Trials@uspto.gov
571-272-7822

Paper 9
Entered: May 13, 2021

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

NVIDIA CORP.,
Petitioner,

v.

ADVANCED CLUSTER SYSTEMS, INC.,
Patent Owner.

IPR2021-00108
Patent 8,676,877 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, 5, and 8–11 (the "challenged claims") of U.S. Patent No. 8,676,877 B2 (Ex. 1001, the "'877 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. *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 '877 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 '877 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 '877 patent: IPR2020-01608, IPR2021-00019, IPR2021-00020, and IPR2021-00075. Pet. 3;

Paper 3, 1.

C. The '877 patent

The '877 patent, titled "Cluster Computing Using Special Purpose Microprocessors," "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), 1:22–24.

Embodiments of the '877 patent include enabling a software package to benefit from a plurality of nodes in a cluster. Ex. 1001, 2:12–24. 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 2:21–24.

Figure 1 of the ’877 patent follows:

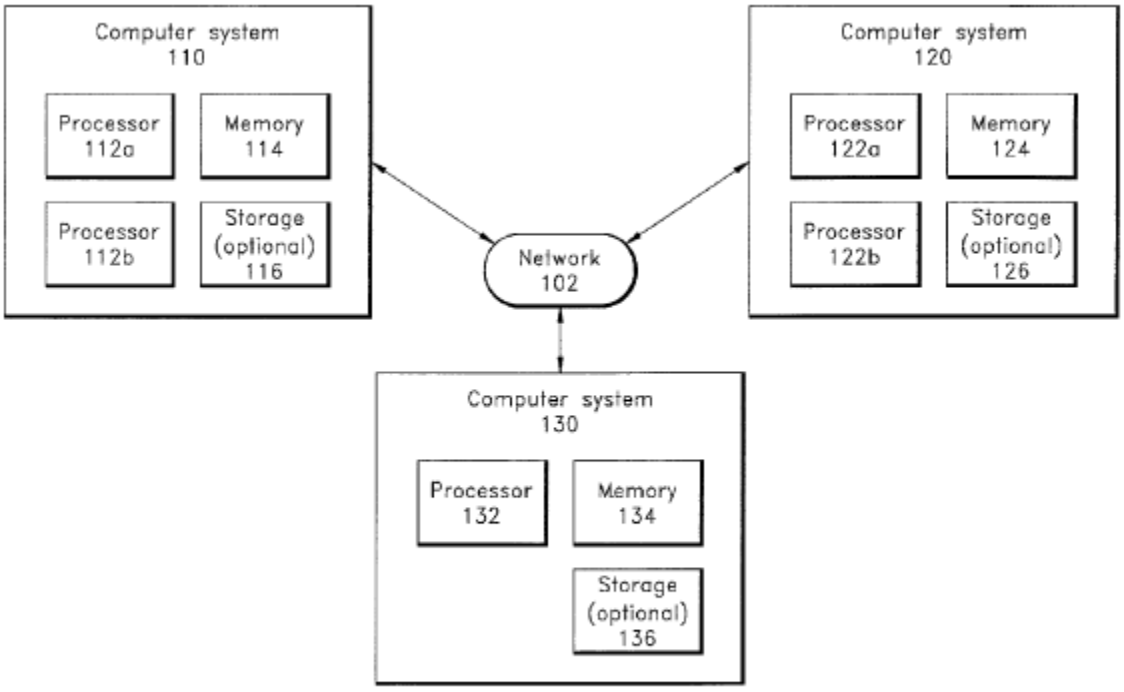


FIG. 1

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, 4:65–5:1. 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 5:1–32. Each processor includes an independent processing core, or “node,” that is capable of single-threaded execution. *See id.* at 4:48–50, 5:8–11.

Figure 2 of the ’877 patent, showing relationships among software modules in computer cluster 100 (Ex. 1001, 5:18–20), follows:

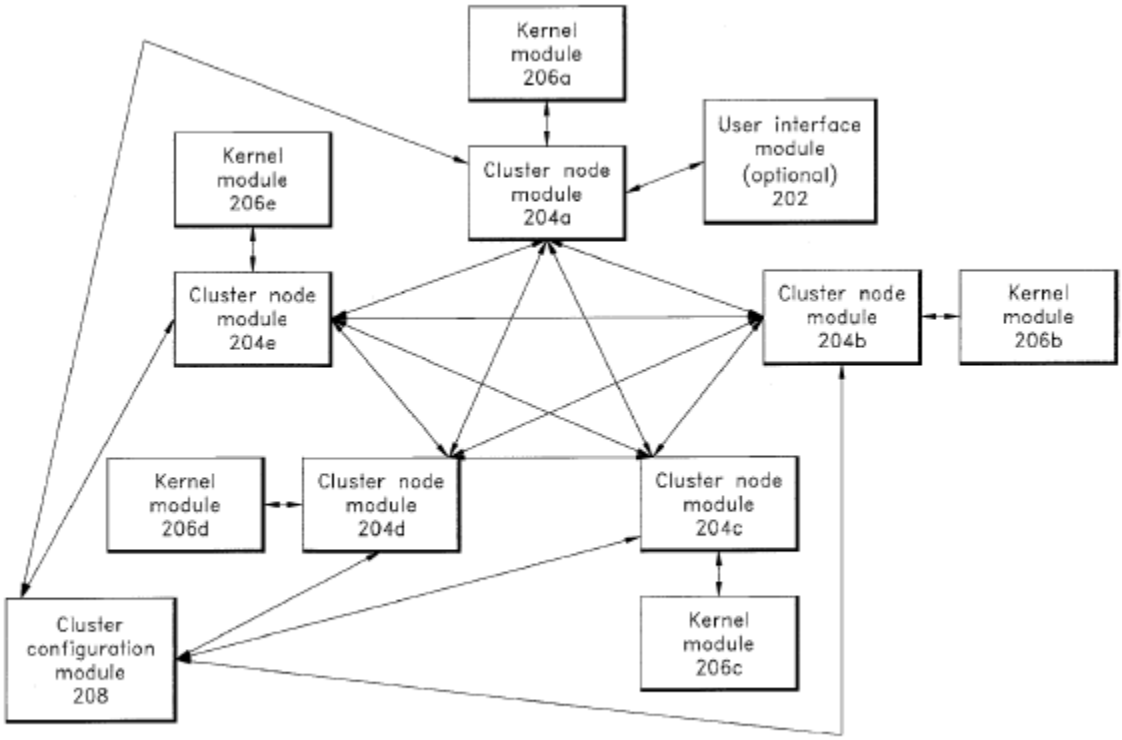


FIG. 2

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, 5:34–55.

In the embodiment of Figure 2, each kernel module communicates with a single cluster node module. *See* Ex. 1001, 5:34–38. 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 11:34–53. In

the Figure 2 embodiment, each cluster node module communicates with each of the other cluster node modules. *Id.* at 5:42–55. 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 23:2–8, 23:44–50, 24:11–18. Each cluster node module communicates the command to its respective kernel module, such as for example, Mathematica kernels. *Id.* at 23:18–23, 24:27–33. Each kernel module processes the command and returns a result to its respective cluster node module. *Id.* at 24:34–38. The cluster node module can report the result to the other cluster node modules. *Id.* at 24:38–57. 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 24:62–67.

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, 1:37–46.

D. Illustrative Claims 1 and 10

The Petition challenges claims 1, 5, and 8–11. Pet. 1. Claims 1, 8 and 10 are independent. Independent claims 1 and 10 illustrate the challenged claims at issue:

> 1. [a.][i.] A system for performing an instruction received from a front end [ii] by executing commands [iii.] on one or more special purpose microprocessors, the system comprising:
>
> [b.][i.] a plurality of nodes, wherein each node is configured to access a computer-readable memory system comprising [ii.] program code for a single-node kernel module, and wherein each single-node kernel module is [iii.] configured to interpret instructions received by the single-node kernel module into commands that are executable by a special purpose microprocessor;
>
> [c.][i.] a plurality of cluster node modules, wherein [ii.] each cluster node module is stored in a computer-readable memory system and [iii.] configured to communicate with a single-node kernel and with one or more other cluster node modules, [iv.] to accept instructions, and to interpret at least some of the instructions such that the plurality of cluster node modules communicate with one another in order to act as a cluster in executing commands using one or more hardware processors; and
>
> [d.] a communications system configured to connect the plurality of nodes;
>
> [e.] wherein the plurality of cluster node modules cooperate to interpret and translate, as needed, the instruction 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 front end.

Ex. 1001, 29:54–30:12 (information added to conform to Petitioner's nomenclature).

> 10. [a.] A method of performing an instruction received from a front end by executing commands on one or more special purpose microprocessors, the method comprising:
>
> [b.] communicating an instruction from a front end to one or more cluster node modules connected to one another by a communications system;

> [c.] for each of the one or more cluster node modules, communicating a message based on the instruction to a respective kernel module associated with the cluster node module, wherein the respective kernel module is configured to interpret the message into commands that are executable by a special purpose microprocessor;
>
> [d.] for each of the one or more cluster node modules, receiving a result from the respective kernel module associated with the cluster node module; and
>
> [e.] for at least one of the one or more cluster node modules, responding to messages from other cluster node modules.

Ex. 1001, 30:59–31:9 (information added 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 (Ex. 1012–Ex. 1018) (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);

SUN Debuts UltraSPARC IV, EE Times, October 15, 2003 ("SPARC IV Article") (Ex. 1019);

AMD to Unveil Dual-Core PC Chips, Los Angeles Times, May 31, 2005 ("AMD Article") (Ex. 1020);

Maple V, PC Magazine, August 1992, 419–425 ("PC Magazine1") (Ex. 1029);

Maple V Looks Better With Improved Graphics, PC Magazine, July 1993, 50 ("PC Magazine2") (Ex. 1030);

Anshuman Nayak, et al., "A Library based compiler to execute MATLAB Programs on a Heterogeneous Platform" ("Nayak") (Ex. 1031);

Kelly et al., U.S. Pat. No. 6,691,216 (filed October 24, 2001, issued Feb. 10, 2004) ("Kelly") (Ex. 1032).

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, 8–11	103(a) [1]	Schreiner1, Schreiner2, Schreiner3, Maple Guide,

[1] The filing date of the application resulting in the '877 patent precedes the effective date of Leahy-Smith America Invents Act ("AIA"), Pub. L. No.

Claim(s) Challenged	35 U.S.C. §	Reference(s)/Basis
		Distributed Maple Code, PC Magazine1, PC Magazine 2
1, 10, 11	103(a)	Schreiner1, Schreiner2, Schreiner3, Maple Guide, Distributed Maple Code, PC Magazine1, PC Magazine2, Nayak
5	103(a)	Schreiner1, Schreiner2, Schreiner3, Maple Guide, Distributed Maple Code, PC Magazine1, PC Magazine2, Nayak, Kelly
5	103(a)	Schreiner1, Schreiner2, Schreiner3, Maple Guide, Distributed Maple Code, PC Magazine1, PC Magazine2, SPARC IV Article, AMD Article

See Pet. 11. Petitioner submits declarations of Henry Tufo, Ph.D. (Ex. 1005) and Wolfgang Schreiner, Ph.D. (Ex. 1006), and Sylvia Hall-Ellis, Ph.D. (Ex. 1007) 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. 2036), in support of its Preliminary Response.

II. ANALYSIS

Petitioner challenges claims 1, 5, and 8–11 as obvious based on the grounds listed above. Pet. 1. Patent Owner disagrees. Prelim. Resp. 1–8.

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. 14 (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 '877 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 one of two claim construction disputes here:

> a plurality of cluster node modules, wherein each cluster node module is stored in a computer-readable memory system and configured to communicate with a single-node kernel and with one or more other cluster node modules, to accept instructions, and to interpret at least some of the instructions such that the plurality of cluster node modules communicate with one another in order to act as a cluster in executing commands using one or more hardware processors.

Each challenged claim recites a "cluster node module," although the challenged claims do not all recite the limitations (as quoted immediately above) that define how each cluster node module is "configured" in claim 1. For example, in addition to reciting receiving and responding limitations (*supra* § I.D), claim 10 also recites the following "cluster node module" limitations, which raises the second claim construction dispute here:

> communicating an instruction from a front end to one or more cluster node modules connected to one another by a communications system;
>
> for each of the one or more cluster node modules, communicating a message based on the instruction to a respective kernel module associated with the cluster node module, wherein the respective kernel module is configured to interpret the message into commands that are executable by a special purpose microprocessor.

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. 14.[3] Notwithstanding Petitioner's statement, Patent Owner

[3] Neither party argues that term "module" as recited in the challenged claims (i.e., "cluster node module" or "kernel module") is a means-plus-function

argues that "[t]he Petition does not comply with the requirement to construe the claims because it does not construe ***any*** claim terms, even though construction is necessary to resolve the Petition." Prelim. Resp. 47. To the contrary, Petitioner sufficiently identifies how it reads the claims onto Schreiner1's system, as discussed below. *See infra* § II.D.8.a. Also, it is 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 two central claim construction disputes underlying Petitioner's implicit claim constructions.

First, Patent Owner notes that each challenged claim recites a "cluster node module." Prelim. Resp. 9. This raises the first claim construction dispute as indicated above. Specifically, Patent Owner argues that

> [t]he 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."

Id. at 16 (citing Ex. 2001 ¶ 50).

---

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.

Patent Owner raises the second claim construction dispute with respect to claims 8 and 10.[4] *See* Prelim. Resp. 16–19. Patent Owner contends that independent claim 10 recites "communicating an instruction ***from*** a front end ***to*** one or more cluster node modules" and "for each of the one or more cluster node modules, communicating a message based on the instruction ***to*** a respective kernel module associated with the cluster node module." Prelim. Resp. 16 (emphasis by Patent Owner). According to Patent Owner, the Board should construe these recited phrases

> as establishing the following relative order in which an instruction is processed by the front end, cluster node modules, and kernels: (1) first, an instruction starts at the front end, (2) second, an instruction is "communicat[ed] [. . .] from [the] front end to" the "cluster node modules," and, (3) third, a "message based on the instruction" is "communicat[ed] . . . to a respective kernel.

Id. at 19–20 (quoting claim 10) (all alterations except the ellipses by Patent Owner).

Under this "relative order," Patent Owner argues that claim 10 "does ***not*** cover a different relative order in which the front end sends instructions to the kernels and the kernels then forward instructions to the cluster node modules." Prelim. Resp. 17. Patent Owner summarizes its claim construction argument for claim 10 in the following diagram:

[4] Patent Owner focuses only on claim 10 under this claim construction argument. However, independent claim 8 recites the same relevant limitations as independent claim 10 with respect to this dispute.

Correct order:	Front End → Cluster Node Module → Kernel
Incorrect order:	Front End → 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 10 that allows a kernel to pass an instruction from a front end to a cluster node module. *See id.* at 16–17 (citing Ex. 2001 ¶ 51). In other words, Patent Owner argues that claim 10 "excludes the kernels receiving instructions from the front end and forwarding instructions to the cluster node modules." *Id.* at 19–20. However, Patent Owner clarifies that the instructions can pass indirectly through *any* "device[]" or "component[]" *except a kernel*: "To be clear, this construction does ***not*** require instructions to be transmitted directly from the front end to the cluster node modules without passing through other devices, such as routers, switches, or other components." *See id.* at 20 n.4.

Therefore, both claim construction disputes involve assertions of implied negative limitations with respect to the cluster node modules as recited in the challenged claims. Contrary to Patent Owner's arguments, the plain language of claim 1 does not preclude indirect communications between cluster node modules "through a central server or master node." Similarly, the plain language of claim 10 does not preclude instructions from passing indirectly through a "kernel" before they reach the cluster node modules.

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, 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 anything about a central server or master node. The language also requires "each cluster node module . . . to accept instructions, and to interpret at least some of the instructions such that the plurality of cluster node modules communicate with one another in order to act as a cluster in executing commands using one or more hardware processors." This language does not prevent the communications from passing through a central server or master node.

Similar remarks apply to the plain language of claim 10. As Patent Owner notes, claim 10 first recites "communicating an instruction from a front end to one or more cluster node modules" and then recites "for each of the one or more cluster node modules, communicating a message based on the instruction to a respective kernel module associated with the cluster node module." *See* Prelim. Resp. 16. But these limitations simply do not recite the proposed negative limitation "without passing through a kernel." The first "communicating" limitation recites "communicating an instruction from a front end to one or more cluster node modules" without mentioning a kernel and without mentioning a direct connection. Just because the cluster mode module "communicat[es] a message based on the instruction [from a front end] to a respective kernel module associated with the cluster node module," this does not preclude the instruction from passing through a kernel or any other device or component on its way from a user interface to one or more cluster node modules. As noted above, Patent Owner generally argues that claim 10 allows an instruction to pass through intervening

"devices" or "other components" on its way from a user interface to one or more cluster node modules. Prelim. Resp. 20 n.4.

Contrary to Patent Owner's other arguments, the specification does not limit the claims in the manner argued. *See* Prelim. Resp. 13 (citing Ex. 1001, 22:45–46, 23:12–14, Fig. 2, Ex. 1001 ¶ 47). Patent Owner relies on Figure 2 and other selected passages to incorporate limitations into "cluster node modules" as recited in each challenged claim (*see id.*), but the specification 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, 3:65–67 (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, 2:7–11. One passage mimics the broad language of claims 8 and 10, 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.

Id. at 3:24–30.

This generic passage allows each cluster node module "to accept instructions from the user interface," without requiring such a module 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 and the specification as showing "'direct connections' between each cluster node module" in a peer-to-peer architecture. Prelim. Resp. 13 (citing Ex. 1001, 22:45–46, 23:12–14; Ex. 2001 ¶ 47). Patent Owner similarly contends that "there is no connection between the user interface module 202 and the kernel module 206a or any of the other kernel modules." *Id.* at 19. At one of the cited passages, the specification states "[t]he cluster node modules 204a–e establish communication with one another at 404. *In one embodiment*, each of the cluster node modules 204a–e establish[es] *direct connections* . . . with other cluster node modules . . . *launched on the computer cluster 100 by the cluster configuration module 208*." Ex. 1001, 23:12–17 (emphasis added).

As noted above, however, embodiments in the specification generally do not limit the claim terms. *See Hill-Rom Servs.*, 755 F.3d at 1371. Moreover, the cited passage implies that any direct connection only applies to "one embodiment." Ex. 1001, 23:12–17. Further, that embodiment includes "cluster configuration module 208." *See id.* But Patent Owner specifically argues that such modules "are optional parts of the cluster node modules," because the specification refers to "one embodiment." *See* Prelim. Resp. 10–11. Patent Owner also argues other claims in other patents specifically recite "optional parts of the cluster node modules," further showing that these components are optional. *Id.* at 11 (citing "IPR2020-01608, Ex. 1001, at claim 8 (received message queue), claim 9 (message

receiving queue), claim 13 (advanced functions module), claim 26 (MPI module)").

Patent Owner's logic extends to claims in other patents that specifically recite "peer-to-peer" architecture. *See* IPR2021-00019, Ex. 1001, 30:33–35 (claim 1 reciting "a mechanism for the nodes to communicate . . . with each other using a peer-to-peer architecture"). In other words, Patent Owner's arguments that the disclosed "peer-to-peer" architecture precludes an indirect connection between "cluster modules" in the challenged claims (*see* Prelim. Resp. 12–14) contradict its other arguments that unclaimed features here that other patent claims specifically recite imply the features are optional (*id.* at 10–11).

The specification also contradicts Patent Owner's arguments on this preliminary record with respect to claim 10. *See* Prelim. Resp. 16–20. 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, 22:41–44 (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." *Id.* at 25:30–34 (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. This connection between a kernel and a user interface implies that instructions pass through a kernel from a user interface to a cluster node module.

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, 22:14–39 (emphasis added).

Patent Owner argues that portions of this passage support its construction of claim 10. Prelim. Resp. 17 (citing Ex. 1001, 22:14–16, 22:32–34). 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. *See* Ex. 1001, 22:14–39. 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.* (emphasis added). In other words, to the extent altering this "behavior" involving cluster node modules somehow limits 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. *See id.*

In addition, the passage verifies that packets *normally* pass to kernel module 206a (at least for some contemplated embodiments). *See* Ex. 1001, 22:14–39. 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 or between cluster node modules, by precluding an intervening kernel, master node, or server. 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* node," while otherwise generally allowing for other intervening "components" or "devices," lacks requisite support in the specification. *See* Prelim. Resp. 20 n.4.

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.[5] *See, e.g.*, Ex. 1001, Fig. 2, 6:19–23 ("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."), 11:34–37 ("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."), 23:55–57 ("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 22 passage discussed and reproduced above. It also controls the other cluster node modules in a "procedure to shut down the system." *See id.* at 25:46–47.

The specification also indicates that a load balancing embodiment includes a "root processor" that assigns tasks to each of the cluster nodes. *See* Ex. 1001, 21:3–12. On this preliminary record, this further shows that the challenged claims do not require a cluster node module "to exchange messages such that each node can communicate tasks and data with other

[5] 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, 4:48–50. The specification also describes "computers, microprocessors, and/or processor cores ('nodes')." *Id.* at 1:26–27.

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, 71 ("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 claims 1 and 10. *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 and 8–11

Petitioner relies on the combined teachings of Schreiner1, Schreiner2, Schreiner3, the Maple Guide, the Distributed Maple Code, PC Magazine1, and PC Magazine2 to allege obviousness of claims 1 and 8–11. *See* Pet. 11.

1. Schreiner1 (Ex. 1008)

Schreiner1, entitled "Distributed Maple: parallel computer algebra in networked environments," bears a copyright date of 2003. Ex. 1008, 3.[6] Schreiner1 is a journal article authored by Dr. Schreiner, Christian Mittermaier, and Karoly Bosa. *Id.* 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.* 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

[6] All page number citations herein refer to page numbers added to the Exhibits by Petitioner. In some cases (for example with respect to Ex. 1008, and Ex. 1011), the Board converted citations by Petitioner to original page numbers into the page numbers added by Petitioner. The Board urges the parties to cite to the page numbers added by the parties to the Exhibits in future briefing.

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.

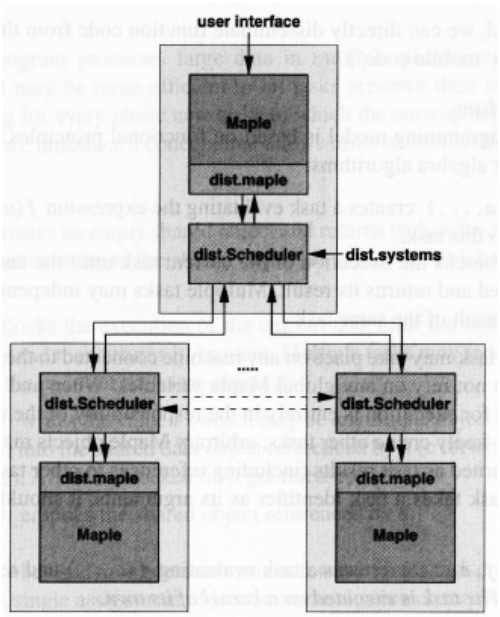


Fig. 1. Software architecture.

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. Schreiner1 (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. Schreiner2 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.* 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. Ex. 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 (also citing http://www.risc.uni-linz.ac.at/software/distmaple).

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

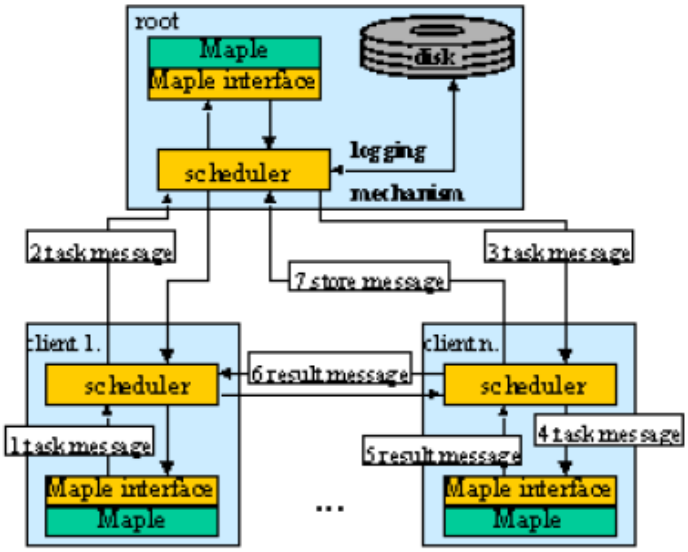


Figure 1: Execution Model

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. PC Magazine1 (Ex. 1029)

PC Magazine1 is a trade journal article describing aspects of Maple V, by Waterloo Maple Software. It describes an 80387 coprocessor, and states that "Maple V includes libraries for two- and three-dimensional geometry, projective geometry, differential forms, group, theory, Boolean algebra, number theory," and other mathematical functions. *See* Ex. 1029, 7 (Libraries, Fact File).

7. PC Magazine2 (Ex. 1030)

PC Magazine2 is a trade journal article describing aspects of Maple V, by Waterloo Maple Software, including its specialized mathematical functions and graphics. *See* Ex. 1030, 6.

8. Claims 1 and 8–11

a. Analysis of Claims 1 and 8–11

Petitioner relies on the combined teachings of Schreiner1, Schreiner2, Schreiner3, the Maple Guide, the Distributed Maple Code, PC Magazine1, and PC Magazine2, as supported by the testimony of Dr. Tufo and Dr. Schreiner, to allege obviousness of claims 1 and 8–11. *See* Pet. 15–70.

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. 15. Petitioner essentially contends that a person of ordinary skill would have consulted the references to learn details about the Maple system, including fault tolerances and capabilities, in order to combine desired features for running the software modules and system. *See id.* at 15–16.

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. 16–17 (citing Ex. 1008, 44; Ex. 1006 ¶ 49).[7] 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

[7] 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.

> 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 17 (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. 16 (citing Ex. 1008, 5–6; Ex. 1009, 4 (Abstract); 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 available under the GNU Library General Public License at http://www.risc.uni-linz.ac.at/software/distmaple." Ex. 1008, 5. Depending on the specific alleged prior art reference, Petitioner provides other motivation as detailed further below.

Claim's 1 preamble recites "[a] system for performing an instruction received from a front end by executing commands on one or more special purpose microprocessors, the system comprising." *See* Pet. 22; *supra* § I.D. Petitioner contends that even if the preamble limits the claim, Schreiner1 "contains actual screenshots of a Maple user interface frontend, showing instructions entered by the user after each '>' prompt." *Id.* at 24 (reproducing Ex. 1008, 6–7 (screen shot of a user interface display); citing Ex. 1005 ¶ 64). Petitioner also annotates and reproduces Schreiner1's Figure 1 (reproduced below and *see supra* § II.D.1), which shows a "user interface" connected to first node that includes a Maple kernel. *Id.* at 23.

Petitioner explains that Schreiner1's "nodes execute commands in response to the user instructions." Pet. 25. As an example, Petitioner explains that "Schreiner1 describes a user entering the Distributed Maple 'dist[all]' instruction into the root Maple kernel user interface: 'dist[all](*command*) lets the Maple statement command be executed on every Maple kernel connected to the distributed session.'" *Id.* (quoting Ex. 1008, 9; citing Ex. 1009, 30) (emphasis omitted). Petitioner also cites the example of the "dist[start] instruction" as an "instruction entered into the user interface." *Id.* (citing Ex. 1008, 8, 10–12).

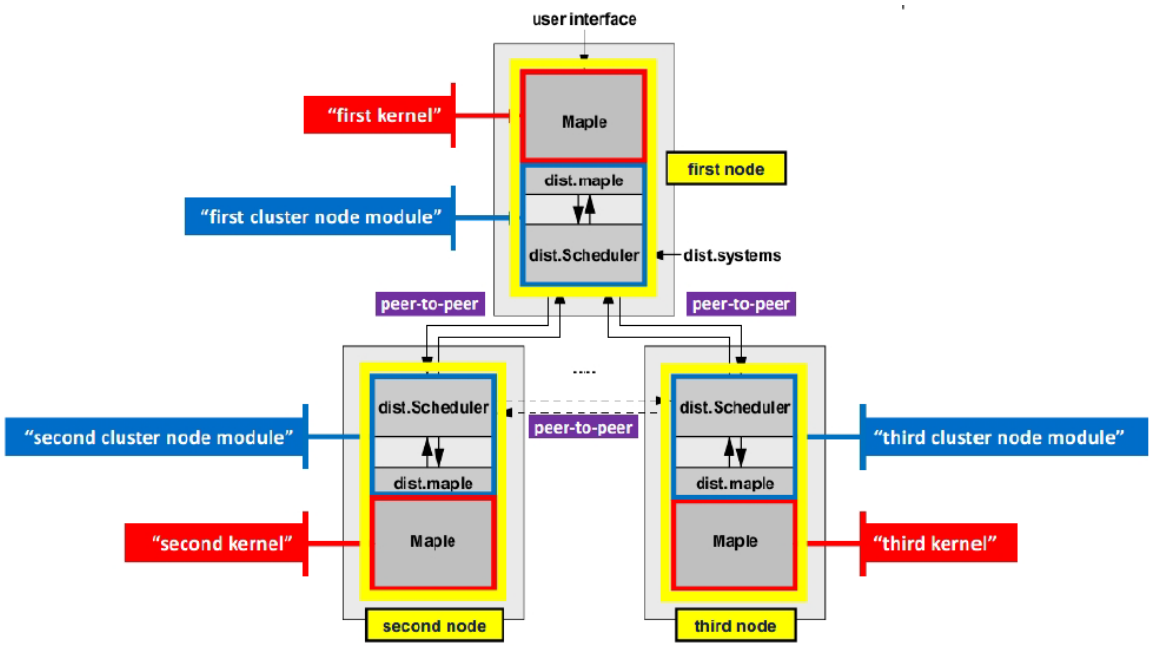
Petitioner relies on PCMagazine1 and PC Magazine2 as suggesting the use of special purpose microprocessors to run Maple V such as the "80387 coprocessor" or other math coprocessor. Pet. 26–27 (citing Ex. 1029; Ex. 1030). Petitioner contends that PC Magazine2 recommends a "math coprocessor." *Id.*; *see* Ex. 1030, 6 ("While Maple will use a math coprocessor, one is not essential, as it is for Mathematica."); Ex. 1029, 7 (Maple V "[r]equires" an "80387 coprocessor"). According to Petitioner, "a POSITA would have been motivated to read other materials on Maple V, including PC Magazine1 and PC Magazine2, which confirm that use of a special purpose math coprocessor was a basic feature of Maple V." Pet. 27 (citing Ex. 1005 ¶ 69). Petitioner adds that it would have been obvious to perform Maple math processes using a special purpose math coprocessor. *See id.* (citing Ex. 1005 ¶ 69).

Claim 1 also recites the following limitations:

> [b.] [i.] a plurality of nodes, wherein each node is configured to access a computer-readable memory system [ii.] comprising program code for a single-node kernel module, and wherein each single-node kernel module is configured to interpret instructions received by the single-node kernel module

into commands that are executable by a special purpose microprocessor.

To address these limitations, Petitioner annotates Schreiner1's Figure 1 as follows (Pet. 28):



According to Petitioner's annotations and descriptions, Figure 1 above shows single-node kernel modules (first kernel, second kernel, and third kernel) at each node (first node, second node, and third node), with "a processor executing a Maple kernel and Distributed Maple stored in memory." Pet. 29 (citing Ex. 1008, 9; Ex. 1005 ¶ 72). Petitioner explains that "[t]he Maple 'kernel consists of highly optimized C code.'" *Id.* (quoting Ex. 1011, 98; citing *id.* at 201 ("program such as Maple"), 277 ("either with Maple or with another program"), 11, 40, 102, 272 (Maple programs)). Petitioner explains that "Distributed Maple is also a software program . . . . likewise installed in storage and loaded in memory for execution by one or more programs." *Id.* (citing Ex. 1005 ¶ 74; Ex. 1008, 5, 7–22 (describing

the software system), 22–42 (describing application programs); Ex. 1009, 9; Ex. 1015; Ex. 1016).

Petitioner contends that "Schreiner1 teaches the use of Maple as a single-node kernel" and "[e]ach Maple kernel was configured to interpret user instructions." Pet. 30 (citing Ex. 1008, Abstract, 3, 8–9, Fig. 1; Ex. 1005 ¶ 77). Petitioner quotes the Maple Guide, stating the Maple kernel "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." *Id.* at 31 (quoting Ex. 1011, 98; citing Ex. 1005 ¶ 79) (emphasis by Petitioner). Petitioner adds that Schreiner1 describes "a 128 processor SGI Origin 3800 distributed shared memory multiprocessor" for implementing Distributed Maple and Maple. *Id.* at 29 (citing Ex. 1008, 25).

Petitioner also relies on Schreiner2 as "teach[ing] that high-level commands are interpreted by the kernel into lower-level code for execution by the processors." Pet. 31 (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 '877 patent." *Id.* at 32 (citing Ex. 1008, 4; Ex. 1005 ¶ 82).

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 memory system and [iii.] configured to communicate with a single-node kernel and with one or more other cluster node modules, [iv.] to accept instructions, and to interpret at least some of the

> instructions such that the plurality of cluster node modules communicate with one another in order to act as a cluster in executing commands using one or more hardware processors.

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), 34 (same). Petitioner explains that Schreiner's "Figure 1 depicts the dist.maple and dist.Scheduler components of each cluster node module." *Id.* at 34 (citing Ex. 1008, Fig. 1; Ex. 1005 ¶ 86).

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. 34 (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. 35 (quoting Ex. 1009, 24; citing Ex. 1005 ¶ 87).

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. 35.

Addressing limitation c.iii, Petitioner explains that the scheduler sends some of the Distributed Maple commands to all kernels. Pet. 36–41 (citing Ex. 1008, 9–10, 12, 14; Ex. 1009, 15). As an example, Petitioner quotes Schreiner1 as follows: "All remote schedulers send new tasks to the root node scheduler which distributes them among all machines." *Id.* at 37 (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). Petitioner also contends that "[t]he dist.maple component 'implements the interface between kernel and scheduler,'" providing the final link in sending commands to the kernels. *Id.* at 36 (quoting Ex. 1008, 8).

Relying on its annotated version of Schreiner's Figure 1 (as 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. 37–40 (citing Ex. 1008, 8–10, 14, Fig. 1 (arrows showing communications between dist.Scheduler), Fig. 5 (similar); Ex. 1010, 5, Fig. 5 (similar)).

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. 39 (quoting Ex. 1008, 13; citing Ex. 1008, 17; Ex. 1005 ¶ 97) (alteration by Petitioner).

As noted above, claim 1 also recites limitation c.iv, "a plurality of cluster node modules . . . configured . . . to accept instructions, and to interpret at least some of the instructions such that the plurality of cluster node modules communicate with one another in order to act as a cluster in executing commands using one or more hardware processors." Claim 1 does not explicitly require instructions from a user interface. To the extent claim 1 requires a user interface (or front end) to send instructions, Petitioner relies on the user interface in Schreiner1's Figure 1 (reproduced above) and provides other citations to Schreiner1.[8] *See* Pet. 43 (citing Ex. 1008, 7 ("The user interacts with Distributed Maple via a conventional Maple frontend

[8] In contrast to independent claim 1, independent claim 10 specifically recites "communicating an instruction from a front end to one or more cluster node modules" in the body of the claim. *See supra* §§ I.E; II.C). To address claim 10, Petitioner partly relies on its showing regarding the user interface in connection with claim 1 as discussed herein. *See* Pet. 64–65.

(text or graphical), i.e. she operates within the familiar Maple environment for writing and executing parallel programs.”)).

Regarding the limitation of the cluster node modules “accept[ing] instructions, and to . . . communicate with one another in order to act as a cluster in executing commands,” Petitioner contends that “the root dist.maple component waits for Distributed Maple ‘dist’ message-passing commands from the user interface, and forwards them to other cluster node modules using its scheduler” and “the cluster node modules . . . act as a cluster” to execute the commands. Pet. 42–43 (citing Ex. 1008, 9–12).

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. 43 (citing Ex. 1008, Fig. 1; Ex. 1005 ¶ 102).

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. For 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, ¶104.

Pet. 44.

Petitioner provides another example as disclosed in Scheiner1. *See* Pet. 45–46 (relying on a "dist[start] instruction" entered from a "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, 14; Ex. 1009, 30)).

Patent Owner responds by arguing that in Schreiner1, the root node is a master node and it schedules tasks between other nodes. *See* Prelim. Resp. 23–24. Accordingly, Patent Owner argues that Schreiner1 does not teach limitation c.iv under its proposed claim construction of "cluster node modules." *See id.*; *supra* § II.C. For example, Patent Owner argues as follows: "[B]ecause the purported cluster node modules of the prior art do not 'communicate tasks and data with other nodes without the tasks and data being required to go through a central server or master node,' they do not satisfy the 'cluster node module' limitation as properly construed." Prelim. Resp. 24. Contrary to this line of argument, as determined above for purposes of institution, the preliminary 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.

The final limitations of claim 1 follow:

[d.] "a communications system configured to connect the

> plurality of nodes;"
>
> [e.] "wherein the plurality of cluster node modules cooperate to interpret and translate, as needed, the instruction 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. 47–48 (quoting Ex. 1008, 3). Petitioner also relies on other teachings in Schreiner1 describing computer nodes with processors connected in a network. *See id.* (citing Ex. 1008, 23, 25; Ex. 1005 ¶ 110).

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. 48–51. 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 48–49. 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" using a load-balancing scheme. EX-1008, 316. 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, ¶116.

Id. at 51 (alteration by Petitioner).

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. 52 (citing Ex. 1008, 8; Ex. 1009, 5; Ex. 1005 ¶¶ 120–121). Petitioner also relies on a graphical output example for display in Schreiner1's user interface. *See id.* at 52–53 (reproducing Ex. 1008, Fig. 13).

Building on its showing with respect to claim 1, Petitioner's showing with respect to independent claims 8 and 10 somewhat tracks its showing with respect to independent claim 1. Pet. 53–62, 64–68. Petitioner supports its showing by citations to the record evidence, including the testimony of Dr. Tufo. *See id.* The same remarks apply to dependent claims 9 and 11. *See id.* at 62–64, 68–70.

Patent Owner does not address Petitioner's showing with respect to claims 8 and 9. Patent Owner addresses independent claim 10 and its dependent claim 11 by asserting that claim 10 precludes passing an instruction indirectly from a user interface, via a kernel, to a cluster node module. *See* Prelim. Resp. 25–29. Specifically, Patent Owner argues that "[t]he Petition does not show that the prior art discloses the required relative order of processing instructions" under its proposed claim construction. *Id.* at 25 (Limitations in claim 10 "should be construed to require the following relative order . . . front end → cluster node module → kernel.") As set forth above, however, the preliminary record does not support Patent Owner's narrow claim construction that precludes an intervening kernel between the front end and cluster node modules. *See supra* § II.C.

Based on the foregoing discussion, after considering other arguments and evidence presented by Patent Owner as addressed further below, the

Petition presents a sufficient showing supported by the record with respect to claims 1 and 8–11.

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. 20–23 (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, the Maple Guide, and other references, to *support* its showing based on the latter references. *See, e.g.*, Pet. 31 ("This is evident in the dist.maple code as well.").

Patent Owner does not directly argue that the other references do not support institution sufficiently without the Distributed Maple Code (i.e., source code). *See* Prelim. Resp. 21 ("The Board *may* deny the Petition on that basis alone.") (emphasis added). 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." *Id.* at 21 (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 Schreiner1 as referring to the source code and listing the noted website, and Patent Owner acknowledges that Schreiner1 lists the website. *See* Pet. 15 (citing

Ex. 1008, 5; Ex. 1006 ¶¶ 22–23; Ex. 1005 ¶ 45); Prelim. Resp. 21–22; Ex. 1008, 5–6. 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." Prelim. Resp. 23.

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."[9] Prelim. Resp. 21 (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.[10] 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.

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. 21–22), 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 publicly 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,

---

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

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

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) ("Extrinsic 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, the Maple Guide, PC Magazine1, and PC Magazine 2, teach claim 1 with or without the supporting source code as disclosed in the Distributed Maple Code.[11]

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*

[11] Patent Owner also contends that Petitioner's declarant impermissibly relies on "public use" information—i.e., information about commercial embodiments. *See* Prelim. Resp. 49–53; PO Sur-reply 10. Petitioner disagrees. Pet. Reply 9–10. For purposes of institution on this preliminary 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.)

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. 33–46. 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 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 [were] notoriously difficult, time-consuming, and expensive." Prelim. Resp. 32 (citing Ex. 2036 ¶¶ 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.* at 33. 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 34 (citing Ex. 2036 ¶¶ 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. 34–35. 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 35–36 (citing Ex. 2036 ¶¶ 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. 34–35. 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 36 (citing Ex. 2036 ¶¶ 24, 27–29; Ex. 2001 ¶¶ 65–70). Patent Owner argues this alleged claimed architecture obtained "superior performance" and "ease of use" that "were unexpected" and "surprising." *Id.* at 36–37. 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 38–39.

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. 40. 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 44. 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.* (citing Ex. 2036 ¶ 37). Patent Owner argues the "claimed architecture" is responsible for all of its asserted "objective indicia of nonobviousness." *Id.* (citing Ex. 2036 ¶¶ 21–31, 39–42, 46, 47).

Nevertheless, even if the SEM™ and SET™ products fall within the broad scope of the challenged claims, 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. 36. 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. 34–35, 37. As construed above, however, the challenged 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 challenged claims of the '877 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. 34. Patent Owner argues that its SEM and SET products met this

need. *Id.* at 35–37 (citing Ex. 2001 ¶¶ 65–70; Ex. 2007 ¶¶ 10, 12; Ex. 2008 ¶¶ 25–28; Ex. 2036 ¶¶ 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. 32; *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. 32–36. 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. 2036 ¶ 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. 36–38 (citing Ex. 2036 ¶¶ 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 '877 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. 2036 ¶ 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. 36–38 (citing Ex. 2036 ¶¶ 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. 2036 ¶¶ 38–41. Dr. Dauger is an inventor of the '877 patent, was Patent Owner's CTO, and is currently a consultant employed by Patent Owner. Ex. 1001, code (75); Ex. 2036 ¶ 1; Ex. 2015, 2 (SET presentation I); Ex. 2018, 2 (SET presentation II). On this preliminary record, Dr. Dauger's testimony about his personal surprise at the SEM and SET products that he helped create is 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. 2036 ¶ 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 '877 patent claims. Prelim. Resp. 38–39. 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 '877 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 '877 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. 39 (citing Ex. 2036 ¶¶ 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. 2036 ¶ 44 (citing Ex. 2018, 4); *see also* Ex. 2018, 2 (listing Dean E. Dauger, Ph.D. as the author). On this preliminary 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. 2036 ¶ 45 (citing Ex. 2023, 2). Initially, it is not clear what significance a "white paper" carries. Moreover, 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. 39–40. Regarding the SEM product, Patent Owner relies solely on the testimony of Dr. Bhansali and Mr. Bancroft. *Id.* at 39 (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. 2036 ¶ 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. 39 (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. 2036 ¶ 52; Ex. 2019, 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.[12]

Other reviewers similarly identify a lack of detail in the submissions. *See* 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 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

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

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. 45–46 (citing Ex. 2036 ¶ 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™."[13] *Id.* at 46 (citing Ex. 2036 ¶ 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

[13] 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. 2036 ¶¶ 56–57 (citing Ex. 2020, 4, Fig. 3).

that the inventors of the ’877 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 by Petitioner) (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 ’877 patent. On this record, Patent Owner’s conclusory assertions are inadequate to establish copying of the claimed invention by Petitioner.

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

d. Summary

Based on the foregoing discussion, Petitioner sufficiently establishes for purposes of institution that the combination of Schreiner1, Schreiner2, Schreiner3, the Maple Guide, PC Magazine1, and PC Magazine 2, with or without the Distributed Maple Code, renders claim 1 and 8–11 obvious.

Accordingly, Petitioner establishes a reasonable likelihood of prevailing with respect to claims 1 and 8–11.

E. Alleged Obviousness of Claims 1, 10, and 11 over Schreiner1, Schreiner2, Schreiner3, Maple Guide, Distributed Maple Code, PC Magazine, PC Magazine 2, and Nayak

1. Nayak (Ex. 1031)

Nayak teaches the use of digital signal processors (DSP) for executing MATLAB based software. Ex. 1031, 1 (Abstract), Fig. 4.

2. Alleged Obviousness for Claims 1, 10, and 11

Petitioner contends claims 1, 10, and 11 would have been obvious over the combination of Schreiner1, Schreiner2, Schreiner3, Maple Guide, Distributed Maple Code, PC Magazine, PC Magazine 2, and Nayak. *See* Pet. 71–76.

Petitioner relies on Nayak further to suggest using a special purpose microprocessor in Schreiner1's system. *See* Pet. 72–77. Petitioner contends that Nayak teaches "an MPI cluster-based approach for executing MATLAB mathematics software that is similar to Maple and Mathematica." *Id.* at 71. (citing Ex. 1031, 1 (Abstract)). Petitioner contends that "Nayak takes account of the fact that MATLAB, like Maple and Mathematica, is an interpreter, and teaches an approach that a POSITA would find obvious to apply to the Distributed Maple Publications in order to deploy Distributed Maple interpreter kernels on DSP cluster nodes." *Id.* at 75 (citing Ex. 1005 ¶ 163). Petitioner also contends that Nayak teaches advantages of implementing matrix operations on DSPs, and it would have been obvious to implement Schreiner1's ssiPlot matrix operation functions in the manner taught by Nayak." *Id.* at 74 (citing Ex. 1031, section 5; Ex. 1008, 32; Ex. 1005 ¶ 161). Petitioner provides record citations to evidence, including the testimony of Dr. Tufo, to support its showing. *See id.* at 72–77.

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, PC Magazine, PC Magazine 2, and Nayak renders obvious the subject matter of claims 1, 10, and 11. Other than presenting arguments as addressed

above with respect to claims 1 and 10, Patent Owner does not address this alternative showing by Petitioner of claims 1, 10, and 11 based on Nayak.

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 1, 10, and 11.

F. Alleged Obviousness of Claim 5 over Schreiner1, Schreiner2, Schreiner3, Maple Guide, Distributed Maple Code, PC Magazine, PC Magazine 2, Nayak, and Kelly

Petitioner adds Kelly to the references addressed in the previous section to address claim 5. *See* Pet. 76–77. Claim 5 depends from claim 1 and recites "wherein the special purpose microprocessor comprises multiple processor cores." Petitioner provides record citations to evidence, including the testimony of Dr. Tufo, to support its showing. *See id.*

Based on a review of the record, for purposes of institution, Petitioner sufficiently shows that the combination of Schreiner1, Schreiner2, Schreiner3, Maple Guide, Distributed Maple Code, PC Magazine, PC Magazine 2, Nayak, and Kelly renders obvious the subject matter of dependent claim 5. Patent Owner does not address claim 5 separately. Nonetheless, the burden remains on Petitioner to demonstrate unpatentability. *See Dynamic Drinkware,* 800 F.3d at 1378.

Accordingly, Petitioner establishes a reasonable likelihood of prevailing with respect to claim 5.

G. Alleged Obviousness of Claim 5 over Schreiner1, Schreiner2, Schreiner3, Maple Guide, Distributed Maple Code, PC Magazine, PC Magazine 2, the SPARC IV Article, and the AMD Article

Petitioner alternatively employs the SPARC IV Article and the AMD Article to the references employed against independent claim 1 to support its showing of alleged obviousness of dependent claim 5. *See* Pet. 77–80. Petitioner provides record citations to evidence, including the testimony of Dr. Tufo, to support its showing. *See id.*

Based on a review of the record, for purposes of institution, Petitioner sufficiently shows that the combination of Schreiner1, Schreiner2, Schreiner3, Maple Guide, Distributed Maple Code, PC Magazine, PC Magazine 2, the SPARC IV Article, and the AMD Article, renders obvious the subject matter of dependent claim 5. Patent Owner does not address claim 5 separately. Nonetheless, the burden remains on Petitioner to demonstrate unpatentability. *See Dynamic Drinkware,* 800 F.3d at 1378.

Accordingly, Petitioner establishes a reasonable likelihood of prevailing with respect to claim 5.

H. The Petition's Word Limit

Patent Owner alleges that the Petition "undercount[s] the true word count" by "about 900 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 22, 24, 52, and 55–58 of the Petition to avoid quoting text and counting the words." Prelim. Resp. 54 (citing *Starbucks Corp. v. Ameranth, Inc.*, CBM2015-00091, Paper 16 at 2–3 (PTAB Jan. 29, 2016)). It is not clear if by "900 excess words," Patent Owner implies that the total word count would be 14,900, or if it would be 900 words in excess of the certified word count of

13,882. *See id.* (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 reproduces a "screenshot of the Maple text interface" on a user interface in Schreiner1, but Petitioner simply could have cited the pages to make the same point. *See* Pet. 21 (reproducing Ex. 1008, 7–8). Petitioner again reproduces most of the same figure on page 24, but only to illustrate that the screenshot represents a portion of a user interface in Schreiner. The information carried in the text is not reproduced for what the text actually represents, other than to show it represents code "entered by the user after each '>' prompt." *See id.* at 24. Similar remarks apply to the screenshot on page 52 of the Petition. *Id.* at 52 ("Schreiner 1 shows a screenshot of a session in which the user inputs several instructions . . . ."). The reproduced snapshots on pages 55–57 simply show computer code and not much of it (i.e., about 9 or 10 "words" of code on each page except page 57, which shows about 20 "words" of code). Counting computer code as words that violate the Petition's word limits in each image of code, where Petitioner repeats the same images of code, is not reasonable in the context of Petitioner's showing here, because each "word" of code primarily carries meaning by association with the code words that surrounds it, much like an image.

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 all of its unpatentability challenges and the challenged claims. 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 ’877 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

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