

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

AMAZON.COM, INC. and
AMAZON WEB SERVICES, INC.,
Petitioner,

v.

SWARM TECHNOLOGY LLC,
Patent Owner.

IPR2022-00283
Patent 9,852,004 B2

Before MICHAEL R. ZECHER, GREGG I. ANDERSON, and
SCOTT B. HOWARD, *Administrative Patent Judges*.

HOWARD, *Administrative Patent Judge*.

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

INTRODUCTION

A. Background and Summary

Amazon.com, Inc. and Amazon Web Services, Inc. (collectively “Petitioner”) filed a Petition requesting *inter partes* review of claims 1–12 of U.S. Patent No. 9,852,004 B2 (Ex. 1001, “the ’004 patent”). Paper 2 (“Pet.”). Swarm Technology LLC (“Patent Owner”) filed a Preliminary Response. Paper 5 (“Prelim. Resp.”).

We have authority, acting on the designation of the Director, to determine whether to institute an *inter partes* review under 35 U.S.C. § 314 and 37 C.F.R. § 42.4(a). *Inter partes* review may not be instituted unless “the information presented in the petition filed under section 311 and any response filed under section 313 shows that there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition.” 35 U.S.C. § 314(a) (2018). “When instituting *inter partes* review, the Board will authorize the review to proceed on all of the challenged claims and on all grounds of unpatentability asserted for each claim.” 37 C.F.R. § 42.108(a) (2021).

For the reasons set forth below, upon considering the briefing and the evidence of record, we determine that the information presented in the Petition establishes a reasonable likelihood that Petitioner will prevail with respect to at least one of the challenged claims. Accordingly, we institute *inter partes* review on all of the challenged claims based on all of the grounds identified in the Petition.

B. Real Parties in Interest

Petitioner identifies Amazon.com, Inc. and Amazon Web Services, Inc. as the real parties in interest. Pet. 78.

Patent Owner identifies Swarm Technology LLC as the real party in interest. Paper 3, 1 (Patent Owner's Mandatory Notices).

C. Related Matters

The parties identify the following district court proceedings involving the '004 patent: (1) *Juniper Networks, Inc. v. Swarm Technology LLC*, No. 3:20-cv-03137-JD (N.D. Cal.) and (2) *Swarm Technology, LLC v Amazon.com, Inc.*, No. 2:21-cv-00438-DJH (D. Az.). Pet. 79; Paper 3, 1–2. The parties also identify the following pending *inter partes* review proceeding involving the '004 patent: *Juniper Networks, Inc. v. Swarm Technology LLC*, IPR2021-01445. Pet. 79; Paper 3, 2.

Additionally, the parties identify a number of patents and patent applications related to the '004 patent and *inter partes* review proceedings involving some of those patents. Pet. 79; Paper 3, 2–3.

D. The '004 Patent

The '004 patent is titled “System and Method for Parallel Processing Using Dynamically Configurable Proactive Co-Processing Cells” and is generally directed to “a processing architecture which involves autonomous co-processors configured to proactively retrieve tasks from a task pool populated by a central processing unit.” Ex. 1001, code (54), 1:14–18.

According to the '004 patent, “[c]omputer processors traditionally execute machine coded instructions serially. To run a plurality of applications concurrently, a single processor interleaves instructions from various programs and executes them serially, although from the user's perspective the applications appear to be processed in parallel.” Ex. 1001, 1:42–47. The '004 patent further states that “[t]rue parallel or multi-core processing, on the other hand, is a computational approach that breaks large computational tasks into individual blocks of computations and distributes

them among two or more processors.” *Id.* at 1:47–50. “A typical multiprocessor system includes a central processing unit (‘CPU’) and one or more co-processors. The CPU partitions the computational requirements into tasks and distributes the tasks to co-processors. Completed threads are reported to the CPU, which continues to distribute additional threads to the co-processors as needed.” *Id.* at 1:56–61.

The ’004 patent identifies a problem with using the CPU to control the distribution of tasks:

Presently known multiprocessing approaches are disadvantageous in that a significant amount of CPU bandwidth is consumed by task distribution; waiting for tasks to be completed before distributing new tasks (often with dependencies on previous tasks); responding to interrupts from co-processors when a task is completed; and responding to other messages from co-processors. In addition, co-processors often remain idle while waiting for a new task from the CPU.

Ex. 1001, 1:61–2:3. The ’004 patent addresses that problem using a system that reduces CPU management overhead and which “more effectively harnesses and exploits available co-processing resources.” *Id.* at 2:4–7.

Figure 1 of the '004 patent is reproduced below.

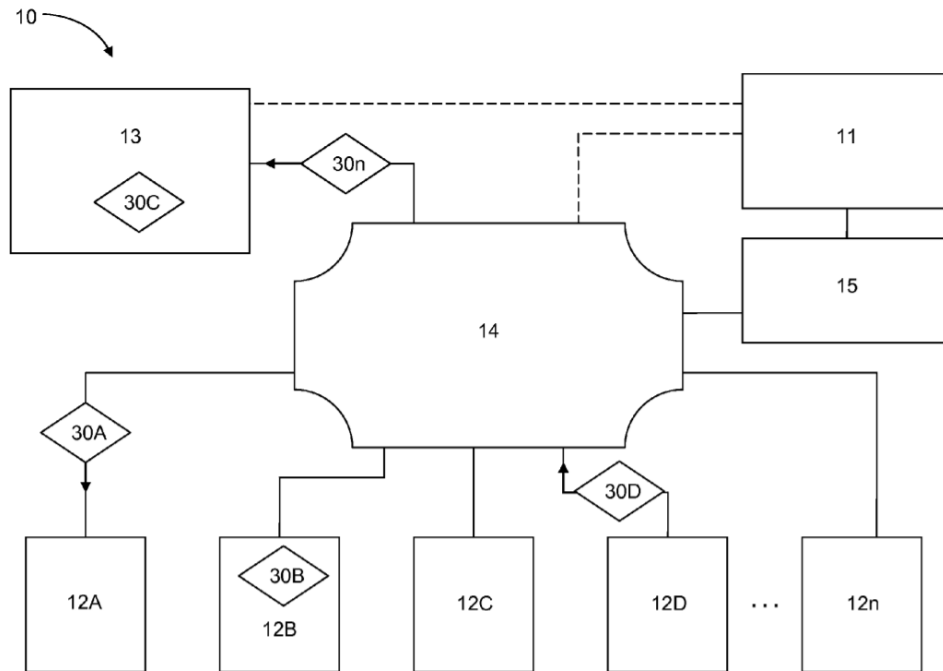


FIG. 1

Figure 1 “is a schematic block diagram of a parallel processing architecture including a CPU, memory, task pool, and a plurality of co-processors configured to communicate through a fabric.” Ex. 1001, 3:56–59. More specifically, Figure 1 shows “a single or multi-core CPU 11 and one or more solidarity or co-processing cells 12A-12[n] configured to communicate with a task pool 13 through a cross-bar switching fabric 14. The solidarity cells 12 may also communicate with each other through the switching fabric 14 or through a separate cell bus (not shown).” *Id.* at 4:30–36. “The CPU 11 may communicate with the task pool 13 directly or through the switching fabric 14 as shown. One or more memory units 15 each contain data and instructions” to perform computations. *Id.* at 4:36–39.

E. Illustrative Claim

Claims 1 and 3 are independent. Claim 1, reproduced below, is illustrative of the claimed invention.

1. A processing system, comprising:

a task pool;

a controller configured to populate the task pool with a plurality of first tasks and a plurality of second tasks;

a first co-processor configured to successively: retrieve a first task from the task pool; deliver the first task to the first co-processor; process the first task; generate first resulting data; and update the task pool to reflect completion of the first task, all without any communication between the first co-processor and the controller; and

a second co-processor configured to successively: retrieve a second task from the task pool; deliver the second task to the second co-processor; process the second task; generate second resulting data; and update the task pool to reflect completion of the second task, all without any communication between the second co-processor and the controller;

wherein the processing system is configured to dynamically accept the first co-processor, the second co-processor, and an additional co-processor into the processing system on a plug-and-play basis without any communication with the controller.

Ex. 1001, 14:10–32.

F. Prior Art and Asserted Grounds

Petitioner asserts that claims 1–12 would have been unpatentable based on the following grounds:

Claim(s) Challenged	35 U.S.C. § ¹	Reference(s)/Basis
1	103(a) ²	Lurie ³
2, 3, 6–11	103(a)	Lurie, Ethernet Specification ⁴
4, 5	103(a)	Lurie, Ethernet Specification, Sander ⁵
12	103(a)	Lurie, Ethernet Specification, Chow ⁶

Petitioner also relies on the testimony of Dr. David Lowenthal (Ex. 1003 (Declaration of Dr. Lowenthal)) and Dr. Ingrid Hsieh-Yee (Ex. 1005 (Declaration of Ingrid Shieh-Yee, Ph.D.)).

¹ The Leahy-Smith America Invents Act (“AIA”) included revisions to 35 U.S.C. §§ 102, 103 that became effective on March 16, 2013. Because the ’004 patent claims the benefit of the priority date of U.S. Patent Application No. 13/750,696, which was filed before March 16, 2013, and neither party has argued that the provisions of the AIA apply, we apply the pre-AIA versions of the statutory bases for unpatentability. *See* Ex. 1001, code (63).

² It is unclear why Petitioner identifies this ground under § 103(a) and not § 102(b). Petitioner argues that Lurie discloses each of the elements set forth in claim 1 and does not identify any differences between the prior art and the challenged claim. *See* Pet. 15–33. Nor does Petitioner set forth modifications a person having ordinary skill in the art would have made to Lurie. *See id.* The parties may wish to address whether Petitioner’s argument is, in fact, that claim 1 is anticipated as opposed to obvious and what impact, if any, that has on our analysis.

³ US 2007/0124363 A1, published May 31, 2007 (Ex. 1007).

⁴ The Ethernet, A Local Area Network, Data Link Layer and Physical Layer Specifications, dated Sept. 30, 1980 (Ex. 1009). All citations are to the native pagination.

⁵ US 2012/0192201 A1, published July 26, 2012 (Ex. 1010).

⁶ US 2005/0132380 A1, published June 16, 2005 (Ex. 1008).

ANALYSIS

A. *Legal Standards*

In *Graham v. John Deere Co. of Kansas City*, 383 U.S. 1 (1966), the Supreme Court set out a framework for assessing obviousness under 35 U.S.C. § 103 that requires consideration of four factors: (1) the “level of ordinary skill in the pertinent art,” (2) the “scope and content of the prior art,” (3) the “differences between the prior art and the claims at issue,” and (4) if in evidence, “secondary considerations” of non-obviousness such as “commercial success, long-felt but unsolved needs, failure of others, etc.” *Id.* at 17–18. “While the sequence of these questions might be reordered in any particular case,” *KSR Int’l Co. v. Teleflex Inc.*, 550 U.S. 398, 407 (2007), the U.S. Court of Appeals for the Federal Circuit has repeatedly emphasized that “it is error to reach a conclusion of obviousness until all those factors are considered,” *WBIP v. Kohler*, 829 F.3d 1317, 1328 (Fed. Cir. 2016).⁷

B. *Level of Ordinary Skill in the Art*

In determining whether an invention would have been obvious at the time it was made, we consider the level of ordinary skill in the pertinent art at the time of the invention. *Graham*, 383 U.S. at 17. “The importance of resolving the level of ordinary skill in the art lies in the necessity of maintaining objectivity in the obviousness inquiry.” *Ryko Mfg. Co. v. Nu-Star, Inc.*, 950 F.2d 714, 718 (Fed. Cir. 1991). The “person having ordinary skill in the art” is a hypothetical construct, from whose vantage

⁷ Because neither party addresses objective evidence of non-obviousness, we focus solely on the first three *Graham* factors.

point obviousness is assessed. *In re Rouffet*, 149 F.3d 1350, 1357 (Fed. Cir. 1998).

Factors pertinent to a determination of the level of ordinary skill in the art include “(1) the educational level of the inventor; (2) type of problems encountered in the art; (3) prior art solutions to those problems; (4) rapidity with which innovations are made; (5) sophistication of the technology; and (6) educational level of active workers in the field.” *Envtl. Designs, Ltd. v. Union Oil Co. of Cal.*, 713 F.2d 693, 696–697 (Fed. Cir. 1983) (citing *Orthopedic Equip. Co. v. All Orthopedic Appliances, Inc.*, 707 F.2d 1376, 1381–82 (Fed. Cir. 1983)). “Not all such factors may be present in every case, and one or more of these or other factors may predominate in a particular case.” *Id.*

Petitioner argues that a person having ordinary skill in the art “would have had a bachelor’s degree in computer science, computer engineering, or a related field, and about five years of industry or academic research experience in parallel or distributed computing, system architecture, or an equivalent.” Pet. 4 (citing Ex. 1003 ¶¶ 40–42). Petitioner further argues that “[a] post-graduate degree or additional formal education can compensate for less experience, and vice-versa.” *Id.* (citing Ex. 1003 ¶¶ 40–42).

Patent Owner does not address the level of skill in the art. *See* Prelim. Resp.

Accordingly, for purposes of this Decision, we adopt Petitioner’s proposed formulation of the level of ordinary skill in the art because it appears to be consistent with the ’004 patent and the asserted prior art.⁸

⁸ If Patent Owner proposes a different level of ordinary skill in the art in its Response, the parties are encouraged to address whether there are any

C. Claim Construction

We apply the same claim construction standard used in the federal courts, in other words, the claim construction standard that would be used to construe the claim in a civil action under 35 U.S.C. § 282(b), which is articulated in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc). *See* 37 C.F.R. § 42.100(b). Under the *Phillips* standard, the “words of a claim ‘are generally given their ordinary and customary meaning,’” which is “the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention, i.e., as of the effective filing date of the patent application.” *Phillips*, 415 F.3d at 1312–13.

Petitioner states that “[t]he prior art relied on in this Petition discloses the subject matter of the challenged claims under any reasonable construction, including their plain meaning.” Pet. 5.

Patent Owner does not address claim construction. *See* Prelim. Resp.

For purposes of this Decision, we need not expressly construe any claim terms. *See Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017) (noting that “we need only construe terms ‘that are in controversy, and only to the extent necessary to resolve the controversy’” (quoting *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999))).

D. Asserted Obviousness in View of Lurie

Petitioner argues that claim 1 would have been obvious over Lurie. *See* Pet. 14–33. Based on the current record, we are sufficiently persuaded

material differences between the two proposals and what impact, if any, the different level has on the obviousness analysis.

that Petitioner has established a reasonable likelihood of prevailing with respect to the sole challenged claim.

1. Summary of Lurie

Lurie is titled “Instrument-Based Distributed Computing System” and is directed “providing an instrument-based distributed computing system that accelerates the measurement, analysis, verification and validation of data in the distributed computing environment.” Ex. 1007, code (54), ¶ 2.

Lurie describes an “automatic distribution mode” of a distributed computing environment. Ex. 1007 ¶ 19. Lurie includes technical computing client 250 (“TCC”) which “can be a technical computing software application that provides a technical computing and graphical modeling environment for generating block diagram models and to define mathematical algorithms for simulating models.” *Id.* ¶ 55. It further includes a number of technical computing workers 270A-N (“TCWs”) which “can be a technical computing software application that provides a technical computing environment for performing technical computing of tasks, such as those tasks defined or created by the technical computing client 250.” *Id.* ¶ 58. In an embodiment, Lurie further describes Automatic Task Distribution Member 260 (“ATD Mechanism”) which can be “one or more application software components to provide for the automatic distribution of tasks from the [TCC] to the [TCWs].” *Id.* ¶ 61. This embodiment is shown in Figure 3B, which is reproduced below.

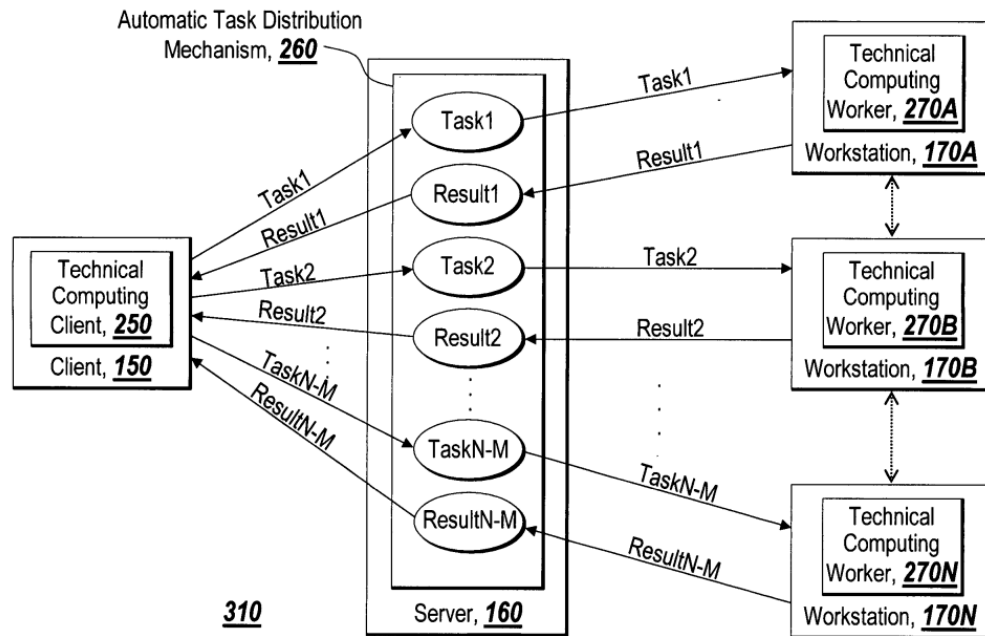


Fig. 3B

Figure 3B “is a block diagram of [Lurie’s] automatic distribution mode” embodiment which depicts (1) tasks and results flowing between the TCC and the ATD Mechanism and (2) tasks and results flowing between the ATD Mechanism and the TCWs. EX. 1007 ¶ 19, Fig. 3B. More specifically, the ATD Mechanism of the system “is intended to provide a configuration where the user does not want to control which [TCW] executes a particular task.” *Id.* ¶ 71. Instead, the TCC is in communication with the ATD Mechanism, which, in turn, is in communication with the various TCWs. *Id.* “Under this mode of operation, the [TCC] is not required to have any specific knowledge of the [TCWs], e.g., the name of the workstation running a [TCW], or the availability of the [TCW] to perform technical computing of a task.” *Id.* The TCC submits task to the ATD Mechanism, which, in turn, makes them available to the TCWs. *Id.* ¶ 72. “A [TCW] takes a task from the [ATD Mechanism] for technical computing of the task, computes a result

for the task and provides the result to the [ATD Mechanism].” *Id.* The TCC is then able to obtain the results from the ATD Mechanism. *Id.*

An alternative “batch mode” embodiment is illustrated in Figure 3C, which is reproduced below. Ex. 1007 ¶ 20.

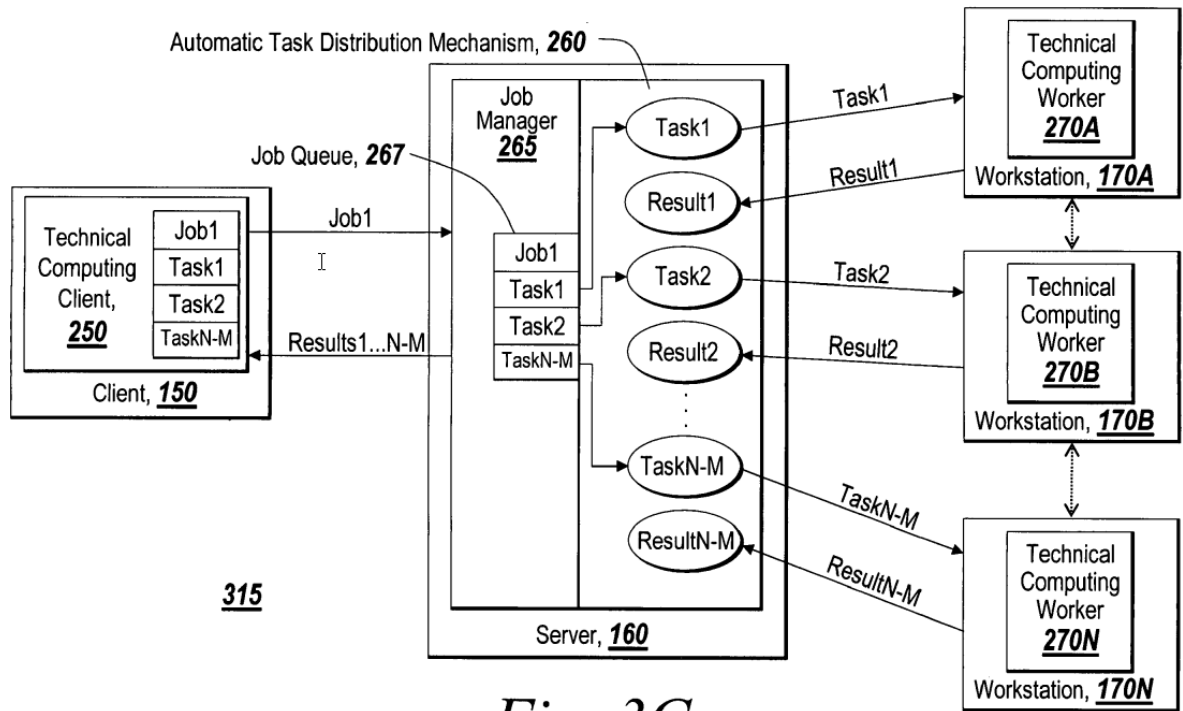


Fig. 3C

Figure 3C “is a block diagram of the batch automatic distribution mode of operation.” *Id.* The “batch mode . . . is intended to provide a configuration where the user can specify a group of related tasks as a job and provide the batch of tasks, or the job, to the [ATD Mechanism].” *Id.* ¶ 73. In this embodiment, the TCC “is in communication with the job manager 265 on the server 160” and “[t]he job manager 265 interfaces and communicates with the [ATD Mechanism] running on the same server 160.” *Id.* As with the Figure 3B embodiment, “[e]ach of the [TCWs] 270A-270N is in communication with the [ATD Mechanism].” *Id.*

In batch mode, “[i]nstead of submitting each task separately as depicted in FIG. 3B, the [TCC] submits the job containing all the associated

tasks to the job manager 265.” Ex. 1007 ¶ 74. “The job manager 265 obtains the tasks from the job and provides the tasks to the [ATD Mechanism] for [TCWs] 270A-270N to take and compute results.” *Id.* Lurie describes the operation of the components as follows:

For example, [TCC] defines a job, Job1, with a set of three tasks: Task1, Task2 and TaskN-M. The [TCC] then submits Job1 to the job manager 265. The job manager 265 obtains Job1 and obtains each of the tasks, Task1-TaskN-M from Job 1. Then, . . . the job manager 265 submits each of the tasks to the [ATD Mechanism] for technical computing by a [TCW]. [TCW] 270A may take Task1 from the [ATD Mechanism], compute a Result1 for Task1 and provide the Result1 to the [ATD Mechanism]. [TCW] 270B and [TCW] 270N, in a similar fashion, compute and provide results for Task2 and TaskN-M respectively. The job manager 265 then obtains the set of results for the completed job of Job1 and provides the results of each of the tasks to the [TCC].

Id. When there are multiple jobs, “the job manager 265 may handle jobs in a first-in first-out (FIFO) manner. In this case, the job manager 265 does not process the next job until all the tasks from the current job have been processed by the [ATD Mechanism].” *Id.* ¶ 75.

2. *Analysis of Claim 1*

a) *Undisputed Limitations*

The preamble of claim 1 recites “[a] processing system.” Ex. 1001, 14:10. Petitioner argues Lurie teaches the preamble. Pet. 15–16. Specifically, Petitioner argues Lurie teaches that “‘computing system’ (i.e.,

processing system) handles ‘tasks that are processed and/or managed collectively.’” *Id.* at 15 (citing Ex. 1007, code (57); Ex. 1003 ¶ 107).⁹

Claim 1 further recites “a task pool.” Ex. 1001, 14:11. Petitioner argues Lurie teaches this limitation. Pet. 16–20. Specifically, Petitioner argues Lurie’s ATD Mechanism “is a task pool because it is a data or memory structure configured to hold, store, monitor, populate with, and update tasks, and make them available to TCWs.” *Id.* at 15 (citing Ex. 1003 ¶¶ 111–115). Petitioner also argues that the ATD Mechanism “can be implemented using a technology known as Javaspaces, creating a ‘space’ or ‘object exchange repository’ in which tasks reside, i.e., a task pool.” *Id.* at 18 (citing Ex. 1003 ¶¶ 116–117).

Claim 1 further recites “wherein the processing system is configured to dynamically accept the first co-processor, the second co-processor, and an additional co-processor into the processing system on a plug-and-play basis without any communication with the controller.” Ex. 1001, 14:28–32. Petitioner argues that Lurie teaches this limitation. Pet. 31–33. Specifically, Petitioner argues that the Lurie has multiple TCWs and that Lurie “discloses dynamic acceptance of its TCWs (co-processors) without any communication with the TCC (controller).” *Id.* at 31 (citing Ex. 1003 ¶ 161). According to Petitioner, “Lurie explains that the TCC need not have

⁹ On the current record, there is no dispute that Lurie teaches a processing system. Accordingly, for purposes of institution, we need not decide whether the preamble is limiting. *See Nidec*, 868 F.3d at 1017 (noting that “we need only construe terms ‘that are in controversy’” (quoting *Vivid*, 200 F.3d at 803)).

If Patent Owner argues that Lurie does not teach a processing system in the Response, the parties must address whether the preamble is limiting.

any ‘specific knowledge’ of a TCW, including its ‘name’ or ‘availability,’ for a coprocessor to participate in the system.” *Id.* at 32 (quoting Ex. 1007 ¶ 71). Petitioner also argues that “Lurie’s TCWs can be activated using an activation daemon, to start and stop the TCW,” and that “[r]egistration with the activation daemon, and subsequent activation of the TCW, permits the TCW to be dynamically accepted at any time, and constitutes acceptance on a plug-and-play basis.” *Id.* (citing Ex. 1007 ¶ 108; Ex. 1003 ¶¶ 166–169).

After reviewing Petitioner’s arguments and evidence, including the Lowenthal Declaration (Ex. 1003), which Patent Owner does not address at this stage (*see* Prelim. Resp.), we are persuaded that Petitioner sufficiently demonstrates, for purposes of this Decision, that the Lurie teaches all of the undisputed limitations of claim 1 discussed in this subsection.

b) A Controller Configured to Populate the Task Pool with a Plurality of First Tasks and a Plurality of Second Tasks.”

(1) Petitioner’s Arguments

Claim 1 further recites “a controller configured to populate the task pool with a plurality of first tasks and a plurality of second tasks.” Ex. 1001, 14:12–13.

Petitioner argues that Lurie teaches that limitation. Pet. 21–24. Specifically, Petitioner argues that Lurie’s TCC is a controller that is “configured to populate the task pool (ATD Mechanism) with a plurality of first tasks and a plurality of second tasks.” *Id.* at 21 (citing Ex. 1003 ¶¶ 122–124). According to Petitioner, Lurie teaches that the TCC “‘submits one or more tasks . . . to the [ATD Mechanism],’ which ‘obtains the tasks . . . to make the[m] available for distribution to any of the [TCWs].’” *Id.* (quoting Ex. 1007 ¶ 72 and citing Ex. 1007, Fig. 3B; Ex. 1003 ¶ 125).

Petitioner sets forth two theories regarding the plurality of first tasks and plurality of second tasks. First, Petitioner argues that Lurie’s Task1 through TaskN-M can be subdivided in many different ways to create a plurality of first tasks and a plurality of second tasks. Pet. 22–23 (citing Ex. 1003 ¶¶ 127–129).

Second, Petitioner focuses on the batch mode. *See* Pet. 23–24. Petitioner argues that, in batch mode, the TCC populates the task pool with a plurality of jobs and that each job has a plurality of tasks:

In “batch mode,” a “group of related tasks” is collectively referred to as a “job,” such that “a plurality of first tasks” associated with one job (e.g., Task1 through TaskN-M associated with Job1 in Figure 3C), and “a plurality of second tasks” associated with another job, are populated in the task pool by the TCC. . . . In this mode, the TCC submits a job “containing all the associated tasks to the job manager,” and the job manager “obtains the tasks from the job and provides the tasks to the automatic task distribution mechanism 260 for technical computing workers 270A-270N to take and compute results.”

Id. at 23 (citing Ex. 1007 ¶¶ 73–74, Figs. 3B, 3C; Ex. 1003 ¶¶ 130–131).

Petitioner also argues that “[t]he [job] manager and the ATD Mechanism are capable of managing multiple jobs,” and that “[m]ultiple jobs would include a second job, consisting of a plurality of second tasks associated with that second job.” *Id.* at 24 (citing Ex. 1007 ¶ 75; Ex. 1003 ¶ 133).

(2) *Patent Owner’s Arguments*

Patent Owner disputes both of Petitioner’s theories. Prelim. Resp. 15–22. With regard to Petitioner’s first theory, Patent Owner argues that “Petitioner impermissibly modifies Lurie by arbitrarily sub-dividing Lurie’s separate tasks into two indiscriminate groups (separating Task1 and Task2 on the one hand, and Task3 to TaskN-M on the other).” Prelim. Resp. 16. Patent Owner also argues that the testimony of Dr. Lowenthal does not have

any support and, instead, “improperly substitutes Lowenthal’s own hindsight reconstruction for Lurie’s disclosure.” *Id.* at 17. According to Patent Owner,

Dr. Lowenthal’s declaration violates [United States Court of Appeals for the] Federal Circuit case law and the PTAB’s published position that “in an obviousness analysis, conclusory assertions from a third party about general knowledge in the art cannot, without supporting evidence of record, supply a limitation that is not evidently and indisputably within the common knowledge of those skilled in the art.”

Id. (quoting *K/S Himpp v. Hear-Wear Techs., LLC*, 751 F.3d 1362,1365 (Fed. Cir. 2014)).

With regard to Petitioner’s second theory, Patent Owner argues that, contrary to Petitioner’s argument, Lurie does not teach that the TCC—which Petitioner maps to the controller—populates task to the ATD Mechanism. Prelim. Resp. 19. According to Patent Owner, in batch mode, the TCC is not in communication with the ATD Mechanism; instead, “Lurie teaches that the TCC is in communication with a separate and distinct module called the ‘job manager.’” *Id.* at 19–20 (citing Ex. 1007 ¶ 73).

Patent Owner further argues that, even in batch mode, Lurie does not teach “a task pool ‘populated with a plurality of first tasks and a plurality of second tasks.’” Prelim. Resp. 21–22. According to Patent Owner, Lurie teaches that the ATD Mechanism is only populated with the tasks associated with a single job at a time. *Id.* (citing Ex. 1007 ¶ 75).

(3) *Our Analysis*

Based on the current record, Petitioner has sufficiently shown that Lurie teaches that the TCC (controller) populates the ATD Mechanism (task pool) with a plurality of first tasks and a plurality of second tasks. As shown in Lurie Figure 3B, the TCC populates the ATD Mechanism with various

tasks Task1 through TaskN-M. Ex. 1007 Fig. 3B; Ex. 1003 ¶¶ 127–128. Petitioner has sufficiently shown that a person having ordinary skill in the art would have understood that the tasks could be divided in various ways, including “a plurality of first tasks (*e.g.*, Task1 and Task2) and a plurality of second tasks (*e.g.*, Task3 and Task4, or TaskN and TaskM).” Ex. 1003 ¶ 128. Although we agree with Patent Owner that this division is arbitrary, Patent Owner has not proposed a construction of first tasks or second tasks that would preclude such an arbitrary grouping.

We do not agree with Patent Owner’s argument that Petitioner is improperly relying on general knowledge or that *K/S Himpp* applies. *See* Prelim. Resp. 17. To the contrary, Dr. Lowenthal is simply providing testimony regarding Lurie’s teachings from the perspective of a person of ordinary skill in the art. *See* Ex. 1003 ¶ 128. This is not an improper use of expert testimony. *See Sundance, Inc. v. DeMonte Fabricating Ltd.*, 550 F.3d 1356, 1361 n.3 (Fed. Cir. 2008) (“What a prior art reference discloses or teaches is determined from the perspective of one of ordinary skill in the art.”); Consolidated Trial Practice Guide (Nov. 2019) at 35 (“Experts may testify about the teachings of the prior art and how they relate to the patentability of the challenged claims.”); *see also id.* at 36 (“Expert testimony may be presented to establish the scope and content of the prior art for determining obviousness and anticipation.”).

Accordingly, based on the current record, Petitioner sufficiently shows that Lurie teaches “a controller configured to populate the task pool with a plurality of first tasks and a plurality of second tasks,” as recited in claim 1.

Having found that Petitioner presented sufficient evidence regarding its first theory, we need not address its second theory. However, we do so in order to provide guidance to the parties for the trial.

Based on the current record, we do not agree with Patent Owner's argument that Lurie does not teach the "controller" limitation because the ATD Mechanism only has one plurality of tasks at a time. Although Patent Owner accurately describes how Lurie operates (*see* Ex. 1007 ¶ 75), it is inapposite that the job manager does not process the second job and its second plurality of tasks until after the first job and its first plurality of tasks have been completed. There is nothing in claim 1 that requires both the plurality of first tasks and the plurality of second tasks to be in the task pool at the same time; instead, claim 1 does not have any temporal component and simply recites "a controller configured to populate the task pool with a plurality of first tasks and a plurality of second tasks." Ex. 1001, 14:12–13.

However, based on the current record, we question whether Petitioner has shown Lurie teaches the "controller" limitation under its second theory. Specifically, although Petitioner identifies the TCC as the controller, based on Lurie's teachings it appears that the job manager populates the ATD Mechanism with tasks, not the TCC. *See* Ex. 1007 ¶ 74 ("Instead of submitting each task separately as depicted in FIG. 3B, the technical computing client 250 submits the job containing all the associated tasks to the job manager 265. . . . The job manager 265 obtains the tasks from the job and provides the tasks to the automatic task distribution mechanism 260 for technical computing workers 270A-270N to take and compute results."), Fig. 3C (showing job manager 265 populating the ATD Mechanism). Because the TCC does not populate the ATD Mechanism with tasks, it is unclear how the TCC satisfies this disputed limitation.

c) First Co-processor

(1) The Parties' Arguments

Claim 1 further recites “a first co-processor configured to successively: retrieve a first task from the task pool; deliver the first task to the first co-processor; process the first task; generate first resulting data; and update the task pool to reflect completion of the first task, all without any communication between the first co-processor and the controller.” Ex. 1001, 14:14–19.

Petitioner argues that Lurie teaches this limitation. Pet. 24–29. Specifically, Petitioner argues that one of Lurie’s TCWs is a first co-processor which “‘takes a task from the [ATD Mechanism]’ and delivers the task to where it will be processed by the co-processor (retrieve and deliver the first task).” *Id.* at 24–25 (quoting Ex. 1007 ¶ 72) (citing Ex. 1003 ¶ 136). Petitioner also argues that the “TCW (co-processor) then ‘computes a result for the task’ (processing the task and generating resulting data)” to “‘generate[] a result (first resulting data).” *Id.* at 25 (quoting Ex. 1007 ¶ 72 and citing Ex. 1003 ¶¶ 139–140). Petitioner also argues that the TCW “‘then ‘provides the result [back] to the [ATD Mechanism],’ thereby updating the ATD Mechanism (task pool) to reflect completion, as indicated by the result.” *Id.* (quoting Ex. 1007 ¶ 72 and citing Ex. 1003 ¶¶ 141–143). According to Petitioner, because the ATD Mechanism “makes the results . . . available to the [TCC],” “the TCC does not communicate with the TCWs.” *Id.* (quoting Ex. 1007 ¶ 72 and citing Ex. 1007 ¶¶ 61, 72, Figs. 3B, 6A; Ex. 1003 ¶¶ 144–145); *see also id.* at 27–29 (expanding upon argument). In discussing the ATD Mechanism, Petitioner directs us to Lurie Figure 6A, which shows Object Exchange Repository 662. *Id.* at 26.

Patent Owner presents two arguments challenging Petitioner's showing. First, Patent Owner argues that Lurie teaches that the ATD Mechanism distributes the tasks to the TCWs, thus "affirmatively teach[ing] away from a co-processor configured to autonomously and proactively 'retrieve' a task." Prelim. Resp. 23 (citing Ex. 1007 ¶¶ 6, 43–44).

Second, Patent Owner argues the "Lurie's Object Exchange Repository embodiment is substantively the same as the conventional 'controller/responder' paradigm that the '004 Patent expressly overcomes." Prelim. Resp. 25; *see also id.* at 24–25 (discussing Lurie's Exchange Repository embodiment). According to Patent Owner, "a TCW must register with the Object Exchange Repository and only retrieves a task in response to a notification from the Object Exchange Repository. Lurie's TCW is *not* proactive or autonomous like the co-processors of the '004 Patent." *Id.* at 25.

(2) *Our Analysis*

Based on the current record, Petitioner has sufficiently shown that Lurie teaches or suggests the recited "first co-processor." Specifically, Lurie teaches how a TCW "takes a task from the [ATD Mechanism] 260 for technical computing of the task, computes a result for the task and provides the result to the [ATD Mechanism] 260." Ex. 1007 ¶ 72. A person having ordinary skill in the art would have realized "that the task is then delivered to the first co-processor after the task is retrieved in order for the co-processor subsequently to process the task." Ex. 1003 ¶ 138. A person having ordinary skill in the art would also have realized that providing the results back to the ATD Mechanism teaches "updating the task pool to reflect completion of the first task." *Id.* ¶ 141. Additionally, Lurie shows

that the TCWs communicate only with the ATD Mechanism and not the TCC. *See id.* Figs. 3B, 5B, 6A, ¶¶ 61, 72, 94; *see also* Ex. 1003 ¶¶ 144–151.

Based on the current record, we do not agree with Patent Owner’s arguments. With regard to Patent Owner’s first argument, it is true, as Patent Owner argues, that in some places Lurie describes “distributing” the tasks to the processors. *See* Ex. 1007 ¶¶ 6 (“The method further includes the step of *distributing* the job to remote computing workers for the distributed execution of the job. (emphasis added)), 43 (“The task can be *directly distributed* by the technical computing client to one or more technical computing workers.” (emphasis added)), 44 (“Additionally, a task or a group of tasks, in a job, can be submitted to an automatic task distribution mechanism to *distribute* the one or more tasks automatically to one or more technical computing workers providing technical computing services.” (emphasis added)). However, Lurie also teaches other embodiments in which the TCW takes the tasks. *See* Ex. 1007 ¶ 72 (“A technical computing worker 270A-270N *takes* a task from the automatic task distribution mechanism 260 for technical computing of the task. . . .” (emphasis added)). Accordingly, based on the current record, Lurie appears to describe two different embodiments, one in which the TCWs take tasks (which Petitioner relies upon) and another in which the tasks are distributed. The embodiment which Petitioner relies on is sufficient for purposes of institution.

Moreover, even if the ATD mechanism distributes the tasks to the various TCWs, that is inapposite. Patent Owner has not pointed to anything in the claims that require the first co-processor to select which tasks it will perform. Instead, the claim language simply requires that the first co-processor “retrieve a first task from the task pool.” Ex. 1001, 14:14–19. Patent Owner has not pointed to any claim language that precludes the ATD

Mechanism from distributing/assigning the tasks to the various TCWs so long as the TCW retrieves the task.

Patent Owner's second argument is similarly inapposite. Patent Owner has not identified any claim language that requires the first co-processor to be pro-active or autonomous. Instead, as discussed in the preceding paragraph, claim 1 is silent on how the tasks are assigned.

d) Second Co-processor

(1) The Parties' Arguments

Claim 1 recites “a second co-processor configured to successively: retrieve a second task from the task pool; deliver the second task to the second coprocessor; process the second task; generate second resulting data; and update the task pool to reflect completion of the second task, all without any communication between the second co-processor and the controller.”

Ex. 1001, 14:21–27.

Petitioner argues that Lurie teaches that limitation. Pet. 29–31. Specifically, Petitioner argues that “Lurie discloses multiple TCWs (coprocessors), as shown in Figure 3B.” *Id.* at 29 (citing Ex. 1007 Fig. 3B, ¶¶ 71, 72; Ex. 1003 ¶ 154). Petitioner further argues that for the same reason Lurie teaches

“a *first* co-processor” (first TCW) configured to retrieve a *first* task from the task pool, deliver the *first* task to the co-processor, process the *first* task, generate *first* resulting data, and update the task pool to reflect completion of the *first* task for element [1c], Lurie also discloses a “*second* co-processor” (second TCW) configured to retrieve a *second* task from the task pool, deliver the *second* task to the co-processor, process the *second* task, generate *second* resulting data, and update the task pool to reflect completion of the *second* task.

Id. at 30–31 (citing Pet. § V.A.1.c; Ex. 1003 ¶¶ 156–158).

Patent Owner disputes Petitioner’s characterization. According to Patent Owner, Petitioner relies on Lurie’s Object Exchange Repository embodiment shown in Figure 6A, which only has a single TCW. Prelim. Resp. 26–27. Patent Owner further argues that Petitioner combines two different embodiments (Figure 6A and Figure 3B), without providing a reason why a person having ordinary skill in the art would have made that modification. *Id.* at 27–28.

(2) *Our Analysis*

Based on the current record, Lurie teaches using multiple TCWs. *E.g.*, Ex. 1007 Fig. 3B. Petitioner distinctly identifies a first TCW as the first processor and a second TCW as the second co-processor. Pet. 24 n.7. For the same reasons discussed in the proceeding section (*see supra* § D.2.c.2), we find, based on the current record, that Lurie’s second TCW meets the requirements of this limitation.

We do not agree with either of Patent Owner’s arguments at this time. First, Patent Owner reads too much into Figure 6A. Figure 6A is designed to highlight an aspect of the invention in which the ATD Mechanism uses an object exchange repository. Ex. 1007 ¶ 90. That is, according to Lurie, Figure 6A “is a block diagram depicting the details of the [ATD Mechanism].” *Id.* ¶ 26. Patent Owner does not, at the present time, point to any part of Lurie that indicates it was designed to be used with only a single TCW as opposed to multiple TCWs in a distributed computing system. To the contrary, Lurie describes an example in which multiple TCWs are used with the object exchange repository. *See id.* ¶ 92.

Second, although Lurie describes Figure 6A as one embodiment, looking at Lurie as a whole, it is more accurate, based on the current record, to describe it as an aspect of the invention that could be used with the

embodiments that have an ATD Mechanism. *See* Ex. 1007 ¶ 90 (“The automatic task distribution mechanism 260 may be object-oriented and comprise an object exchange repository 662, such as Javaspace, a Sun Microsystems, Inc. technology for distributed application development built using Jini network technology also from Sun Microsystems, Inc.”). That is, Figure 6A is an aspect of the Figure 3B embodiment—which uses the ATD Mechanism—but is different from the Figure 3A embodiment—in which the TCC communicates directly with the TCWs. *See id.* ¶ 92 (describing how the object exchange repository can be used as an intermediary between the TCC and the TCWs). Although we agree with Patent Owner that a party challenging the patentability of a claim based on a single reference needs to provide a reason to combine distinct embodiments (*see* Prelim. Resp. 27–28), that does not apply to using different aspects of the same embodiment.

e) Conclusion Regarding Claim 1

For the reasons set forth above, Petitioner has demonstrated, on this record, a reasonable likelihood of prevailing on its assertion that claim 1 would have been obvious over Lurie.

E. Asserted Obviousness in View of Lurie and Ethernet Specification

1. Ethernet Specification

Ethernet Specification “contains the specification of the Ethernet, a local area network.” Ex. 1009, i. Ethernet Specification “is intended as a design reference document, rather than an introduction or tutorial.” *Id.*

Relevant to Petitioner’s argument, chapter 6 provides the “Ethernet Data Link Layer Specification.” Ex. 1009, 19–44. Figure 6-1 is reproduced below.

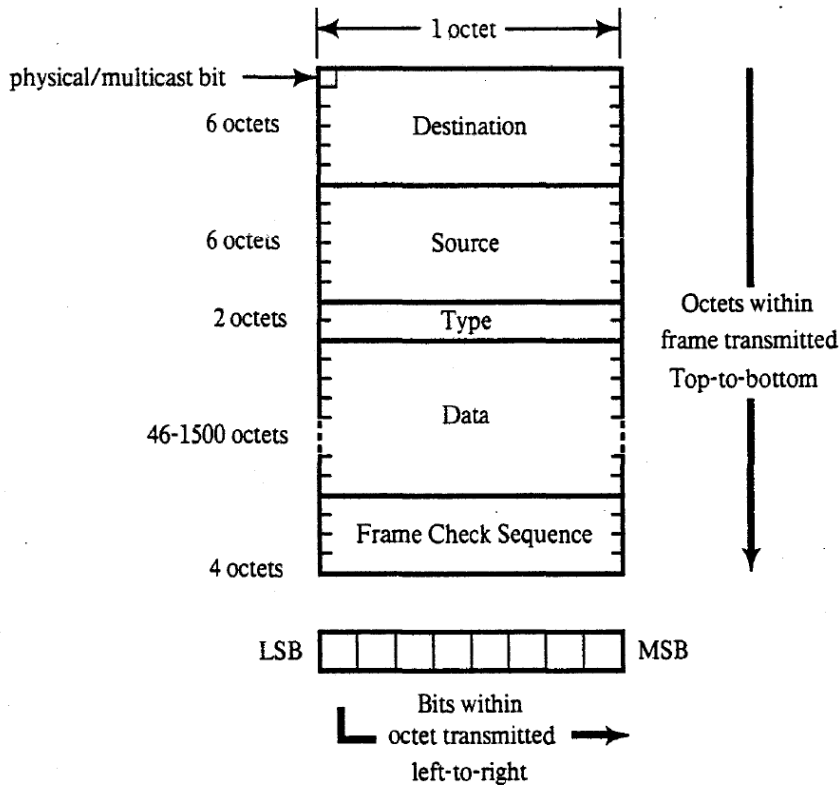


Figure 6-1: Data Link Layer Frame Format

Ex. 1009, 20. Figure 6-1 depicts a “Data Link Frame Format” which includes a destination, source, type, data, and frame check sequence. *Id.*

2. *Claims 2, 3, and 6–11*

Because Petitioner has demonstrated a reasonable likelihood of success in proving that at least one claim of the ’004 patent is unpatentable, we institute on all grounds and all claims raised in the Petition. *See* 37 C.F.R. 42.108(a). Therefore, at this stage of the proceeding, it is not necessary for us to provide an assessment of every challenge in this ground, especially as Patent Owner has not presented any responsive arguments beyond those addressed above for claim 1.

Nevertheless, we note that Petitioner provides detailed explanations supported by the testimony of Dr. Lowenthal and specific citations to the relevant references indicating where in the reference Petitioner argues the

limitations of claims 2, 3, and 6–11 are taught and why a person of ordinary skill in the art would have combined the various teachings of Lurie and Ethernet Specification. *See* Pet. 33–63. Accordingly, at this stage of the proceeding, we are persuaded the information presented in the Petition establishes there is a reasonable likelihood that Petitioner would prevail on its assertion that claims 2, 3, and 6–11 would have been obvious over Lurie and Ethernet Specification.

F. Asserted Obviousness in View of Lurie, Ethernet Specification, and Sander

1. Sander

Sander is entitled “Dynamic Work Partitioning on Heterogeneous Processing Devices” and is directed to “an architecture for unifying the computational components within a computer system.” Ex. 1010, code (54), ¶ 3.

Sander describes various elements of a computer system including a CPU (central processing unit) and an APD (accelerated processing unit). Ex. 1010 ¶¶ 5, 12–13. According to Sander, both the CPU and the APD can be “formed on a single silicon die or package, combining CPU 102 and APD 104 to provide a unified programming and execution environment.” *Id.* ¶ 28.

2. Analysis of Claims 3 and 4

As discussed above, because Petitioner has demonstrated a reasonable likelihood of success in proving that at least one claim of the ’004 patent is unpatentable, we institute on all grounds and all claims raised in the Petition. Therefore, at this stage of the proceeding, it is not necessary for us to provide an assessment of every challenge raised by Petitioner, especially as

Patent Owner has not presented any responsive arguments directed to the additional limitations set forth in claims 3 and 4. *See* Prelim. Resp.

Nevertheless, we note that Petitioner provides detailed explanations supported by the testimony of Dr. Lowenthal and specific citations to the relevant references indicating where in the reference Petitioner argues the limitations of claims 3 and 4 are taught and why a person of ordinary skill in the art would have combined the various teachings of Lurie, Ethernet Specification, and Sander. *See* Pet. 63–69. Accordingly, at this stage of the proceeding, we are persuaded the information presented in the Petition establishes there is a reasonable likelihood that Petitioner would prevail on its assertion that claims 3 and 4 would have been obvious over Lurie, Ethernet Specification, and Sander.

G. Asserted Obviousness in View of Lurie, Ethernet Specification, and Chow

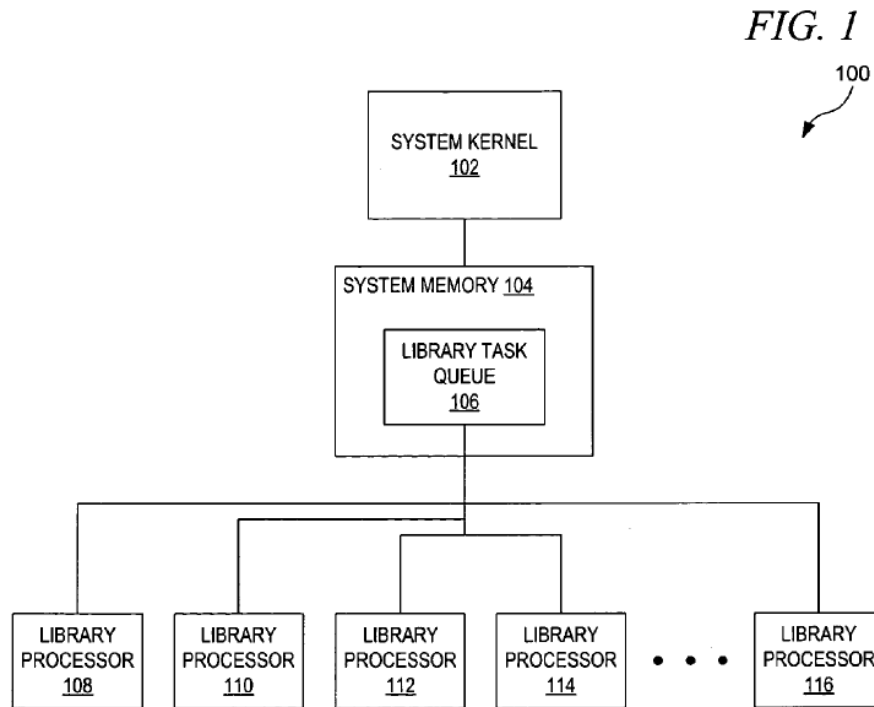
1. Chow

Chow is entitled “Method for Hiding Latency in a Task-Based Library Framework for a Multiprocessor Environment” and is directed to “a task-based library framework for dynamic load balancing in a multiprocessor environment, and to a method of latency hiding in this framework.”

Ex. 1008, code (54), ¶ 1).

Chow recites “[a] task-based library framework for load balancing using a system task queue in a tightly-coupled multiprocessor system. The system memory holds a queue of system tasks. The library processors fetch tasks from the queue for execution.” Ex. 1008, code (57). Chow further recites that “[t]he library processors fetch the tasks when they have a light load. A library processor can fetch a task while executing another task.” *Id.*

Chow Figure 1 is reproduced below.



Chow Figure 1 “schematically depicts a tightly-couple multiprocessor system with a task-based library framework and library processors.” Ex. 1008 ¶ 7.

Referring to FIG. 1 of the drawings, the reference numeral 100 generally designates a tightly-coupled multiprocessor system with a task-based library framework. The system 100 comprises a system kernel 102, a system memory 104, and a number of library processors, 108, 110, 112, 114, and 116. The ellipsis indicates that the system 100 can comprise additional library processors. The system memory comprises a queue of tasks 106 to be assigned to the library processors (library task queue 106). Each library processor has access to the library task queue 106. When tasks arrive at the system kernel 102 for processing, they are subdivided into subtasks and placed into the library task queue 106. The library processors 108, 110, 112, 114, and 116 fetch the subtasks from the library task queue 106.

Id. ¶ 13.

2. *Analysis of Claim 12*

As discussed above, because Petitioner has demonstrated a reasonable likelihood of success in proving that at least one claim of the '004 patent is unpatentable, we institute on all grounds and all claims raised in the Petition. Therefore, at this stage of the proceeding, it is not necessary for us to provide an assessment of every challenge raised by Petitioner, especially as Patent Owner has not presented any additional arguments not previously discussed. *See* Prelim. Resp.

Nevertheless, we note that Petitioner provides detailed explanations supported by the testimony of Dr. Lowenthal and specific citations to the relevant references indicating where in the reference Petitioner argues the limitations of claim 12 are taught and why a person of ordinary skill in the art would have combined the various teachings of Lurie, Ethernet Specification, and Chow. *See* Pet. 69–75. Accordingly, at this stage of the proceeding, we are persuaded the information presented in the Petition establishes there is a reasonable likelihood that Petitioner would prevail on its assertion that claim 12 would have been obvious over Lurie, Ethernet Specification, and Chow.

CONCLUSION

Following § 314, we have determined that the totality of the information presented at this stage shows there is a reasonable likelihood that Petitioner would prevail with respect to at least one of the claims challenged in the Petition, and we institute on all grounds raised and all claims challenged in the Petition.

Our factual findings, conclusions of law, and determinations at this stage of the proceeding are preliminary, and based on the evidentiary record developed thus far. This is not a final decision as to the patentability of

claims for which IPR is instituted. Our final decision will be based on the record as fully developed during trial.

ORDER

In consideration of the foregoing, it is hereby:

ORDERED that, pursuant to 35 U.S.C. § 314(a), an *inter partes* review of all challenged claims of the '004 patent is instituted with respect to all grounds set forth in the Petition; and

FURTHER ORDERED that, 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 commencing on the entry date of this decision.

IPR2022-00283
Patent 9,852,004 B2

FOR PETITIONER:

Jonathan Strang
Adam Greenfield
David Zucker
Kimberly Li
LATHAM & WATKINS LLP
janathan.strang@lw.com
adam.greenfield@lw.com
david.zucker@lw.com
kimberly.li@lw.com

FOR PATENT OWNER:

Daniel Pote
JENNINGS STROUSS & SALMON PLC
p@jsslaw.com

Michael Kelly
BEUS GILBERT MCGRODER PLLC
mkelly@beusgilbert.com