

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

VMWARE, INC.; DELL TECHNOLOGIES INC.; DELL INC.;
AND EMC CORPORATION,
Petitioner,

v.

WSOU INVESTMENTS, LLC
D/B/A BRAZOS LICENSING AND DEVELOPMENT,
Patent Owner.

IPR2021-00572
Patent 9,164,800 B2

Before THU A. DANG, BRIAN J. McNAMARA, and
JOHN D. HAMANN, *Administrative Patent Judges*.

HAMANN, *Administrative Patent Judge*.

JUDGMENT
Final Written Decision
Determining All Challenged Claims Unpatentable
35 U.S.C. § 318(a)

I. INTRODUCTION

In this *inter partes* review, instituted pursuant to 35 U.S.C. § 314, VMware, Inc.; Dell Technologies Inc.; Dell Inc.; and EMC Corporation (collectively “Petitioner”) challenges the patentability of claims 1, 2, 6–9, 13, 14, and 18–21 (“the challenged claims”) of U.S. Patent No. 9,164,800 B2 (Ex. 1001, “the ’800 patent”), owned by WSOU Investments, LLC d/b/a/ Brazos Licensing and Development (“Patent Owner”). We have jurisdiction under 35 U.S.C. § 6. This Final Written Decision is entered pursuant to 35 U.S.C. § 318(a) (2018) and 37 C.F.R. § 42.73 (2021).

For the reasons discussed herein, we determine that Petitioner shows by a preponderance of the evidence that claims 1, 2, 6–9, 13, 14, and 18–21 are unpatentable.

A. Procedural History

Petitioner filed a Petition requesting *inter partes* review of the challenged claims of the ’800 patent. Paper 1 (“Pet.”). Patent Owner filed a Preliminary Response. Paper 7 (“Prelim. Resp.”).

We instituted *inter partes* review of all of the challenged claims of the ’800 patent on all of the grounds raised in the Petition. Paper 8 (“Dec. on Inst.”), 7, 52–53. Patent Owner filed a Response to the Petition. Paper 10 (“PO Resp.”). Petitioner filed a Reply to Patent Owner’s Response. Paper 13 (“Pet. Reply”). Patent Owner filed a Sur-reply to Petitioner’s Reply. Paper 14 (“PO Sur-reply”).

An oral hearing was held on June 7, 2022. A transcript of the oral hearing is included in the record. Paper 19 (“Tr.”).

B. Real Parties-in-Interest

VMware, Inc.; Dell Technologies Inc.; Dell Inc.; and EMC Corporation identify themselves as the real parties-in-interest. Pet. 1. Patent Owner identifies itself as the real party-in-interest. Paper 4, 1.

C. Related Matters

The parties identify *WSOU Investments, LLC v. Dell Technologies*, 6-20-cv-00481-ADA (W.D. Tex.) (“the parallel litigation”) in which the ’800 patent is asserted, and which may affect, or be affected by, a decision in this proceeding. Pet. 1; Paper 4, 2.

D. The Challenged Patent

The ’800 patent generally relates to cloud computing. Ex. 1001, 1:7–8. According to the ’800 patent, “[m]any cloud operators . . . host cloud services using a few large data centers, providing a relatively centralized operation.” *Id.* at 1:12–14. “This centralized operation, however, may not be well suited for hosting various types of applications, such as those with strict delay or reliability requirements.” *Id.* at 1:17–20. The ’800 patent discloses that on the other hand, “[d]istributed data center architectures . . . provide a larger number of smaller data centers that may be geographically distributed.” *Id.* at 1:21–23. “Under such a distributed system, the effects of network propagation delay may be reduced by providing cloud resources that are closer to a requestor in terms of geographic or network distance than a centralized cloud may be able to provide,” according to the ’800 patent. *Id.* at 1:25–29.

The ’800 patent discloses that “a cloud may provision compute nodes, such as virtual machines, to process data stored by data nodes within the cloud.” *Id.* at 1:30–32. However, “a compute node may be located on a different server or rack in the data center of the data node, or may be located

in a different data center from the data node.” *Id.* at 1:34–37. According to the ’800 patent, “[i]n such a case, issues such as propagation delay and bandwidth may further impact the communication latency between data nodes and compute nodes” *Id.* at 1:37–40. The ’800 patent discloses that “[n]etwork latencies may be kept relatively low by thoughtful placement of the compute node in relation to the nodes with which the compute node will communicate.” *Id.* at 4:40–42.

Figure 1, shown below, illustrates an exemplary network in accordance with the invention of the ’800 patent. *Id.* at 3:55–56.

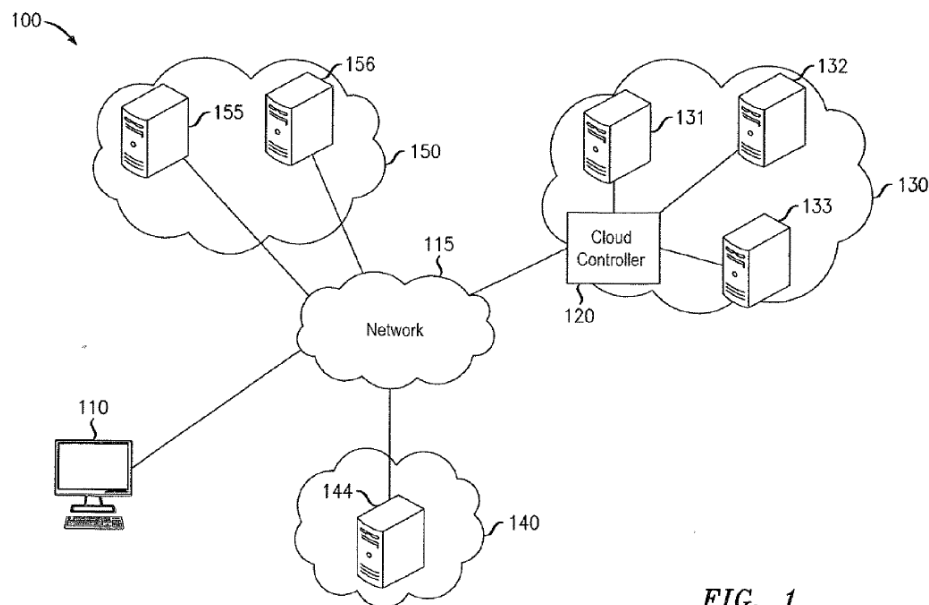


FIG. 1

Figure 1 “illustrates an exemplary network 100 for providing cloud resources.” *Id.* at 4:49–50. As illustrated, network 100 “include[s] a client device 110, a network 115, a cloud controller 120, and data centers 130, 140, [and] 150.” *Id.* at 4:50–53. The ’800 patent discloses that the client device 110 may be any device (e.g., “a desktop computer, laptop, tablet, mobile device, server, or blade”) that is “configured to utilize one or more cloud resources” (e.g., virtual machines or storage devices). *Id.* at 4:54–57. The client device 110 can communicate with the cloud controller 120, via the

network 115, to request the use of cloud resources. *Id.* at 4:57–59. In the ’800 patent, the cloud controller 120 is configured to allocate to the client device 110 the requested cloud resources, which can be provided by the cloud devices 131, 132, 133, 144, 155, and 156. *Id.* at 5:34–37. The ’800 patent discloses that the cloud devices “may be a desktop computer, laptop, tablet, mobile device, server, or blade.” *Id.* at 5:61–63. After allocation of the cloud resources by the cloud controller 120, the client device 110 may communicate with the cloud devices to utilize the allocated cloud resources. *Id.* at 4:64–5:1.

The data centers are locations supporting one or more of the cloud devices that provide cloud resources. *Id.* at 5:42–43. As illustrated, the data center 130 hosts the cloud devices 131, 132, and 133; the data center 140 hosts cloud device 144; and the data center 150 hosts the cloud devices 155 and 156. *Id.* at 5:44–46, Fig. 1. “The data centers 130, 140, [and] 150 may be geographically distributed or may be situated at different network distances from the client device 110.” *Id.* at 5:46–49.

The ’800 patent discloses that “[t]he cloud controller 120 may employ various approaches for optimizing assignment of compute nodes to data nodes.” *Id.* at 6:30–31. Figure 2, shown below “illustrates an exemplary bipartite graph of data nodes and compute nodes.” *Id.* at 3:57–58.

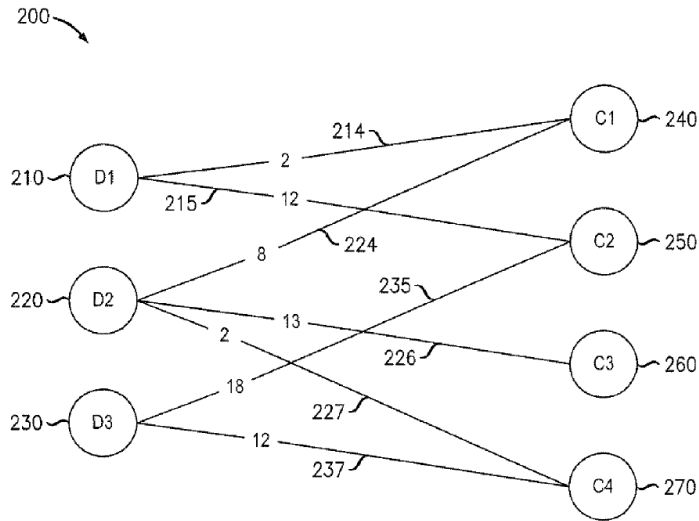


FIG. 2

Figure 2 “illustrates an exemplary bipartite graph 200 of data nodes 210, 220, [and] 230[,] and compute nodes 240, 250, 260, [and] 270.” *Id.* at 6:34–36. “The nodes . . . may be hosted by, for example, one or more of the data centers 130, 140, [and] 150 of the exemplary network 100.” *Id.* at 6:42–44.

According to the ’800 patent, “[t]he bipartite graph 200 illustrates an assignment problem for solution by a cloud controller.” *Id.* at 6:48–49. “Possible assignments between the data nodes 210, 220, [and] 230 and the compute nodes 240, 250, 260, [and] 270 [are] shown as edges 214, 215, 224, 226, 227, 235, [and] 237.” *Id.* at 6:49–52. “Each edge . . . may be associated with a cost representing, for example, a latency or network distance between the two nodes.” *Id.* at 6:52–54.

In the example illustrated in Figure 2, “the cost of assigning compute node 240 to data node 210 may be represented as ‘2,’ while the cost of assigning compute node 250 to the same data node may be represented as ‘12.’” *Id.* at 6:57–60. “Thus, assignment of compute node 250 to the data node 210 may introduce more latency to the cloud system than assignment

of compute node 240 to the data node 210,” according to the ’800 patent. *Id.* at 6:60–63.

In addition, the ’800 patent discloses that “[t]he cloud controller 120 may . . . consider one or more optimization constraints, or ‘assignment objectives,’ when determining an optimum assignment of compute nodes to data nodes.” *Id.* at 7:10–13. “For example, the cloud controller may consider an assignment objective of minimizing a total latency, minimizing a maximum latency, minimizing a total latency within a limit on maximum latency, and minimizing a maximum latency within a limit on total latency.” *Id.* at 7:13–17.

E. The Challenged Claims

Petitioner challenges claims 1, 2, 6–9, 13, 14, and 18–21 of the ’800 patent. Claims 1 and 13 are independent claims. Claim 1 is illustrative of the challenged claims and is reproduced below:

1. A method performed by a cloud controller for assigning compute nodes to data nodes, the method comprising:
 - obtaining, by the cloud controller, a set of compute nodes, a set of data nodes, and a set of edges between compute nodes and data nodes;
 - determining an assignment objective to be achieved in assigning compute nodes from the set of compute nodes to data nodes from the set of data nodes; and
 - assigning a compute node of the set of compute nodes to a data node of the set of data nodes based on the assignment objective and a plurality of costs associated with the set of edges;wherein the assignment objective is based on a latency cost used in obtaining a set of assignments by applying an algorithm.

Ex. 1001, 15:23–37.

F. Instituted Grounds of Unpatentability

We instituted trial based on the following grounds of unpatentability, which are all the grounds of unpatentability raised in the Petition:

Claim(s) Challenged	35 U.S.C. §¹	Reference(s)/Basis
1, 2, 7–9, 13, 14, 19–21	102(b)	Queck ²
1, 2, 6–9, 13, 14, 18–21	103(a)	Queck
1, 2, 6–9, 13, 14, 18–21	103(a)	Queck, Singh ³
1, 8, 9, 13, 20, 21	103(a)	Li, ⁴ Singh

Pet. 6, 12–65. Petitioner submits in support of its arguments (i) the Declaration of Nader F. Mir, Ph.D. (Ex. 1003); (ii) the Supplemental Declaration of Nader F. Mir, Ph.D. (Ex. 1020); (iii) the Declaration of Dr. James L. Mullins, Ph.D. (Ex. 1011); (iv) the Supplemental Declaration of Dr. James L. Mullins, Ph.D. (Ex. 1023); and (v) the Declaration of Julia O’Keefe (Ex. 1022).

II. LEVEL OF ORDINARY SKILL IN THE ART

To determine 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 v. John Deere Co.*, 383 U.S. 1, 17 (1966). In assessing the level of ordinary skill in the art, various factors may be considered, including the “type of problems encountered in the art;

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 ’800 patent issued from an application filed before March 16, 2013, we apply the pre-AIA version of the statutory basis for unpatentability.

² EP 1978445 B1, issued Oct. 6, 2010 (Ex. 1006, “Queck”).

³ Aameek Singh et al., *SPARK: Integrated Resource Allocation in Virtualization-Enabled SAN Data Centers*, IBM Research Report, Apr. 18, 2007 (Ex. 1007, “Singh”).

⁴ US 8,972,983 B2, issued Mar. 3, 2015 (Ex. 1010, “Li”).

prior art solutions to those problems; rapidity with which innovations are made; sophistication of the technology; and educational level of active workers in the field.” *In re GPAC, Inc.*, 57 F.3d 1573, 1579 (Fed. Cir. 1995) (citing *Custom Accessories, Inc. v. Jeffrey-Allan Indus., Inc.*, 807 F.2d 955, 962–63 (Fed. Cir. 1986)). “[O]ne or more factors may predominate.” *Id.*

In our Decision on Institution, we adopted Petitioner’s proposed definition for one having ordinary skill in the art at the time of the invention of the ’800 patent as one who “would have had at least a bachelor’s degree in computer science, computer engineering, or electrical engineering and two to three years of work experience related to administering distributed computing environments”; and that “[t]his work experience may have included, for example, experience in cloud network design and analysis, client-server, cloud controller communication analysis and evaluation, load balancing, virtualization, latencies optimization, and managing physical and virtual computing environments in the cloud”; and that “[a] master’s degree or higher would have been acceptable in place of the work experience described above.” Dec. on Inst. 21 (quoting Pet. 7–8 (citing Ex. 1003 ¶ 23)).

Patent Owner does not dispute our adoption of Petitioner’s definition, and does not provide its own definition for the level of ordinary skill at the time of the invention of the ’800 patent. *See* PO Resp. 7–8.

Because Petitioner’s definition of the level of skill in the art is consistent with the ’800 patent and the asserted prior art, we maintain Petitioner’s definition for purposes of this Final Written Decision. *See Okajima v. Bourdeau*, 261 F.3d 1350, 1355 (Fed. Cir. 2001); *GPAC*, 57 F.3d at 1579; *In re Oelrich*, 579 F.2d 86, 91 (CCPA 1978). We apply Petitioner’s definition in our analysis below.

III. CLAIM CONSTRUCTION

Because the Petition was filed after November 13, 2018, we apply the same claim construction standard that would be used in a civil action under 35 U.S.C. § 282(b), following the standard articulated in *Phillips v. AWH Corp.*, 415 F.3d 1303 (Fed. Cir. 2005) (en banc). 37 C.F.R. § 42.100(b); 83 Fed. Reg. 51,340, 51,340–41, 51,343 (Oct. 11, 2018). In applying such standard, claim terms are generally given their ordinary and customary meaning, as would be 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*, 415 F.3d at 1312–13. “In determining the meaning of the disputed claim limitation, we look principally to the intrinsic evidence of record, examining the claim language itself, the written description, and the prosecution history, if in evidence.” *DePuy Spine, Inc. v. Medtronic Sofamor Danek, Inc.*, 469 F.3d 1005, 1014 (Fed. Cir. 2006) (citing *Phillips*, 415 F.3d at 1312–17).

The parties do not provide their own proposed constructions for any claim terms. The district court, however, in the parallel litigation construed two terms of the challenged claims: “[determining/determine] an assignment objective” and “latency cost.” Ex. 2004, 17 & n.1. The district court concluded that these terms should be afforded their plain and ordinary meaning. *Id.* In addition, the district court further concluded that for the “[determining/determine] an assignment objective” term that “[t]he plain-and-ordinary meaning only excludes hard-coded single-objective assignments[, and i]t does not, however, exclude a determination where there may only be a single assignment objective available when the ‘determination’ step is made.” *Id.* The parties agree that we should apply

the district court’s construction for this term for this proceeding. *See* Pet. 6; Tr. 18:17–20:19; PO Sur-reply 3.

We have reviewed the district court’s constructions. *See* 37 C.F.R. § 42.100(b) (“Any prior claim construction determination concerning a term of the claim in a civil action . . . that is timely made of record in the *inter partes* review proceeding will be considered.”). We agree with the district court and the parties that the terms of the challenged claims should be afforded their plain and ordinary meanings. And we adopt the district court’s statements about exclusion regarding the scope of the plain and ordinary meaning for “[determining/determine] an assignment objective.” We conclude that these exclusion statements are consistent with the claim language of the challenged claims and the Specification of the ’800 patent.

We do not provide any further constructions for any claim terms. *See, e.g., Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 868 F.3d 1013, 1017 (Fed. Cir. 2017) (quoting *Vivid Techs., Inc. v. Am. Sci. & Eng’g, Inc.*, 200 F.3d 795, 803 (Fed. Cir. 1999)) (“[W]e need only construe terms ‘that are in controversy, and only to the extent necessary to resolve the controversy.’”).

IV. PRINCIPLES OF LAW

A claim is unpatentable under 35 U.S.C. § 102 “if each and every [claim] limitation is found either expressly or inherently in a single prior art reference.” *Nidec Motor Corp. v. Zhongshan Broad Ocean Motor Co.*, 851 F.3d 1270, 1273 (Fed. Cir. 2017). “A claim limitation is inherent in the prior art if it is necessarily present in the prior art, not merely probably or possibly present.” *Akamai Techs., Inc. v. Cable & Wireless Internet Servs., Inc.*, 344 F.3d 1186, 1192 (Fed. Cir. 2003) (citing *Rosco, Inc. v. Mirror Lite, Co.*, 304 F.3d 1373, 1380 (Fed. Cir. 2002)).

A claim is unpatentable under 35 U.S.C. § 103(a) 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 of the invention to a person having ordinary skill in the art. *KSR Int’l Co. v. Teleflex, Inc.*, 550 U.S. 398, 406 (2007). The question of obviousness is resolved on the basis of 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 ordinary skill in the art; and (4) objective evidence of non-obviousness, if present.⁵ See *Graham*, 383 U.S. at 17–18. When evaluating a claim for obviousness, we also must “determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue.” *KSR*, 550 U.S. at 418 (citing *In re Kahn*, 441 F.3d 977, 988 (Fed. Cir. 2006)).

V. ALLEGED OBVIOUSNESS OVER QUECK

Petitioner argues that Queck renders claims 1, 2, 6–9, 13, 14, and 18–21 of the ’800 patent obvious. Pet. 12–46. We have reviewed the parties’ arguments and the evidence of record. For the reasons that follow, we determine that by a preponderance of the evidence Petitioner shows that claims 1, 2, 6–9, 13, 14, and 18–21 would have been obvious over Queck.

A. Summary of *Queck*

Queck relates to “[g]rid computing environments [which] generally [comprise] distributed, networked computing models in which a plurality of computers are used collaboratively to solve one or more problems.” Ex. 1006 ¶ 2. Queck teaches that these problems could be computationally

⁵ Patent Owner does not present arguments or evidence of such objective evidence of non-obviousness. See generally PO Resp.; PO Sur-reply.

intensive or database intensive (i.e., “requir[ing] large and/or frequent calls (e.g., queries) to a database”). *Id.* ¶¶ 2–3. Queck teaches “distributing data and task instances in [the] grid environment, to thereby allow database-intensive applications to benefit from the grid environment.” *Id.* ¶ 16. To that end, Queck teaches that its system “may be configured to determine a number of factors to decide whether and how to deploy the task instance[s] . . . and the data partition[s].” *Id.* ¶ 37. For example, the system “may determine currently-available computing and memory resources available at each of the nodes . . . of the grid environment,” and “may determine a latency associated with one or more pairs of the nodes.” *Id.*

The system then “may make a determination as to whether and how [a] data partition . . . should be deployed in order to improve an execution time of [a] task instance.” *Id.* For example, Queck teaches “determin[ing] a pair of . . . potential processing nodes and . . . potential data nodes that have the lowest latency (e.g., that are closest to one another).” *Id.* ¶ 85. Queck teaches that its system “may determine whether the transfer of the data partition from the initial node to the selected data node (along with deployment of the task instance to the processing node . . .), will improve an overall execution time of the task instance.” *Id.* ¶ 87. “If the execution time will be improved with the transfer, then the task instance and the data partition may be deployed” *Id.* ¶ 88.

B. Challenged Claim 1

Petitioner argues that Queck teaches claim 1. Pet. 17–36. Patent Owner argues that Queck fails to teach claim 1’s “determining” step. PO Resp. 9–13. As we discuss below, we find that Petitioner shows that Queck teaches all of the limitations of claim 1, and renders claim 1 obvious.

1. Method Performed by a Cloud Controller

Claim 1’s preamble recites “[a] method performed by a cloud controller for assigning compute nodes to data nodes.” Ex. 1001, 15:23–24. Petitioner first argues that “Queck’s disclosure of central node 102 teaches a ‘cloud controller.’” Pet. 18 (citing Ex. 1003 ¶ 94). In particular, Petitioner argues that central node 102 may be “configured to deploy the task instance 116, and related task instances of tasks of the task repository 114, to one or more nodes 104a, 104b of the grid environment,” and may be “configured to determine whether and how to deploy (e.g., transfer) [a] data partition 122a within the grid environment of F[igure] 1.” *Id.* (quoting Ex. 1006 ¶ 36; citing *id.* ¶¶ 37–39, 16:12–25 (reciting claim 1), 16:40–50 (reciting claim 6); *see also id.* (quoting Ex. 1006 ¶ 17) (arguing that “Queck explains, central node 102 ‘may refer to hardware and/or software (e.g., virtual) implementations of a device(s) of a grid network’”). And Petitioner argues that “Queck further teaches that ‘[g]rid computing environments generally refer to distributed, networked computing models in which a plurality of computers are used collaboratively to solve one or more problems.’” *Id.* at 19 (quoting Ex. 1006 ¶ 2). According to Petitioner, one of ordinary skill in the art “would have understood that such a ‘grid computing environment’ is a type of ‘cloud’ network.” *Id.* (citing Ex. 1003 ¶ 95). Petitioner argues that one of ordinary skill in the art “would have understood that, in deploying task instances and data partitions to nodes, Queck’s central node 102 allocates resources and therefore is a ‘cloud controller.’” *Id.* at 18 (citing Ex. 1003 ¶ 94).

Second, Petitioner argues that “[i]n Queck, nodes are referred to as either ‘data nodes’ or ‘processing nodes,’” and one of ordinary skill in the art “would have understood that Queck’s ‘processing nodes’ constitute

‘compute nodes.’” *Id.* at 19 (citing Ex. 1006 ¶ 5). Petitioner argues that Queck teaches “that a processing node, in executing a task instance, will make calls to data stored in a data node and will process that data.” *Id.* at 19–20 (citing Ex. 1006 ¶¶ 29–30, 33–35, 44). Petitioner adds that Queck teaches “that central node 102 . . . deploys task instances 116a and 116b to processing nodes 104c and 104f, respectively, so that the processing nodes make calls to data partition 122a stored on data nodes 104d and 104e, respectively.” *Id.* at 20 (citing Ex. 1006 ¶¶ 42–49, Fig. 2). Petitioner argues that “[a]s a result, . . . central node 102 (the ‘cloud controller’ . . .) assigns processing node 104c to data node 104d . . . , and also assigns processing node 104f to data node 104e.” *Id.* (citing Ex. 1006, Fig. 2; Ex. 1003 ¶ 97).

After reviewing Petitioner’s arguments and evidence, including Dr. Mir’s testimony, which are not addressed by Patent Owner (*see generally* PO Resp.), we are persuaded that Petitioner demonstrates by a preponderance of the evidence that Queck teaches claim 1’s preamble. In light of this finding, we need not, and thus do not, reach whether claim 1’s preamble is limiting.

2. Obtaining by the Cloud Controller

Claim 1 further recites “obtaining, by the cloud controller, a set of compute nodes, a set of data nodes, and a set of edges between compute nodes and data nodes.” Ex. 1001, 15:25–27. Petitioner argues that Queck teaches that the central node 102 obtains a plurality of processing nodes and a plurality of data nodes. Pet. 22 (citing Ex. 1006 ¶¶ 26, 52–53, Figs. 1–2, 16:40–50 (reciting claim 6)).

As to edges, Petitioner argues that “although not illustrated specifically as edges in a graph, Queck discloses edges in its description of latency values ‘between each pair of nodes in the grid computing

environment,’ and further teaches that these latency values ‘may be stored in a latency matrix 610 or other format.’” *Id.* (quoting Ex. 1006 ¶ 75).

According to Petitioner, one of ordinary skill in the art “would have understood that each position in such a latency matrix represents a possible connection between two nodes.” *Id.* (citing Ex. 1003 ¶ 100). Petitioner argues that “[b]ecause every processing/data node assignment must result from a connection between nodes, [one of ordinary skill in the art] would have understood that the possible processing/data node assignments are a subset of the latency matrix.” *Id.* at 23 (citing Ex. 1003 ¶ 100). “As a result, the positions in the latency matrix teach the ‘edges’ (possible assignments) between processing and data nodes,” according to Petitioner. *Id.*

After reviewing Petitioner’s arguments and evidence, including Dr. Mir’s testimony, which are not addressed by Patent Owner (*see generally* PO Resp.), we are persuaded that Petitioner demonstrates by a preponderance of the evidence that Queck teaches “obtaining, by the cloud controller, a set of compute nodes, a set of data nodes, and a set of edges between compute nodes and data nodes.”

3. *Determining an Assignment Objective*

Claim 1 further recites “determining an assignment objective to be achieved in assigning compute nodes from the set of compute nodes to data nodes from the set of data nodes.” Ex. 1001, 15:28–30. Petitioner argues that Queck teaches this limitation. Pet. 24–34; Pet. Reply 13–18. Patent Owner disputes that this limitation is taught. PO Resp. 9–18; PO Sur-reply 8–21.

We agree with Petitioner and find that Queck teaches this limitation. First, we agree with Petitioner and find that Queck teaches at least two

different assignment objectives (i.e., goals) to be achieved in assigning processing nodes (i.e., compute nodes) to data nodes.⁶ Ex. 1006

¶¶ 111–112; Pet. 25. In particular, Queck teaches one assignment objective that optimizes performance of a *single* task instance, and teaches a second assignment objective that optimizes average performance across *multiple* instances of the same task. Ex. 1006 ¶¶ 108, 111–112. In other words, in deciding how to deploy task instances and data partitions to processing and data nodes, Queck teaches (i) “evaluat[ing] whether there may be a performance improvement when transferring the data partition on the basis of a single task instance,” or alternatively, (ii) considering “the accumulated performance of all associated task instances” “to improve the average performance of a task as a whole.” *Id.* ¶ 111; *see also id.* ¶¶ 37, 47 (teaching that provisioning manager 134 optimizes the performance of task instances by deploying a task instance and a data partition to a processing node and a data node, respectively, so as to minimize execution time), 108 (providing an algorithm for optimizing for a single task instance), 112 (providing an algorithm for optimizing across multiple task instances).

In addition, we find persuasive Dr. Mir’s testimony that the first goal “allows the system to optimize the performance of a particular task instance, which may come at the expense of other task instances, [while the second goal] allows the system to balance the demands of all instances of a task to optimize the performance of the task as a whole.” Ex. 1003 ¶ 110. We also find persuasive Dr. Mir’s testimony that “optimizing a single task’s

⁶ Because we find that Queck teaches these two different assignment objectives, we do not reach the parties’ arguments about whether Queck also teaches a third alternative assignment objective. *See* Pet. 27–29; PO Resp. 13–14; Pet. Reply 17 n.7; PO Sur-reply 6.

performance is different from . . . optimizing the average performance of the task as a whole (*i.e.*, the performance of 2+ task instances).” *Id.*; *see also id.* ¶¶ 109–111 (explaining Queck’s goals). We find that both of these portions of Dr. Mir’s testimony are consistent with Queck’s teachings. *Compare id.* ¶¶ 109–111, *with* Ex. 1006 ¶¶ 87, 108, 111–112. For example, Queck teaches that whether a data partition should be transferred may depend on the algorithm (which corresponds to an assignment objective) which is used. Ex. 1006 ¶¶ 108, 111–112 (providing different algorithms).

Second, we agree with Petitioner and find that Queck teaches that central node 102 determines which assignment objective is to be achieved when determining whether to optimize performance for a single task instance or multiple task instances. Pet. 30. For example, Queck teaches that “the provisioning manager 134 may be configured to determine a number of factors to decide whether and how to deploy the task instance 116 and the data partition 122a.” Ex. 1006 ¶ 37. This language teaches that Queck’s provisioning manager 134 determines which goal to optimize for so that the corresponding algorithm for that goal may be employed in deciding whether and how to deploy task instances and data partitions. *Id.* ¶¶ 37, 111–112. The provisioning manager 134 needs to have determined which algorithm to use—which depends on, and follows from, the goal—to decide whether and how to deploy task instances and data partitions. *Id.* ¶¶ 108, 111–112.

Moreover, Queck’s paragraph 76 teaches that “allocation module 604^[7] receives the resource information from the resource manager 606 and

⁷ Allocation module 604 is within provisioning manager 134, which is within central node 102. *E.g.*, Ex. 1006, Figs. 1, 6.

implements *one or more algorithms based on characteristics of the task instance and data partition to be deployed*, using the resource information provided by the resource manager 606 (including, e.g., the latency information).” Ex. 1006 ¶ 76 (emphasis added). Again, Queck teaches algorithms for optimizing for single task instances, and for optimizing for multiple task instances, depending on the determined assignment objective. *Id.* ¶¶ 108, 111–112. We are persuaded that the recited “one or more algorithms” includes these algorithms for a single task instance, as well as for multiple task instances. *Id.* ¶ 76 (teaching that example algorithms are provided in Queck that compare execution times). More specifically, we find that paragraph 76 teaches that allocation module 604 determines which assignment objective and which algorithm to use (the two algorithms correspond separately to the two assignment objectives) for assigning nodes based on characteristics of the task instance (e.g., whether there is a single task instance or multiple task instances are expected) and data partition to be deployed. *Id.*

We disagree with Patent Owner that “based on” in this passage “expressly refers to circumstantial variables used in the calculation for comparing whether a transfer would improve execution time,” and does not “refer to distinguishing between *different assignment objectives*.” PO Resp. 17. The full sentence teaches that the algorithm chosen for implementation is based on characteristics of the task instance and data partition, and that the resource information is used with the determined algorithm. *See* Ex. 1006 ¶ 76 (“The allocation module 604 receives the resource information from the resource manager 606 and implements one or more algorithms based on characteristics of the task instance and data partition to be deployed, using the resource information provided by the resource manager 606 (including,

e.g., the latency information)"); *see also id.* ¶ 111 (teaching that “the provisioning manager 134 may determine at the time the first task instance enters the queue 602 that, in fact, a total of ‘m’ task instances are expected”—a task instance characteristic). “Ultimately, the allocation module 604 may determine to deploy the task instance to a selected processing node, and may determine whether and where to deploy the associated data partition, as well.” *Id.* ¶ 77.

In further support of our findings, we also are persuaded by Petitioner’s argument that one of ordinary skill in the art would have found it obvious for central node 102 to “determine[] which among the ‘assignment objectives’ applies to a task instance to be deployed.” Pet. 30. In particular, we find persuasive Dr. Mir’s testimony that one of ordinary skill in the art would have been “motivated to enable provisioning manager 134 to determine an assignment objective from amongst the [disclosed] assignment objective(s) to give the system the flexibility to optimize goals relevant for different types of tasks.” Ex. 1003 ¶ 116. This flexibility is consistent with Queck’s teachings of the provisioning manager 134 being configurable to determine factors to decide how to deploy task instances and data partitions. Ex. 1006 ¶ 37.

We find unavailing Patent Owner’s argument that “this claim limitation requires an active step of ‘determining’ of ‘an assignment objective to be achieved in assigning compute nodes,” and that “[t]here is no such step of ‘determining’ disclosed in Queck.” PO Resp. 9–10; *see also id.* 12 (making same argument). As we find above, Queck teaches that its provisioning manager 134 determines which goal (i.e., assignment objective) for which to optimize. Ex. 1006 ¶ 37. In addition, Queck’s paragraph 76 teaches that allocation module 604—which is within provisioning manager

134—“implements one or more algorithms based on characteristics of the task instance and data partition to be deployed.” Ex. 1006 ¶ 76. Such a characteristic (e.g., whether “m” task instances are expected) may be determined at runtime by the provisioning manager 134; thus Queck teaches determining which goal and algorithm to implement. *Id.* ¶¶ 76, 111. In light of these teachings, we find unavailing Patent Owner’s argument that “Petitioner fail[s] to identify anything within the four corners of Queck directed to the active step of ‘determining’ *which one* of the purported ‘different goals’ to achieve.” PO Resp. 12 (citing Pet. 27, 31); Ex. 1006 ¶¶ 76, 111.

We also find unavailing Patent Owner’s argument that “Queck’s determining of a ‘number of factors to decide whether and how to deploy the task instance’ is not the required determining of an assignment objective,” but instead, “on its face only determines ‘a number of factors’ (i.e. not an assignment objective), and ‘whether to deploy the task instance’ (i.e. not an assignment objective).” PO Resp. 10 (quoting Ex. 1006 ¶ 37). Rather, as we find above, this passage from Queck teaches that the provisioning manager 134 determines factors to decide how to deploy the task instance and data partition; such deployment requires determining the assignment objective as there are different algorithms for each objective. Ex. 1006 ¶¶ 37; *see also id.* ¶¶ 76 (teaching that the allocation module 604 implements algorithms based on characteristics of the task instance and data partition to be deployed), 108, 112.

We also find unavailing Patent Owner’s argument that paragraph 76 of “Queck universally describes *all* the ‘different algorithms’ as being used to compare respective execution times of the same task with and without the transfer.” PO Resp. 14–15 (citing Ex. 1006 ¶ 76). Patent Owner adds that

“Queck does not describe the algorithms themselves as distinct or competing ‘assignment objectives.’” *Id.* at 15. Simply put, we do not agree with Patent Owner that paragraph 76’s second sentence limits its first sentence’s recital of “one or more algorithms” to algorithms that are “used to compare respective execution times of the same task with and without the transfer.” *Id.*; Ex. 1006 ¶ 76. Rather, we find that the second sentence describes certain “example algorithms” that are provided in Queck, but not all of the example algorithms. Ex. 1006 ¶ 76. Moreover, Patent Owner’s argument does not account for Queck’s example equation fourteen that looks at the accumulated total of the multiple associated task instances. *See id.* ¶¶ 111–112; *see also* Ex. 1003 ¶¶ 109–111 (testifying that optimizing a single task’s performance is different from optimizing the average performance of the task as a whole (*i.e.*, two or more task instances)).

In addition, we find inapposite Patent Owner’s argument that Queck describes optimizing execution time as the common goal for both a single task instance and multiple task instances. PO Resp. 12 (citing Ex. 1006 ¶ 111); *see also* PO Sur-reply 14 (making the same substantive argument expressed as both of the corresponding algorithms “evaluate whether there would be an execution improvement if there is a transfer of the data partition for completion of all task instance(s)”; 15–18 (characterizing the objectives as having the same overarching purpose in making the same substantive argument). Queck’s single task instance algorithm seeks to improve the execution time of the single task instance, while Queck’s multiple task instance algorithm seeks to improve the average performance (execution time) of a task as a whole. *Compare, e.g.*, Ex. 1006 ¶¶ 5–6, 37, 47, 87, *with id.* ¶¶ 111–112. This difference provides different assignment objectives, similar to the ’800 patent’s disclosure of having four different assignment

objectives (i.e., minimizing a total latency, minimizing a maximum latency, minimizing a total latency within a limit on maximum latency, and minimizing a maximum latency within a limit on total latency), despite all four relating to latency. *See* Ex. 1001, 7:10–17. We also find persuasive Dr. Mir’s testimony that one of ordinary skill in the art would have understood “that (1) optimizing a single task’s performance is different from (2) optimizing the average performance of the task as a whole (i.e., the performance of 2+ task instances),” and that “goal (1) allows the system to optimize the performance of a particular task instance, which may come at the expense of other task instances, but goal (2) allows the system to balance the demands of all instances of a task to optimize the performance of the task as a whole.” Ex. 1003 ¶ 110. This testimony is consistent with Queck’s equations and how they may be evaluated. *See* Ex. 1006 ¶¶ 108, 112.

We also find unavailing Patent Owner’s argument that “Petitioner relies on what is essentially the same equation in Queck for both the ‘single-task’ and ‘multi-task’ instances.” PO Sur-reply 13. Patent Owner adds that “comparing Queck’s Eq. (13) to Eq. (14) readily reveals that ‘*m*’ [in Eq. 14] is simply 1 for the single-task instance where, by definition, only one task instance is expected.” *Id.* at 14 (citing Ex. 1006 ¶ 112). Queck, however, presents equations 13 and 14 as separate algorithms, with and without “*m*” respectively, which evaluate differently. Ex. 1006 ¶¶ 108, 112. It is this difference which serves to achieve the difference in assignment objectives, which improve the performance of a single task instance, which may come at the expense of other task instances, versus improving the average performance of a task as a whole. Ex. 1006 ¶¶ 108, 111, 112; Ex. 1003 ¶¶ 110, 130–131. Notably, claim 1 does not limit the closeness of

algorithms associated with differing assignment objectives. Ex. 1001, 15:23–37.

We also find unavailing Patent Owner’s argument that this claim limitation recites “nodes,” in plural form, and thus, “is plainly distinguishable from allegedly pairing a *singular* processing node with a *singular* data node.” PO Resp. 13. Nor do we agree with Patent Owner that the language requires making multiple assignments at once between plural compute nodes and plural data nodes. *Id.* at 13–14. Rather, this limitation recites, in relevant part, “determining an assignment objective to be achieved in assigning” nodes. Ex. 1001, 15:28–30. In other words, it relates to determining what is the assignment objective. *Id.* Notably, this limitation does not require actually assigning any nodes. *Id.* On the other hand, the next limitation in claim 1 recites, in relevant part, “assigning *a* compute node of the set of compute nodes to *a* data node of the set of data nodes.” *Id.* at 15:31–32 (emphases added). In other words, claim 1 allows for assigning a singular processing node with a *singular* data node, contrary to Patent Owner’s argument. *See, e.g., Baldwin Graphic Sys., Inc. v. Siebert, Inc.*, 512 F.3d 1338, 1342 (Fed. Cir. 2008) (quoting *KCJ Corp. v. Kinetic Concepts, Inc.*, 223 F.3d 1351, 1356 (Fed. Cir. 2000)) (“This court has repeatedly emphasized that an indefinite article ‘a’ or ‘an’ in patent parlance carries the meaning of ‘one or more’ in open-ended claims containing the transitional phrase ‘comprising.’”). Moreover, Petitioner shows that Queck teaches assigning a compute node to a data node based on the assignment objective. *See infra* Section V(B)(4). And Petitioner shows that Queck teaches making a plurality of such assignments. *See supra* Section V(B)(1).

We also find unavailing Patent Owner’s arguments that Queck’s teaching “of ‘in order to improve an execution time of the task instance’

. . . is plainly and expressly hard-coded and by definition not ‘determined,’” and that Queck’s teaching “of considering ‘the accumulated performance of all associated task instances’ ‘to improve the average performance of a task as a whole’ . . . suffers from the same deficiency.” PO Resp. 10 (quoting Ex. 1006 ¶¶ 37, 111). As we discuss above, we are persuaded that Queck teaches determining an assignment objective from among at least two assignment objectives (i.e., (i) optimizing performance of a *single* task instance, and (ii) optimizing average performance across *multiple* instances of the same task). Hence, we disagree with Patent Owner that Queck teaches having only hard-coded single-objective assignment. *See* Ex. 2004, 17 n.1 (concluding that “[t]he plain-and-ordinary meaning only excludes hard-coded single-objective assignments”). Moreover, Patent Owner’s arguments do not account for the last part of the district court’s construction, which recites that the plain-and-ordinary meaning “does not, however, exclude a determination where there may only be a single assignment objective available when the ‘determination’ step is made.” PO Resp. 10; Ex. 2004, 17 n.1. Thus, even if only one assignment objective is taught—which is not the case—Queck’s teaching of determining the assignment objective (and which algorithm to use) teaches this limitation because Queck does so using factors and run-time characteristics in making the determination, rather than having a hard-coded single-objective assignment. *See* Ex. 1006 ¶¶ 37, 76, 111.

We also find unavailing Patent Owner’s argument that Queck’s teaching that the provisioning manager 134 “may be configured to determine a number of factors to decide whether and how to deploy the task instance 116 and the data partition 122a” teaches a hard-coded configuration. PO Resp. 9 (citing Pet. 25 (quoting Ex. 1006 ¶ 37), 30). Rather, this passage

from Queck teaches configuring the provisioning manager 134 to determine factors to decide how to deploy a task instance and data partition. Ex. 1006 ¶ 37. Thus, as we discuss above, Queck teaches determining an assignment objective and its algorithm, which does not implicate the exclusion of a hard-coded single-objective assignment. *Id.*

Patent Owner has not explained sufficiently why it contends that Queck’s teaching that provisioning manager 134 “may be configured” connotes a “hard-coded configuration.” PO Resp. 9. As Petitioner notes, the ’800 patent discloses that determining the assignment objective to be applied “may be made, for example, based on . . . a cloud operator configuration.” Pet. 30 (citing Ex. 1001, 8:25–27) (alteration in original, emphasis omitted); Ex. 1001, 8:21–27.

We also find unavailing Patent Owner’s arguments concerning that Queck’s algorithms “can and do result in a decision to not transfer the partition,” and “[d]etermining *whether* to transfer, and hence at times not transferring at all, is plainly distinguishable from ‘determining an assignment objective . . . *in assigning*.’” PO Resp. 11–12 (citing Ex. 1006 ¶ 111). First, the claim language recites “determining an assignment objective *to be achieved* in assigning,” and thus, the determining step is separate from the assigning step. Ex. 1001, 15:28–30 (emphasis added). Second, we agree with Petitioner and find that “a determination to not change an assignment *is* a determination based on an assignment objective—the objective may dictate that the ‘goal’ of the particular objective is best accomplished by leaving the then-current assignment in place.” Pet. Reply 16. Third, even if sometimes Queck’s algorithms result in a decision to not transfer a partition is of no moment, because, as we discuss above, Queck teaches a plurality of deployments of task instances and data partitions. *See*

supra Section V(B)(1). Moreover, “prior art that sometimes meet[s] the claim elements [is] sufficient to show obviousness.” *See Unwired Planet, LLC v. Google Inc.*, 841 F.3d 995, 1002 (Fed. Cir. 2016).

We also find unavailing Patent Owner’s argument that Petitioner improperly equates determining an assignment objective with Queck’s teaching that “allocation module 604 *determines which of the one or more algorithms* . . . to implement based on characteristics of the task instance and data partition to be deployed (e.g., whether there is a single task instance or multiple task instances are expected).” PO Sur-reply 8 (citing Pet. Reply 14–15 (citing Paper 8, 41; Ex. 1006 ¶¶ 76, 108, 112)) (alterations in original). As we discuss above, Queck teaches the claimed “determining an assignment objective to be achieved,” including via its teaching that allocation module 604 determines which algorithm to implement based on characteristics of the task instance and data partition to be deployed. Ex. 1006 ¶ 76. More specifically, and as we discuss above, Queck teaches determining the assignment objective and its algorithm to implement based on the characteristics (e.g., a single task instance or “‘m’ task instances are expected”). *See id.* ¶¶ 76, 111. Regardless, determining which algorithm to implement also teaches determining the assignment objective because each of the two assignment objectives has its own algorithm. *See id.* ¶¶ 106, 108, 111–112.

We also disagree with Patent Owner that Petitioner “in its Reply . . . incorrect[ly] argu[es] that Queck’s selective application of an *algorithm* anticipates” determining an assignment objective in the same manner as contemplated by the ’800 patent. PO Sur-reply 8 (citing Pet. Reply 14). Simply put, Petitioner’s argument relates to determining the assignment objective and its algorithm, rather than the “application” of the algorithm, as

Patent Owner argues. *See* Pet. 14. In other words, Petitioner does not argue that the algorithm is applied as part of teaching this limitation. *Id.* at 14–16. Thus, Patent Owner’s arguments concerning Petitioner conflating “readily distinguishable acts” are inapposite as they are based on an incorrect premise. PO Sur-reply 8–12.

In sum, we are persuaded that Petitioner demonstrates by a preponderance of the evidence that Queck teaches “determining an assignment objective to be achieved in assigning compute nodes from the set of compute nodes to data nodes from the set of data nodes.”

4. *Assigning a Compute Node*

Claim 1 further recites “assigning a compute node of the set of compute nodes to a data node of the set of data nodes based on the assignment objective and a plurality of costs associated with the set of edges.” Ex. 1001, 15:31–34. Petitioner argues that “Queck teaches assigning a processing node (a ‘compute node’) of a set of processing nodes to a data node of a set of data nodes.” Pet. 34 (citing Pet. 17–21) (referring to its arguments for the preamble). “For example, in Figure 2 of Queck, processing node 104f is assigned to data node 104e,” according to Petitioner. *Id.* (citing Ex. 1006 ¶ 49). Petitioner argues that “Queck further teaches that this assignment is based on an optimization goal (the claimed ‘assignment objective’), such as optimizing the performance for a single task instance or multiple tasks instances.” *Id.* (citing Ex. 1006 ¶¶ 111–112).

Petitioner argues that “Queck teaches ‘costs’ associated with these edges because it discloses latencies between pairs of nodes (*e.g.*, the values stored at the different positions in latency matrix 610).” *Id.* at 35 (citing Ex. 1006 ¶ 75; Ex. 1003 ¶ 125). Petitioner argues that Queck teaches “node assignment based on these latencies.” *Id.* In particular, Petitioner argues

that Queck teaches in deploying task instance 116b and data partition 122a to processing node 104f and data node 104e, respectively, “it may be assumed that a latency (and/or an execution time) experienced by the task instance 116b may be reduced/improved when the task instance 116b makes calls to the data node 104e, as compared to when the task instance 116b makes data calls to the data/initial node 104d.” *Id.* (quoting Ex. 1006 ¶ 49).

After reviewing Petitioner’s arguments and evidence, including Dr. Mir’s testimony, which are not addressed by Patent Owner (*see generally* PO Resp.), we are persuaded that Petitioner demonstrates by a preponderance of the evidence that Queck teaches “assigning a compute node of the set of compute nodes to a data node of the set of data nodes based on the assignment objective and a plurality of costs associated with the set of edges.”

5. *Wherein the Assignment Objective*

Claim 1 further recites “wherein the assignment objective is based on a latency cost used in obtaining a set of assignments by applying an algorithm.” Ex. 1001, 15:35–37. Petitioner argues that Queck teaches “at least two ‘assignment objective[s],’ e.g., optimizing performance (execution time) for a single task instance or multiple task instances.” Pet. 35 (alteration in original). And Petitioner argues that “both of those objectives are ‘based on a latency cost used in obtaining a set of assignments by applying an algorithm.’” *Id.* at 36 (citing Ex. 1006 ¶¶ 85–86, 108–109, 111–112). Petitioner argues that “[f]or example, Queck discloses that the algorithm associated with the goal of optimizing a single task instance includes the consideration of several factors, including ‘a latency associated with one or more pairs of the nodes 104a, 104b.’” *Id.* (quoting Ex. 1006 ¶ 37).

In addition, Petitioner argues that Queck teaches “that the algorithm associated with optimizing the performance of multiple tasks includes evaluation” of an inequality having a variable l_{i0} that “is the latency between the node i (the compute node) and central node 102 (the data node).” *Id.* (citing Ex. 1006 ¶¶ 104, 112). Petitioner adds that “Queck teaches that the algorithm associated with the goal of minimizing latency so long as the node assignment improves execution time includes determining the processing/data node pairs with the ‘lowest latency.’” *Id.* (citing Ex. 1006 ¶ 85).

Patent Owner argues for the first time in its Sur-reply that Queck fails to teach “obtaining a set of assignments by applying an algorithm,” as recited in this limitation. PO Sur-reply 5–8; *see generally* PO Resp. These are new arguments, and cannot properly be raised in Patent Owner’s Sur-reply. *See* Paper 9 (Scheduling Order), 8 (“Patent Owner is cautioned that any arguments not raised in the response may be deemed waived.”). Nor does Petitioner address this limitation in its Reply. *See generally* Pet. Reply. Accordingly, these new arguments, raised for the first time in Patent Owner’s Sur-reply, will not be considered. *See* 37 C.F.R. § 42.23(b) (“A sur-reply may only respond to arguments raised in the corresponding reply . . .”).

After reviewing Petitioner’s arguments and evidence, which are not addressed by Patent Owner in its Response (*see generally* PO Resp.), we are persuaded that Petitioner demonstrates by a preponderance of the evidence that Queck teaches “wherein the assignment objective is based on a latency cost used in obtaining a set of assignments by applying an algorithm.”

6. *Summary*

In summary, based on the arguments and evidence of record, we find that Petitioner demonstrates by a preponderance of the evidence that claim 1 is unpatentable under 35 U.S.C. § 103(a) in view of Queck.

C. *Challenged Claim 6*

Claim 6 depends from independent claim 1, and adds the following limitations:

weighting an initial cost of an edge of the set of edges based on an amount of data to be processed from a data node associated with the edge to produce a weighted cost of the edge, wherein the plurality of costs associated with the set of edges comprises the weighted cost of the edge.

Ex. 1001, 16:17–23. Petitioner argues that Queck teaches these additional limitations. Pet. 38–39; Pet. Reply 18–22. Patent Owner disagrees. PO Resp. 21–22; PO Sur-reply 18–21. For the reasons provided below, we agree with Petitioner and find that Queck teaches the limitations of claim 6.

For reference, the '800 patent discloses that possible assignments between the data nodes and the compute nodes are termed edges, and each edge “may be associated with a cost representing, for example, a latency or network distance between the two nodes.” Ex. 1001, 6:49–54. The '800 patent adds that “the cost may also be weighted by the amount of data to be processed, thereby accounting for increased network traffic due to a higher volume of data.” *Id.* at 6:54–57.

As to Queck, we find that Queck teaches determining latency values “between each pair of nodes in the grid computing environment,” and that these latency values “may be stored in a latency matrix 610 or other format.” Ex. 1006 ¶ 75. We find that these latencies between the different nodes comprise edges, and reflect a cost associated with each edge. *Id.*; *see also*

Ex. 1003 ¶ 100 (testifying that each position in the latency matrix represents a possible connection between two nodes, and that “[t]he positions in the latency matrix therefore teach the ‘edges’ (possible assignments) between processing and data nodes”). In addition, we find that Queck teaches that “a size of the latency *and/or a quantity or frequency of the data calls*” may be considered in weighting the cost of the edges. Ex. 1006 ¶ 28 (emphases added). As to the latter, Queck explains that data calls are associated with “corresponding data (and corresponding returns of the called data back to the task instances).” *Id.* ¶ 16.

We agree with Petitioner and find that Queck’s teaching concerning “the number of data calls between a compute node and a data node reflects the ‘amount of data that is processed from [the] data node’ . . . because Queck expressly uses, as part of its cost-benefit calculation, the number of data calls to take into account (i.e., to ‘weight’ . . .) the aggregate amount of data provided from the data node’s data call responses to requests by the compute node for processing such data.” Pet. Reply 20 (citing Ex. 1020 ¶ 25). More specifically, Queck provides equations for use in assessing the cost of deploying a particular compute node and/or data node, which weight an amount of data to be processed. *See, e.g.*, Ex. 1006 ¶¶ 95, 105.

More specifically, Queck provides equation 1, which represents “[a]n entire execution time ‘t,’ including the deployment of the task instance on node i and the transfer of the necessary data partition to node j.” *Id.* ¶ 95; *see also id.* ¶¶ 93–94 (explaining the notation used and assumptions made for the equations). Equation 1 reads as follows:

$$t_{ij} = \max(t_{Tbin-i}, t_{Tpart-j}) + t_{exe} + \sum_{k=1}^c t_{call-kij}$$

Id. ¶ 95. We agree with Petitioner and find that the last component of equation 1 represents the “summation of the *times needed for all database calls* between nodes ‘i’ and ‘j.’” *Id.* ¶ 97 (emphasis added); *see also id.* ¶ 93 (explaining notation); Ex. 1020 ¶ 27; Pet. Reply 21.

Furthermore, Queck provides equation 8, which represents the time required for each such data call. Ex. 1006 ¶ 105. Equation 8 reads as follows:

$$t_{data-ki0} = l_{i0} + s_{data-k}/b + l_{0i}$$

Id. We agree with Petitioner and find that equation 8 “includes two latency terms “*l_{i0}*” and “*l_{0i}*” to account for latencies in the request and response for the data call(s)’ *as well as* ‘the size *s* of the data (e.g., in bytes)’ of the data call [(i.e., *s_{data-k}*)] over ‘the bandwidth “*b*.”” Pet. Reply 21; Ex. 1006 ¶¶ 104–105; Ex. 1020 ¶ 28. We also agree with Petitioner and find that “[t]hus, the summation of all *s_{data-k}* for each data call evidences Queck’s ‘weighting’ of the cost (i.e., time) of an edge ‘based on an amount of data to be processed from a data node’ (in addition to weighting based on latency).” Pet. Reply 21; Ex. 1006 ¶¶ 104–105; Ex. 1020 ¶ 28. And we credit the testimony that we cite here of Dr. Mir from his supplemental declaration, as it is consistent with Queck’s teachings. *Compare* Ex. 1020 ¶¶ 25, 27–28, *with* Ex. 1006 ¶¶ 93–95, 97, 104–105.

We find unavailing Patent Owner’s argument that “Petitioner fails to explain how *either* the ‘size of the latency’ *or* the ‘quantity or frequency of the data calls’ (or even their combination) is a weight ‘based on an amount of data to be processed from a data node.’” PO Resp. 22. As we discuss above, Petitioner explains how Queck’s disclosure regarding the “quantity or frequency of the data calls” teaches weighting “based on an amount of data

to be processed from a data node,” including by Petitioner walking through the terms of equations 1 and 8. *See* Pet. Reply 18–22; Ex. 1006 ¶¶ 93–95, 97, 104–105.

We also find unavailing Patent Owner’s argument that “the ’800 patent expressly distinguishes the quantification of ‘latency . . . *between* two nodes’ from the *weights* applied thereto based on ‘the amount of data to be processed.’” PO Resp. 22 (citing Ex. 1001, 6:53–57) (alteration in original); PO Sur-reply 20 (making the same argument). As we discuss above, Queck teaches weighting based on the amount of data to be processed. Ex. 1006 ¶¶ 93–95, 97, 104–105. And, as we discuss above, Queck teaches that “a size of the latency *and/or* a quantity or frequency of the data calls” may be considered in weighting the cost of the edges. Ex. 1006 ¶ 28 (emphasis added). In other words, Queck teaches separately weighting based on the amount of data to be processed. *Id.*

In sum, we are persuaded by a preponderance of the evidence that Queck teaches the limitations on claim 6. And based on the arguments and evidence of record, we find that Petitioner demonstrates by a preponderance of the evidence that claim 6 is unpatentable under 35 U.S.C. § 103(a) over Queck.⁸

⁸ In our Decision on Institution, we stated that “[b]ased on the current record, we are not persuaded that Queck teaches the limitations of claim 6.” Dec. on Inst. 47. However, in light of the full record, we have changed our view of the merits. *See Trivascular, Inc. v. Samuels*, 812 F.3d 1056, 1068 (Fed. Cir. 2016) (“At th[e Institution] point, the Board is considering the matter preliminarily without the benefit of a full record. The Board is free to change its view of the merits after further development of the record”); Dec. on Inst. 52–53 (“We have not made a final determination with respect to the patentability of any challenged claim. Our final determination will be based on the record as fully developed during trial”).

D. Challenged Claims 2, 7–9, 13, 14, and 18–21

Petitioner argues, with specific cites to Queck, as well as Dr. Mir’s testimony, that Queck teaches the limitations recited in claims 2, 7–9, 13, 14, and 18–21. Pet. 37–46. Patent Owner does not separately address Petitioner’s arguments directed to these claims. PO Resp. 17–18.

Based on the evidence and arguments of record, we find that Petitioner demonstrates by a preponderance of the evidence that claims 2, 7–9, 13, 14, and 18–21 are unpatentable under 35 U.S.C. § 103(a) over Queck.

VI. REMAINING INSTITUTED GROUNDS

Petitioner argues that (i) Queck anticipates claims 1, 2, 7–9, 13–14, and 19–21; (ii) the combination of Queck and Singh renders obvious claims 1, 2, 6–9, 13, 14, and 18–21; and (iii) the combination of Li and Singh renders obvious claims 1, 8, 9, 13, 20, and 21 of the ’800 patent. Pet. 6, 12–65. Thus, these grounds of unpatentability challenge the same claims we already determine are unpatentable over Queck. *See supra* Section V (determining Petitioner shows that all of the challenged claims are unpatentable). Under the circumstances of this case, analyzing additional grounds challenging the same claims, which we have determined to be unpatentable, would not be an efficient use of the Board’s time and resources. *See Bos. Sci. Scimed, Inc. v. Cook Grp. Inc.*, 809 F. App’x 984, 990 (Fed. Cir. 2020) (“We agree that the Board need not address issues that are not necessary to the resolution of the proceeding.”).

Accordingly, we do not reach these remaining grounds. *Cf. In re Gleave*, 560 F.3d 1331, 1338 (Fed. Cir. 2009) (not reaching other grounds of unpatentability after affirming the anticipation ground); *see also Beloit*

Corp. v. Valmet Oy, 742 F.2d 1421, 1423 (Fed. Cir. 1984) (determining once a dispositive issue is decided, there is no need to decide other issues).

VII. CONCLUSION⁹

Based on the full record, we determine that Petitioner shows by a preponderance of the evidence that claims 1, 2, 6–9, 13, 14, and 18–21 are unpatentable under 35 U.S.C. § 103(a) in view of Queck.

Claim(s)	35 U.S.C. §	Reference(s) /Basis	Claims Shown Unpatentable	Claims Not Shown Unpatentable
1, 2, 6–9, 13, 14, 18– 21	103(a)	Queck	1, 2, 6–9, 13, 14, 18–21	
1, 2, 7–9, 13, 14, 19–21	102(b) ¹⁰	Queck		
1, 2, 6–9, 13, 14, 18– 21	103(a) ¹¹	Queck, Singh		

⁹ Should Patent Owner wish to pursue amendment of the challenged claims in a reissue or reexamination proceeding subsequent to the issuance of this decision, we draw Patent Owner’s attention to the April 2019 *Notice Regarding Options for Amendments by Patent Owner Through Reissue or Reexamination During a Pending AIA Trial Proceeding*. See 84 Fed. Reg. 16,654 (Apr. 22, 2019). If Patent Owner chooses to file a reissue application or a request for reexamination of the challenged patent, we remind Patent Owner of its continuing obligation to notify the Board of any such related matters in updated mandatory notices. See 37 C.F.R. § 42.8(a)(3), (b)(2).

¹⁰ As we explain above, because we determine that the claims challenged in this ground are unpatentable under § 103(a) over Queck, we decline to address this ground.

¹¹ As we explain above, because we determine that the claims challenged in this ground are unpatentable under § 103(a) over Queck, we decline to address this ground.

1, 8, 9, 13, 20, 21	103(a) ¹²	Li, Singh		
Overall Outcome			1, 2, 6–9, 13, 14, 18–21	

VIII. ORDER

In consideration of the foregoing, it is hereby

ORDERED that, pursuant to 35 U.S.C. § 314(a), Petitioner has shown by a preponderance of the evidence that claims 1, 2, 6–9, 13, 14, and 18–21 of the '800 patent are unpatentable; and

FURTHER ORDERED that parties to the proceeding seeking judicial review of this Final Written Decision must comply with the notice and service requirements of 37 C.F.R. § 90.2.

¹² As we explain above, because we determine that the claims challenged in this ground are unpatentable under § 103(a) over Queck, we decline to address this ground.

IPR2021-00572
Patent 9,164,800 B2

For PETITIONER:

Y. Ernest Hsin
Jaysen Chung
Omar Amin
GIBSON, DUNN & CRUTCHER LLP
EHsin@gibsondunn.com
JSChung@gibsondunn.com
OAmin@gibsondunn.com

For PATENT OWNER:

Ryan Loveless
Jeffrey Stephens
Brett Mangrum
James Etheridge
Brian Koide
Jeffrey Huang
ETHERIDGE LAW GROUP
ryan@etheridgelaw.com
jstephens@etheridgelaw.com
brett@etheridgelaw.com
jim@etheridgelaw.com
brian@etheridgelaw.com
jeff@etheridgelaw.com