discussion and a review of the record, for purposes of institution, Petitioner sufficiently shows that the combination of Schreiner1, Schreiner2, Schreiner3, Maple Guide, Distributed Maple Code, and Deitel renders obvious the subject matter of dependent claims 4, 5, 7, and 8. Patent Owner does not address the dependent claims separately. Nonetheless, the burden remains on Petitioner to demonstrate unpatentability. *See Dynamic Drinkware, LLC v. Nat'l Graphics, Inc.*, 800 F.3d 1375, 1378 (Fed. Cir. 2015)

Accordingly, Petitioner establishes a reasonable likelihood of prevailing with respect to claims 4, 5, 7, 8.

F. Alleged Obviousness of Claims 4, 5, 7, and 8 over Schreiner1, Schreiner2, Schreiner3, Maple Guide, Distributed Maple Code, Deitel, and Chen

Petitioner alternatively adds Chen to its showing of alleged obviousness of claims 4 and 5 as described in the previous section. *See* Pet. 72–75. According to Petitioner, "Chen taught that integrating Distributed Maple's fault tolerance mechanisms implemented in the scheduler with the OS would increase fault tolerance." *Id.* at 73–74 (citing Ex. 1020, 75–76). Petitioner also contends that "integrating the scheduler in kernel space would allow a significant improvement in fault tolerance, and, as a further benefit, eliminate the need for sending copies of task results to the root node described in Schreiner3, resulting in an even more efficient system." *Id.* at 74 (citing Ex. 1020, 76, 81; Ex. 1005 ¶ 158).

Based on the foregoing discussion and a review of the record, for purposes of institution, Petitioner sufficiently shows that the combination of Schreiner1, Schreiner2, Schreiner3, Maple Guide, Distributed Maple Code, Deitel, and Chen renders obvious the subject matter of dependent claims 4 and 5.[11] Patent Owner does not address the dependent claims separately as

[11] Petitioner erroneously alleges that claims 7 and 8 depend from claim 4 and relies on that alleged dependency to assert the obviousness of claims 7 and 8 based further on Chen. Pet. 75. Therefore, Petitioner's showing based on Chen does not address claims 7 and 8 with sufficient particularity in this alternative ground of rejection. *See* Pet. 72–75. Nonetheless, the Board institutes on all challenges, and Petitioner provides sufficient particularity with respect to claims 7 and 8 in its alternative ground based on Deitel as addressed above (§ II.E). *See SAS Inst. Inc. v. Iancu*, 138 S. Ct. 1348, 1355 (2018) ("The Director, we see, is given only the choice 'whether' to institute an inter partes review. That language indicates a binary choice—either institute review or don't.").

noted above. Nonetheless, the burden remains on Petitioner to demonstrate unpatentability. *See Dynamic Drinkware,* 800 F.3d at 1378.

G. The Petition's Word Limit

Patent Owner alleges that the Petition "undercount[s] the true word count" by "about 320 excess words" because it uses "non-standard formats such as 'EX-1001,' 'Schreiner1,' and '¶216' to make two words count as one and cop[ies] text from the alleged prior art as images on pages 23–24, 27, and 50." Prelim. Resp. 55 (citing *Starbucks Corp. v. Ameranth, Inc.*, CBM2015-00091, Paper 16 at 2–3 (PTAB Jan. 29, 2016)). It is not clear if by "320 excess words," Patent Owner implies that the total word count would be 14,320, or if it would be 320 words in excess of the certified word count of 13,882. *See* Prelim. Resp. 55 (noting "37 C.F.R. § 42.24(a)(i) limits the Petition to 14,000 words" and "[t]he Petition certifies its word count is 13,882").

In any case, the allegedly non-standard formats and copied text from images do not result in an excessive word count or violate the spirit of the word count underlying the rule. For example, the Petition repeats the same annotated Figure 1 on pages 23, 28, and 33, but there are no figures on pages 24, 27, and 50. Petitioner adds words to the same annotated figure on pages 18, 23, 28, 33, and 38 of the Petition to identify how Petitioner reads elements of claim 1 onto the figure, resulting in clarity and ease of reading. However, Petitioner simply could have refrained from reproducing the same figure multiple times in order to minimize the word count at the expense of clarity and ease of reading. Patent Owner does not cite a case in which the Board granted relief based on the use of such alleged non-standard formats for citations. Patent Owner also does not specify any form of relief that it seeks. *See id.* Nevertheless, the Board hereby authorizes both parties to use the allegedly non-standard citation formats in future papers in this trial.

III. CONCLUSION

After considering the evidence and arguments presented in the Petition and the Preliminary Response and additional briefing, we determine Petitioner has demonstrated a reasonable likelihood that it would prevail with respect to most of its unpatentability challenges and with respect to the challenged claims. *See supra* note 11. We institute an *inter partes* review on the challenged claims and all of the grounds presented in the Petition. At this stage of the proceeding, we have not made a final determination as to the patentability of these challenged claims.

IV. ORDER

Accordingly, it is

ORDERED that pursuant to 35 U.S.C. § 314, *inter partes* review is instituted as to the challenged claims of the '612 patent with respect to all grounds of unpatentability presented in the Petition; and

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

For PETITIONER:

Brent Yamashita
Jonathan Hicks
DLA Piper LLP
brent.yamashita@dlapiper.com
jonathan.hicks@dlapiper.com
NVIDIA-ACS-IPR@us.dlapiper.com

For PATENT OWNER:

Jon W. Gurka
Ted M. Cannon
Cheryl T. Burgess
Knobbe Martens Olson and Bear, LLP
2jwg@knobbe.com
2tmc@knobbe.com
2ctb@knobbe.com
BoxZTANNL.017LP@knobbe.com