

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.

IPR2023-00390
Patent 9,146,777 B2

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

HOWARD, *Administrative Patent Judge*.

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

I. INTRODUCTION

A. *Background and Summary*

Amazon.com, Inc. and Amazon Web Services, Inc. (collectively “Petitioner”) filed a Petition requesting *inter partes* review of claims 1–14 of U.S. Patent No. 9,146,777 B2 (Ex. 1001, “the ’777 patent”). Paper 2 (“Pet.”). Swarm Technology LLC (“Patent Owner”) filed a Preliminary Response. Paper 6 (“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) (2022).

B. *Real Parties in Interest*

The parties identify themselves as the real parties in interest. Pet. 51–52; Paper 4, 1 (Patent Owner’s Mandatory Notices).

C. *Related Matters*

The parties identify the following related district court proceedings: *Juniper Networks, Inc. v. Swarm Technology LLC*, No. 3:20-cv-03137 (N.D. Cal.) and *Swarm Technology, LLC v. Amazon.com, Inc.*, No. 2:21-cv-00438 (D. Ariz.). Pet. 52; Paper 4, 1–2.

The parties identify the following PTAB proceedings involving the ’777 patent and related patents: *Juniper Networks, Inc. v. Swarm*

Technology LLC, IPR2021-01317 (the '777 patent); *Juniper Networks, Inc. v. Swarm Technology, LLC*, IPR2021-01445 (U.S. Patent No. 9,852,004 B2 (“the '004 patent”)); *Juniper Networks, Inc. v. Swarm Technology, LLC*, IPR2022-00141 (U.S. Patent No. 10,592,275 B2 (“the '275 patent”)); *Amazon.com, Inc. v. Swarm Technology, LLC*, IPR2022-00283 (the '004 patent); and *Amazon.com, Inc. v. Swarm Technology, LLC*, IPR (2022-00633) (the '275 patent). Pet. 52–53; Paper 4, 1–2.

The parties also identify the following related patents and patent applications: the '275 patent; the '004 patent; and U.S. Patent Application No. 16/788,122 filed February 11, 2020. Pet. 52; Paper 4, 2–3.

D. The '007 Patent

The '777 patent is titled “Parallel Processing with Solidarity Cells by Proactively Retrieving from a Task Pool a Matching Task for the Solidarity Cell to Process” and is generally directed to “a method and apparatus that facilitates parallel processing of computing tasks using autonomous co-processors that proactively seek tasks to perform.” Ex. 1001, code (54), 1:9–13.

According to the '777 patent, “[p]arallel processing is a computational approach that breaks a large computational requirement into a group of smaller computations for concurrent resolution across the computer’s processing resources.” Ex. 1001, 1:17–20. A central processing unit (“CPU”) “divides the large requirement into tasks that contain segments of executable code” which it then distributes to co-processors. Ex. 1001, 1:20–28. “Completed threads are reported to the CPU, which continues to distribute threads until the primary processing is complete.” Ex. 1001, 1:28–30.

The '777 patent identifies two major drawback associated with that typical multiprocessing framework. Ex. 1001, 1:31–45. “First, a significant amount of the CPU’s processing time is consumed in managing the co-processing tasks.” Ex. 1001, 1:33–35. “Second, a co-processor will remain idle as it waits for a thread to be assigned to it by the CPU.” Ex. 1001, 1:41–42. The '777 patent addresses those problems by using autonomous co-processors that proactively searches a task pool to acquire tasks for the co-processor to perform. Ex. 1001, 1:46–55.

Figure 1 of the '777 patent is reproduced below.

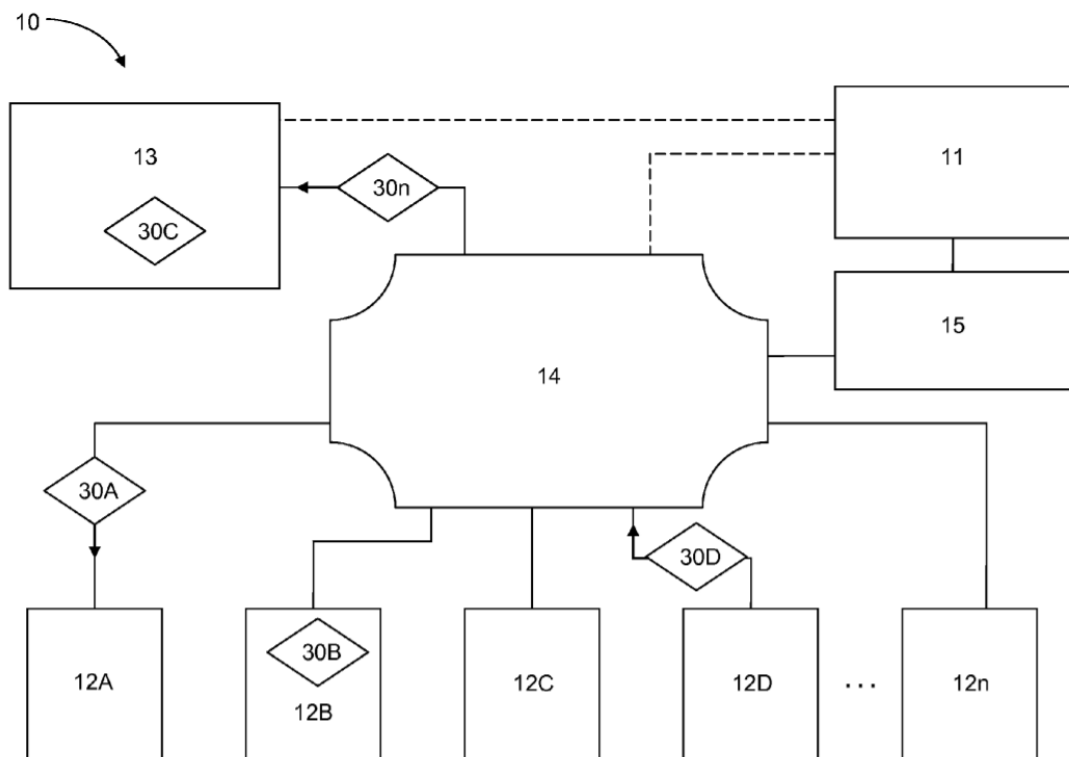


FIG. 1

Figure 1 “is illustrated the preferred embodiment of a parallel processing system 10 for performing computing tasks, such as data-intensive computations and complex programs and algorithms.” Ex. 1001, 2:29–32. “The system 10 has at least one CPU 11 and one or more solidarity

cells 12A, B, C, D . . . n, hereinafter referred to collectively as solidarity cells 12A . . . n to indicate that the system 10 may have any number n of solidarity cells, that cooperate to perform the computing tasks ascribed to the system 10.” Ex. 1001, 2:47–52. “The solidarity cells 12A . . . n communicate with a task pool 13, preferably through a switching fabric 14. The solidarity cells 12A . . . n may further communicate with each other through the switching fabric 14 or through a separate cell bus (not shown).” Ex. 1001, 2:52–56. “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 contain data and instructions to perform computations.” Ex. 1001, 2:56–59.

E. Illustrative Claims

Claims 1 and 14 are independent, reproduced below, and illustrative of the claimed invention.

1. An apparatus for parallel processing of a large computing requirement, the apparatus comprising:

a central processing unit (“CPU”);

a task pool in electronic communication with the CPU; and

a first solidarity cell in electronic communication with the task pool, *the first solidarity cell comprising a first agent configured to proactively retrieve, from the task pool, without requiring an instruction from the CPU, a matching task for the solidarity cell to process;*

wherein the CPU populates the task pool by dividing the requirement into one or more threads and placing the threads in the task pool, each thread comprising one or more tasks, and the matching task being one of the tasks;

wherein each task comprises a descriptor, the descriptor containing at least:

a function to be executed; and

a memory location of data upon which the function is to be executed;

wherein the first agent is a data frame comprising: a source address, a destination address and a payload;

wherein the first agent retrieves the matching task by:

being dispatched by the first solidarity cell to the task pool, during which the source address is the first solidarity cell's address, the destination address is the task pool's address, and the payload comprises a list of functions the first solidarity cell is configured to perform;

searching the task pool for a task that is ready to be processed and has a function that the first solidarity cell can perform; and

returning to the first solidarity cell, during which the source address is the task pool's address, the destination address is the first solidarity cell's address, and the payload comprises the descriptor of the matching task.

14. An apparatus for parallel processing of a large computing requirement, the apparatus comprising:

a central processing unit ("CPU");

a task pool in electronic communication with the CPU;

first solidarity cell in electronic communication with the task pool, *the first solidarity cell comprising a first agent configured to proactively retrieve, from the task pool, a matching task for the solidarity cell to process*; and

a second solidarity cell comprising a second agent that proactively retrieves matching tasks from the task pool for the second solidarity cell to process, wherein the matching task for each solidarity cell is a task in the task pool that is ready to be processed and can be performed by the solidarity cell;

wherein each solidarity cell sends its agent to the task pool when the solidarity cell does not have a matching task to process; and

wherein each agent comprises a source address, a destination address, and a payload, and wherein each agent retrieves a matching task by:

being dispatched by its solidarity cell to the task pool, during which the source address is its solidarity cell's address, the destination address is the task pool's address, and the payload comprises a list of functions the agent's solidarity cell is configured to perform;

searching the task pool for a task that is ready to be processed and has a function that the agent's solidarity cell can perform; and

returning to its solidarity cell, during which the source address is the task pool's address, the destination address is the agent's solidarity cell's address, and the payload comprises a descriptor of the matching task.

Ex. 1001, 7:41–8:9, 8:51–9:15 (emphases added).

F. Prior Art and Asserted Grounds

Petitioner asserts that claims 1–14 would have been unpatentable on the following grounds:

Claim(s) Challenged	35 U.S.C. §¹	Reference(s)/Basis
1–10, 12–14	103(a)	Lurie, ² Ethernet Specification ³
11	103(a)	Lurie, Ethernet Specification, Lippett ⁴

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 ’777 patent issued from an application filed before March 16, 2013, we apply the pre-AIA versions of the statutory bases for unpatentability. *See* Ex. 1001, codes (21), (22) (application filed Jan. 25, 2013).

² US 2007/0124363 A1, published May 31, 2007 (Ex. 1007).

³ 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.

⁴ US 2007/0220517 A1, published Sept. 20 2007 (Ex. 1031).

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

Patent Owner relies on the testimony of Brent Nelson (Ex. 2001).

II. 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(a) 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).

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

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. 6 (citing Ex. 1003 ¶¶ 27–28). Petitioner further argues that “[a] post-graduate degree or additional formal education can compensate for less experience, and vice-versa.” Pet. 6 (citing Ex. 1003 ¶¶ 27–28).

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

Because it is consistent with the ’777 patent and the asserted prior art, we adopt Petitioner’s proposed formulation and apply it in our obviousness evaluations below.

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 proposes construction for the terms “solidarity cell” and “thread(s).” Pet. 6–8. For all other terms, Petitioner argues 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. 6 (emphasis added)

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. Lurie and Ethernet Specification

Petitioner argues that claims 1–10 and 12–14 would have been obvious over Lurie and Ethernet Specification. *See* Pet. 8–48. Based on the current record, we are not sufficiently persuaded that Petitioner has established a reasonable likelihood of prevailing on this asserted obviousness ground with respect to any one of claims 1–10 and 12–14.

1. 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.” Ex. 1007 ¶ 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.” Ex. 1007 ¶ 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].” Ex. 1007 ¶ 61. This embodiment is shown in Figure 3B, which is reproduced below.

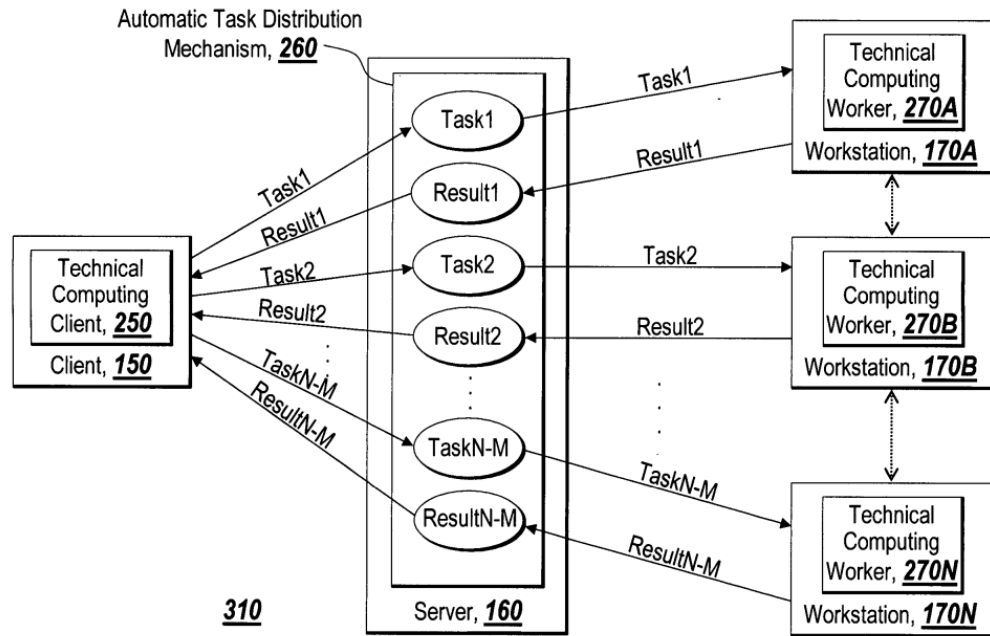


Fig. 3B

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.” Ex.1007 ¶ 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.” Ex.1007 ¶ 71. The TCC submits task to the ATD Mechanism “to make them available for distribution to any of the technical computing workers 270A-270N.” Ex. 1007 ¶ 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].” Ex. 1007 ¶ 72.

The TCC is then able to obtain the results from the ATD Mechanism.

Ex. 1007 ¶ 72.

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

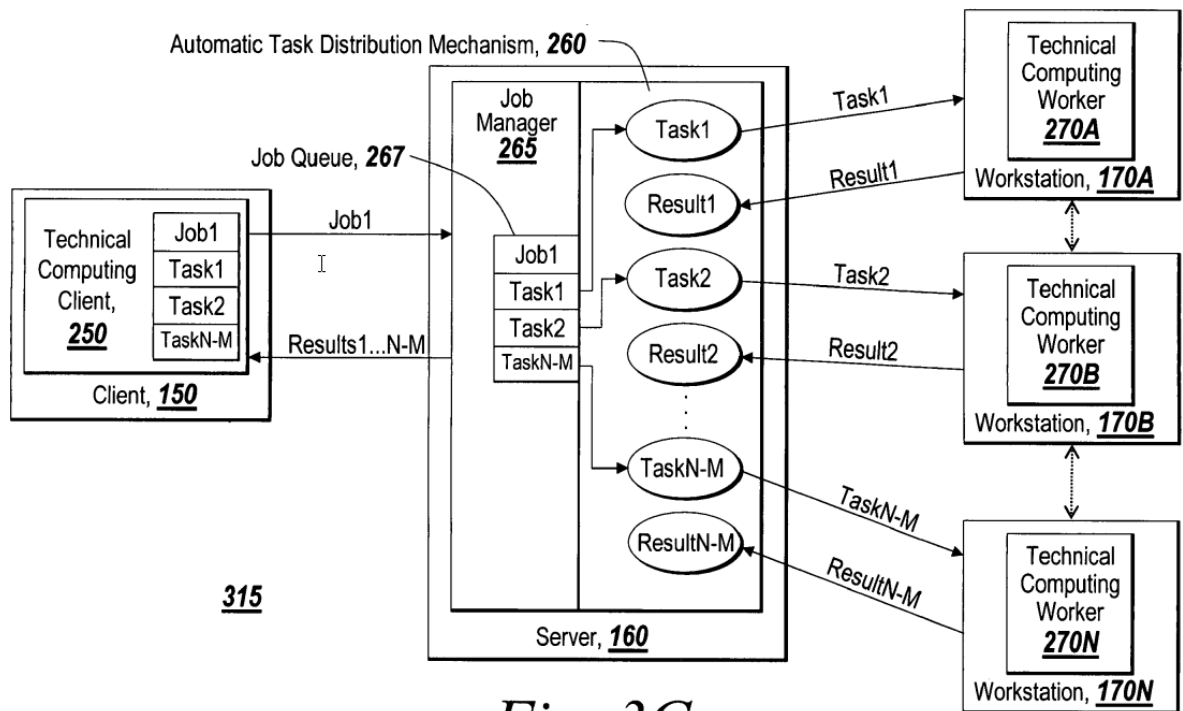


Fig. 3C

Figure 3C “is a block diagram of the batch automatic distribution mode of operation.” Ex. 1007 ¶ 20. 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].” Ex. 1007 ¶ 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].” Ex. 1007 ¶ 73.

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].

Ex. 1007 ¶ 74. 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].” Ex. 1007 ¶ 75.

2. *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.”

Ex. 1009, i.

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

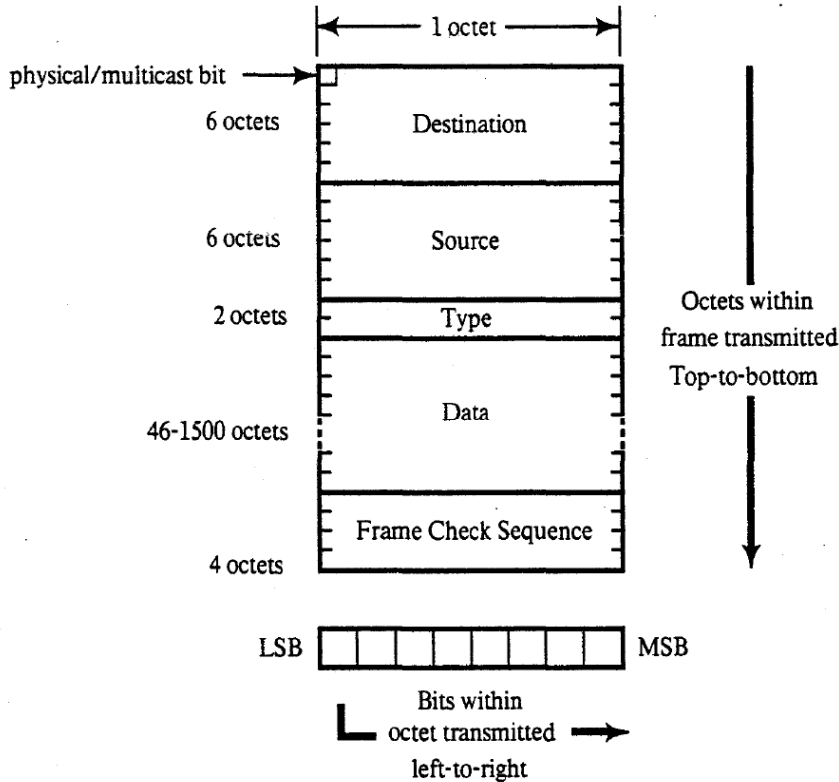


Figure 6-1: Data Link Layer Frame Format

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

Ex. 1009, 20.

3. Analysis of Claim 14

Claim 14⁵ recites, in relevant part, “a first solidarity cell in electronic communication with the task pool, the first solidarity cell comprising *a first agent configured to proactively retrieve*, from the task pool, *a matching task* for the solidarity cell to process.” Ex. 1001, 8:55–58 (emphasis added).

Petitioner argues that Lurie teaches that “[w]ithout receiving any instructions from the TCC (CPU), or the ATD for that matter, Lurie’s TCWs (cell) send a data frame/packet (agent) to the ATD [Mechanism] (task pool)

⁵ Because the Petition focuses first on claim 14, we do the same.

to retrieve a matching task.” Pet. 21 (citing Ex. 1003 ¶¶ 146–158). Specifically, Petitioner argues that “Lurie’s TCWs send agents (data frames/packets) to ‘take’ tasks from the ATD [Mechanism], and do so without requiring an instruction from the TCC to the TCW or its agent.” Pet. 21 (citing Ex. 1007 ¶ 7; Ex. 1003 ¶¶ 149–158). According to Petitioner, “the ATD [Mechanism] is an intermediary: ‘the [ATD Mechanism] provides a level of indirection between the [TCC] and the [TCWs],’ including their agents.” Pet. 22 (first alteration added; other alterations in original) (quoting Ex. 1007 ¶¶ 61, 92).

Petitioner also argues that “Lurie’s TCWs do not receive an instruction to retrieve a task from the ATD, either.” Pet. 23. According to Petitioner, although “Lurie’s TCWs ‘register[]’ with the ATD [Mechanism] ‘to receive notification of one or more tasks (step 527) becoming available, or appearing, in the [ATD Mechanism],’ . . . the ATD’s notification merely informs the TCWs when tasks first become available.” Pet. 23–24 (alterations adding Mechanism added) (quoting Ex. 1007 ¶¶ 87, 97) (citing Ex. 1003 ¶ 154). Petitioner argues that “[s]uch notifications are not instructions, as a TCW is not obligated to respond by retrieving a task.” Pet. 24; *see also* Pet. 32–37 (arguing that either (1) Lurie teaches that only one notification is sent or (2) that it would have been obvious to modify Lurie so that only an initial notification is sent).

Petitioner also argues that “the ’777 patent states that a solidarity cell is ‘proactive’ when ‘it obtains a task to perform by sending its agent to the task pool when the solidarity cell has no processing to perform.’” Pet. 24 (quoting Ex. 1001, 2:8–10); *see also* Ex. 1001, 5:22–25 (“The cells 12A . . . n act proactively in that they seek a task 22 from the task pool 13 as soon as the cells 12A . . . n become available to do further processing.”). According

to Petitioner, “Lurie’s solidarity cells are proactive under this definition, too. Lurie teaches that its TCWs (solidarity cells) obtain tasks when they do not have a task to process.” Pet. 24

We are not sufficiently persuaded that Petitioner has shown that the combination of Lurie and Ethernet Specification teaches or suggests the disputed limitation. There is no dispute that Lurie teaches that the ATD Mechanism sends the TCWs a notification that a task is available before the TCW obtains its first task. *See, e.g.*, Pet. 32 (“TCWs register with the ATD to receive a notification when tasks first become available.”). For example, Lurie describes how the ATD Mechanism sends a notification and that the TCW responds to that notification by obtaining a task: “The automatic task distribution mechanism 260 provides notification (step 538) to the technical computing worker 260 of the availability of the task. In response to receiving the notification (step 540), the technical computing worker obtains (step 544) the task provided (step 540) from the automatic task distribution mechanism 260.” Ex. 1007 ¶ 86; *see also* Ex. 1007, Fig. 5B (steps 538, 540, 544); Prelim. Resp. 29–30. Similarly, Lurie describes how the TCW receives a notification before obtaining a task: “When the task is submitted to the object exchange repository 662 by the job manager 265, the technical computing worker 270 receives a notification and takes the task from the object exchange repository 662 by performing a take operation.” Ex. 1007 ¶ 97; *see also* Prelim. Resp. 29. Lurie also describes how a notification is sent prior to a TCW taking a task when the system is operating in batch mode. *See* Ex. 1007 Fig. 5C (steps 538, 540, 542, 544). Because the TCW only takes a task after a notification, it is not acting proactively. *See* Ex. 2001 ¶ 50. Instead, the TCW is reacting to a notification that tasks are available.

Nor are we sufficiently persuaded by Petitioner’s argument that the notification is only required for the first task and that for all other tasks, the TCW will obtain the task without notification using the bag-of-tasks model. *See* Pet. 33–37. Assuming Petitioner is correct in that Lurie uses the bag-of-tasks model, the TCW still requires the first notification before it starts obtaining tasks. As Dr. Lowenthal explains, the notification is required to “kick off the procedure.” Ex. 1003 ¶ 196. Because of the undisputed need for the initial notification, it does not matter that the TCW can obtain additional tasks without further notifications. A cell that sometimes needs a notification to obtain a task is not proactive; rather, it is at least partially reactive.

We are also not sufficiently persuaded by Petitioner’s argument that a notification is not an instruction for two reasons. *See* Pet. 23–24.

First, Petitioner has not shown that proactive means acting without an instruction. Indeed, Petitioner has not proposed an explicit construction of the term proactive. Therefore, for purposes of this analysis, it does not matter whether the TCW receives an instruction to obtain a task.

Second, Lurie’s notification is, effectively, an instruction. Petitioner has not shown any disclosure in Lurie where a TCW obtains a task without first being notified that the task is available. Instead, as discussed above, Petitioner and its declarant concede that a notification is required before the first task can be obtained. *See* Pet. 32; Ex. 1003 ¶ 196. Accordingly, that notification acts as instruction telling the TCW that it can obtain tasks. And although Lurie uses the term “notification” instead of instruction, there is no requirement in an obviousness analysis for the prior art to “contain a description of the subject matter of the . . . claim in *ipsissimis verbis*.” *In re May*, 574 F.2d 1082, 1090 (CCPA 1978).

We are also not persuaded by Petitioner’s argument regarding the purported definition of proactive in the Specification of the ’777 patent for two reasons. *See* Pet. 24.

First, Petitioner does not argue for a specific construction of the term “proactive.” Instead, Petitioner argues that prior art teaches the claim limitation “under any reasonable construction, including their plain meaning.” Pet. 6. Petitioner has not explained adequately why the plain meaning of “proactive” is “it obtains a task to perform by sending its agent to the task pool when the solidarity cell has no processing to perform.” Ex. 1002, 2:8–10.

Second, we are not persuaded that the Specification provides a definition of the term “proactive.” Instead, the cited sections of the Specification simply describe an aspect of being proactive. *See* Ex. 1001, 2:8–10. Moreover, adopting it as an explicit definition would render other portions of claim 14 superfluous. Specifically, claim 14 already requires that “each solidarity cell sends its agent to the task pool when the solidarity cell does not have a matching task to process.” Ex. 1001, 8:65–67. If proactive means that the cell “obtains a task to perform by sending its agent to the task pool when the solidarity cell has no processing to perform” (Ex. 1001, 2:8–10), then there would be no need for claim 14 to explicitly require it elsewhere in the claim. We do not construe “proactive” in a manner to render other limitations superfluous. *See Wasica Finance GmbH v. Continental Automotive Systems, Inc.*, 853 F.3d 1272, 1288 n.10 (Fed. Cir. 2017) (“It is highly disfavored to construe terms in a way that renders them void, meaningless, or superfluous.”); *Merck & Co. v. Teva Pharms. USA, Inc.*, 395 F.3d 1364, 1372 (Fed. Cir. 2005) (“A claim construction that gives

meaning to all the terms of the claim is preferred over one that does not do so.”).

Accordingly, Petitioner has not shown a reasonable likelihood of showing that the subject matter of claim 14 would have been unpatentable based on Lurie and Ethernet Specification.

4. Claims 1–10, 12, and 13

Independent claim 1 recites substantially the same limitation as discussed above for independent claim 14. *Compare* Ex. 1001, 7:41–8:9, *with id.* at 8:51–9:15. The parties rely on the same evidence and arguments for these two independent claims. *See* Pet. 42–43; Prelim. Resp. 27. Claims 2–10, 12, and 13 depend, directly or indirectly, from claim 1. Accordingly, for the reasons discussed in the previous section, Petitioner has not shown a reasonable likelihood of showing that the subject matter of claims 1–10, 12, and 13 would have been unpatentable based on Lurie and Ethernet Specification. *See In re Fritch*, 972 F.2d 1260, 1266 (Fed. Cir. 1992) (“[D]ependent claims are nonobvious if the independent claims from which they depend are nonobvious.”).

E. Lurie, Ethernet Specification, and Lippett

Petitioner argues that claim 11 would have been obvious over Lurie, Ethernet Specification, and Lippett. *See* Pet. 49–50. Based on the current record, we are not sufficiently persuaded that Petitioner has established a reasonable likelihood of prevailing on this asserted obviousness ground with respect to claim 11.

1. Lippett

Lippett is titled “Scheduling in a Multicore Processor” and is directed to “instructions for performing a method of scheduling executable

transactions within a multicore processor comprising a plurality of processor elements.” Ex. 1031, codes (54), (57).

2. *Analysis of Claim 11*

Petitioner relies on Lippett to teach the additional limitation recited in claim 11. *See* Pet. 49–50. Petitioner does not argue that Lippett cures the deficiency identified above with regard to the combination of Lurie and Ethernet Specification.

Because claim 11 indirectly depends from claim 1, Petitioner has not shown a reasonable likelihood of showing that the subject matter of claim 11 would have been unpatentable based on Lurie, Ethernet Specification, and Lippett. *See Fritch*, 972 F.2d at 1266.

III. CONCLUSION

For the foregoing reasons, we determine that Petitioner fails to show a reasonable likelihood that it would prevail with respect to at least one of the claims challenged in the Petition. We, therefore, do not institute an *inter partes* review of the challenged claims based on the arguments and evidence presented in the Petition.

IV. ORDER

In consideration of the foregoing, it is hereby:

ORDERED that no *inter partes* review is instituted as to any claim of the ’777 patent.

IPR2023-00390
Patent 9,146,777 B2

FOR PETITIONER:

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

FOR PATENT OWNER:

Michael Kelly
BEUS GILBERT MCGRODER PLLC
mkelly@beusgilbert.com

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

Daniel Anderson
ERNST, BROWN & DRAPER, PLLC
daniel.anderson@andersonpatents.com