

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

EMC CORPORATION and
EMC INTERNATIONAL U.S. HOLDINGS, INC.,
Petitioner,

v.

CLOUDING CORP.,
Patent Owner.

Case IPR2015-01923
Patent 6,631,449

Before JAMESON LEE, MICHAEL W. KIM, and RAMA G. ELLURU,
Administrative Patent Judges.

ELLURU, *Administrative Patent Judge.*

FINAL WRITTEN DECISION
35 U.S.C. § 318(a) and 37 C.F.R. § 42.73

I. INTRODUCTION

EMC Corporation and EMC International U.S. Holdings, Inc., (collectively “Petitioner”) filed a petition (“Pet.”) to institute an *inter partes* review of claims 1, 2, and 5 of U.S. Patent No. 6,631,449 B1 (Ex. 1001, the “’449 patent”). Paper 1. Patent Owner, Clouding Corp., filed a Preliminary Response (“Prelim. Resp.”). Paper 8. On March 16, 2016, we instituted an *inter partes* review of claims 1, 2, and 5 of the ’449 patent. Paper 9, 23 (“Dec.”). Patent Owner timely filed a Response to the Petition. Paper 13 (“PO Resp.”). Petitioner subsequently timely filed a Reply to Patent Owner’s Response. Paper 18 (“Pet. Reply”).

An oral hearing for this proceeding was held on November 16, 2016, a transcript of which has been entered in the record. Paper 27 (“Tr.”)

We have jurisdiction under 35 U.S.C. § 6. This Final Written Decision is issued pursuant to 35 U.S.C. § 318(a) and 37 C.F.R. § 42.73.

For the reasons that follow, we determine that Petitioner has shown by a preponderance of the evidence that claims 1, 2, and 5 of the ’449 patent are unpatentable.

A. *Related Proceedings*

The ’449 patent was previously the subject of IPR2013-00089 and IPR2013-00321, which were terminated due to settlement. IPR2013-00089, Paper 21, 3; IPR2013-00321, Paper 15, 3. Petitioner and Patent Owner assert collectively that the ’449 patent is asserted in *Clouding Corp. v. EMC Corp.*, C.A. No. 14-cv-01178 (LPS) (D. Del.), and against another party in another district court case. Pet. 3; Paper 4, 2.

B. *The ’449 Patent (Ex. 1001)*

The ’449 patent sets forth (Ex. 1001, 1:13–18):

The invention relates generally to systems and methods for maintaining storage object consistency across distributed storage networks and more particularly to a dynamic distributed data system and method which provides for minimized latency, reduced complexity and is extensible, scalable and isotropic.

As explained in the '449 patent, when a processor requests data, and the data is not in its cache, a main memory supplies the data and a copy then is stored in the processor cache. *Id.* at 1:24–26. In multiprocessor configurations, each processor cache can have its own copy of the same data. *Id.* at 1:31–33. When the data is updated in the main memory, however, one processor cache may have copied data from the main memory prior to the update, while another processor cache may have copied data from the main memory after the update. *Id.* at 1:33–34. Accordingly, inconsistencies can be present across processor caches for the same data. *Id.* at 1:30–31, 36–37. One prior solution uses a snoopy cache coherence protocol, where each processor keeps track of the data on other processors. *Id.* at 1:43–49. Performance typically is limited by the number of processors a shared interconnect can accommodate, however, making it difficult to scale or extend. *Id.* at 2:7–17. Another prior solution uses a directory-based cache coherence protocol, where a processor periodically performs a table lookup in a directory to determine whether its cache needs to be updated. *Id.* at 2:18–22. A directory based scheme is not scalable or extensible, however, because each cache needs to keep a directory entry for other interested caches in the system and as new caches are added, the existing tables must be expanded. *Id.* at 2:30–33.

The present invention purportedly integrates aspects of snoopy cache coherence and directory-based protocols and extends an integrated protocol to arbitrary network topologies. *Id.* at 3:41–46. The snoopy protocol is utilized over a dynamically-created-and-maintained tree of interested nodes.

Id. at 3:57–58. And the architecture operates in a manner such that each node knows only about its own storage objects and its connections to its neighbors. *Id.* at 3:58–61. “Storage objects include any consistent storage entity such as a file, a block or an extent which may need to be updated atomically.” *Id.* at 5:19–21.

In addition, the invention utilizes a facility called the repository of last resort “RLR.” *Id.* at 4:56–57. “The RLR provides a facility for storing a last or only remaining copy of a data replica that may not be deleted. In this manner the RLR ensures the persistence of the data replica.” *Id.* at 4:57–60. Every node has a routing table, which provides a pointer indicating a direction to the RLR. *Id.* at 5:4–6, 13–14. Thus, in terms of network topology, each node knows only about itself and its neighbors. *Id.* at 5:14–15. Moreover, the process monitors data requests to the RLR to determine whether to migrate the RLR to a requesting node. *Id.* at 3:17–19.

C. Illustrative Claims

We instituted a review of claims 1, 2, and 5. Claims 1 and 5 are independent claims, and claim 2 depends from claim 1. Claim 1 is illustrative of the claimed subject matter and is reproduced below:

1. A method for maintaining storage object consistency among nodes in a distributed storage architecture including a repository of last resort, the method comprising the steps of:

(a) monitoring storage object requests to the repository of last resort;

(b) deciding whether to migrate the repository of last resort;
and

(c) migrating the repository of last resort.

D. Ground of Unpatentability Instituted for Trial

We instituted trial based on the following asserted references.

Reference	Patent	Date	Exhibit
Baylor	U.S. Patent No. 5,893,922	April 13, 1999	Ex. 1010
Herlihy	Michael J. Demmer & Maurice P. Herlihy, <i>The Arrow Distributed Directory Protocol</i> , 119–33	1998	Ex. 1009
Borrill	Paul L. Borrill & Peter Z. Onufryk, <i>Extending Snoopy Cache Consistency to General Interconnection Networks</i> , IEEE Int’l Conf. Indus. Tech., 752–56	Dec. 1994	Ex. 1008
Futurebus	ANSI/IEEE, <i>Information technology – Microprocessor systems – Futurebus+ – Logical protocol specification</i> (“Futurebus+”), International Standard, ISO/IEC 10857, ANSI/IEEE Std. 896.1, First Edition, 11–212	April 27, 1994	Ex. 1011

Petitioner provides testimony by Dr. Paul F. Reynolds, Jr. in a Declaration (“Reynolds Decl.”) (Ex. 1022) and Reply Declaration (“Reynolds Reply Decl.”) (Ex. 1025) in support of its Petition. We instituted review of claims 1, 2, and 5 based on the following grounds. Dec. 23.

Reference(s)	Basis	Claims challenged
Baylor and Herlihy	§ 103	1, 2, and 5
Borrill, Futurebus, and Baylor	§ 103	1, 2, and 5

II. ANALYSIS

A. Claim Construction

In an *inter partes* review, claim terms in an unexpired patent are interpreted according to their broadest reasonable constructions in light of the specification of the patent in which they appear. *See* 37 C.F.R. § 42.100(b); *Cuozzo Speed Techs., LLC v. Lee*, 136 S. Ct. 2131, 2146 (2016). Under the broadest reasonable construction standard, claim terms are presumed to have their ordinary and customary meaning, as would be understood by one of ordinary skill in the art in the context of the entire disclosure. *In re Translogic Tech., Inc.*, 504 F.3d 1249, 1257 (Fed. Cir. 2007). “Absent claim language carrying a narrow meaning, the PTO should only limit the claim based on the specification . . . when [it] expressly disclaim[s] the broader definition.” *In re Bigio*, 381 F.3d 1320, 1325 (Fed. Cir. 2004). “Although an inventor is indeed free to define the specific terms used to describe his or her invention, this must be done with reasonable clarity, deliberateness, and precision.” *In re Paulsen*, 30 F.3d 1475, 1480 (Fed. Cir. 1994).

We determine, for purpose of this Final Written Decision, that the only term that require express construction is “repository of last resort” (or “RLR”). *See, e.g., Wellman, Inc. v. Eastman Chem. Co.*, 642 F.3d 1355, 1361 (Fed. Cir. 2011) (“[C]laim terms need only be construed ‘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))).

All three challenged claims recite “repository of last resort.” In our Decision on Institution, based on the parties’ proposed constructions and the disclosure of the ’449 patent specification, we interpreted “repository of last resort” as a “facility for storing a last or only remaining data replica that may not be deleted.” Dec. 6; Ex. 1001, 3:12–19, 4:56–61. In their papers, Petitioner and Patent Owner indicate agreement with this construction. Pet. 16; Pet. Reply 3; PO Resp. 14 (citations omitted) (“An RLR ‘provides a facility for storing a last or only remaining data replica that may not be deleted.’”); Tr. 49:9–11 (Patent Owner stating “[t]he RLR, as you pointed out at the beginning of the proceeding this morning, is a logical construct. It is a location. It is not the data.”).

At the oral hearing, Patent Owner argued that it advocates a construction for “repository of last resort” that is different from, or, at least, has a substantive distinction with, the construction we adopted in our Decision on Institution. Tr. 55:1–6. Specifically, Patent Owner argued at the oral hearing that a RLR, pursuant to the ’449 patent, must be separate and distinct from ownership. Tr. 54:16–56:20. According to Patent Owner, “[o]wnership of a data replica is an attribute associated with *write access* to that data replica.” PO Resp. 1 (emphasis added); Ex. 1022 ¶ 34 (Dr. Reynolds opines that “owner” would have been understood to refer to “a node that holds a copy of data for the purpose of writing”). Petitioner objected to Patent Owner’s assertion as new argument, not provided in the Patent Owner Response. Tr. 59:10–60:12. In response (*id.* at 72:16–20), Patent Owner referred to page 20 of the Patent Owner Response, wherein Patent Owner states “[w]hile the [prior art] references may describe moving the *owner* of data among nodes, a node’s

owner and its RLR (as described in the '449 patent) are not one and the same.” PO Resp. 20. We do not view this statement, made when describing prior art references, as proposing a construction for “repository for last resort.” Our determination is informed by the section of the Patent Owner Response specifically directed to the construction of this claim term, which states that “the '449 patent provides an unambiguous definition for the term repository of last resort,” and which provides the same construction expressly provided for in the '449 patent and adopted in our Decision on Institution. PO Resp. 14. *See SAS Institute, Inc. v. ComplementSoft, LLC*, 825 F.3d 1341, 1351 (Fed. Cir. 2016) (The Board may not change a claim interpretation from the institution decision where a party did not anticipate that “already-interpreted terms were actually moving targets.”).

Nevertheless, we are not sufficiently persuaded by Patent Owner’s argument that the '449 patent expressly excludes a RLR from also being an owner of data. When arguing that a particular prior art reference does not disclose a “repository of last resort,” Patent Owner refers to disclosure from the '449 patent specification that allegedly distinguishes “the owner [of a storage object] from the RLR.” PO Resp. 23 (citing Ex. 1001, 6:33–45; Fig. 4); Tr. 56:12–20. The disclosure cited at column 6, lines 33–45 of the '449 patent describes an embodiment in which a node requests “ownership” of a storage object, i.e., write access to the storage object. This disclosure, however, does not state that a node that has write access privileges to a data replica, i.e., “ownership,” cannot also be a “facility for storing a last or only remaining data replica that may not be deleted,” i.e., a “repository of last resort.” Similarly, Figure 4 illustrates a “Routing Table in accordance with *one embodiment* of the invention,” wherein a storage object has bits for a “ownership pointer” and a “RLR pointer.” Ex. 1001, 3:29–30 (emphasis

added); 5:16–19. In this particular embodiment, the '449 patent does disclose that the node that has ownership of the illustrated storage object is separate from the node that is the repository of last resort for that object, given that the pointers to the two nodes are different. *Id.* at Fig. 4; *see also id.* at 6:26–49 (Figure 6 distinguishing owner node from RLR). We decline, however, to limit the scope of the claim as Patent Owner proposes, based on a preferred embodiment alone, such that a node with write access privileges to a storage object cannot also be a facility for storing a last or remaining copy of a data replica that may not be deleted. *See Phillips v. AWH Corp.*, 415 F.3d 1303, 1323 (Fed. Cir. 2005) (en banc) (“[A]lthough the specification often describes very specific embodiments of the invention, we have repeatedly warned against confining the claims to those embodiments.”). Neither the '449 patent specification nor the claim language separates the concepts of RLR and ownership. *See* Tr. 60:18–61:1 (Petitioner arguing that the '449 patent specification and challenged claims do not narrow the scope of RLR to exclude ownership).

Indeed, Patent Owner’s expert, Dr. Nettles, agrees that, pursuant to the '449 patent specification, a RLR can also be an owner of a storage object.

Q. Okay. Is it your opinion that a repository of last resort can never be both an owner and an RLR?

A. No. My understanding is that the same storage object can be the RLR and can be the owner, but that’s not necessarily true, and that those two ideas are independent.

I think the way it typically works in this system is that when the object is first constructed, the RLR and the owner are the same object.

Q. All right. But you agree the ['449] patent contemplates that the same node can in fact be an RLR and an owner?

A. Yes, sir. *The same node can be an RLR and an owner, but it doesn't have to be.*

Ex. 1026, 58:14–59:2 (emphasis added). Patent Owner also confirmed at the oral hearing that the RLR and the owner of data can be the same location:

JUDGE ELLURU: So that can be -- the RLR and the owner can be the same location?

MR. FAHMI: It can be. In fact, when the data is first brought into the system, it will be, and Dr. Nettles confirmed this. But the RLR itself is not the owner, and we know this because they're pointed to and represented by different data structures for each individual storage object. And to answer your earlier question --

JUDGE ELLURU: So your position is that it may be, but they don't have to be the same.

MR. FAHMI: Correct. . . .

Tr. 47:6–16.

Therefore, we construe “repository of last resort” as “a facility for storing a last or only remaining data replica that may not be deleted.”

B. *Baylor and Herlihy*

As noted above, we instituted review of claims 1, 2, and 5, as having been obvious under 35 U.S.C. § 103 over the combination of Baylor and Herlihy. Dec. 23.

1. *Baylor*

Baylor describes a mechanism to improve the performance of parallel applications by migrating a home node of a global page to a more suitable node. Ex. 1010, Abstract, 2:9–13. Each node may be a uniprocessor or a multiprocessor. *Id.* at 3:7; see Fig. 1.

Baylor discloses that each global page, which includes one or more data items, is assigned a “static” home node at its initialization by using a distribution function (e.g., successive pages are assigned to the nodes in the node-set in a round robin manner, to get even distribution). *Id.* at 1:42–43, 3:10–14. “The static home node of a global page does not change during the life of the application program.” *Id.* at 3:14–16. According to Baylor, “[t]he static home is responsible for bringing pages in and out of the backing store.” *Id.* at 3:16–17.¹

Each global page also is assigned a “dynamic” home node. *Id.* at 3:21–22. Sometimes the same node may act as both the “static” and “dynamic” home node for a global page. *Id.* at 3:24–25. The dynamic home node enforces the “Invalidation Protocol,” described below, to maintain the coherence of each line. *Id.* at 3:28–30. Unlike the static home node, Baylor’s dynamic home for a global page may change from time to time. *Id.* at 3:22–23. Moreover, the “static” home “always remembers” the location of the dynamic home node for its global page. *Id.* at 4:17–18.

A “client node” for a global page is any node that is accessing one or more lines of that global page. *Id.* at 3:31–33. When a client node does not have a valid copy of a line that it needs, it sends a request message to the dynamic home node of that page. *Id.* at 3:37–39. The request indicates whether the client needs a shared or exclusive copy of the line. *Id.* at 3:39–41. If the request is for a shared copy, the dynamic home responds with a

¹ Petitioner explained at the hearing that the “backing store” is “longer term, less accessible storage.” Tr. 11:18–12:2. Patent Owner did not disagree with this explanation.

copy of the line. *Id.* at 3:42–43. If the line is held by another client in an exclusive state, the dynamic home first retracts the exclusive state, by communicating with that client, and then sends the data to the requester. *Id.* at 3:43–46. If the request is for an exclusive copy, the dynamic home first invalidates copies of the line in all clients by communicating with them. *Id.* at 3:48–50. The dynamic home node then sends the data to the requesting client node in exclusive mode, and it updates the directory to reflect this state. *Id.* at 3:50–51.

Baylor further describes a method of migrating the dynamic home node of a global page, as follows. *Id.* at 2:42–44. A client node maintains the following information about a global page that is mapped into its memory—the number of requests for the global page sent to the dynamic home node, referred to as the “Consultation Count.” *Id.* at 4:11–18. Furthermore, when a dynamic home node receives a client request, it services the request and also updates its list of consultation counts. *Id.* at 5:8–12. “The consultation count of a client node reflects the amount of traffic between the client and the dynamic home node.” *Id.* at 5:27–29. Thus, for example, “[i]f the consultation count of a client node for the global page is greater than a selected threshold, then the dynamic home node is migrated to the client node.” *Id.* at 2:49–51; *see also id.* at 5:30–44 (describing migrating the dynamic home node of a global page to the client node that has the maximum consultation count for that page).

When a static home node receives a migration request from a dynamic home node, along with the prospective candidates for a new dynamic home node, the static home node communicates with the candidate nodes, and

initiates the migration to the most suitable candidate node. *Id.* at 5:40–44, 5:53–57.

Baylor provides the following illustrative example of migrating a dynamic home node for a global page using Figures 4A–4F. Figure 4A of Baylor is reproduced below.

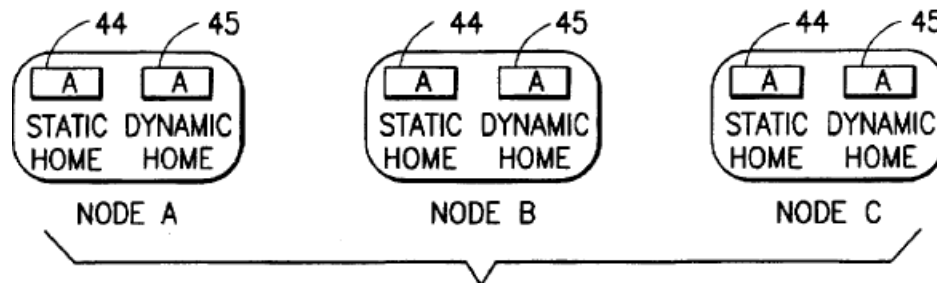


FIG.4A

Figure 4A, reproduced above, shows the Operating System in each of three nodes A through C that share a particular global page, that initially specifies node A as the static home 44 and dynamic home 45. *Id.* at 6:5–8. After some amount of network traffic, nodes A and B negotiate and make node B the “dynamic” home node for that global page. *Id.* at 6:9–12. After node B becomes the dynamic home node, node C is still unaware of the migration. *Id.* at 6:12–13. Node C requests node A for a line in the global page because it thinks node A is the dynamic home node for A. *Id.* at 6:10–15. Node A forwards the request to dynamic home B, which sends the line data and also informs node C that the dynamic home for that global page is now node B. *Id.* at 6:15–18. Node C updates its dynamic home information for that global page. *Id.* at 6:18–19. Node D goes through similar steps by noting that Node A is the static and dynamic homes. *Id.* at 6:20–22. On its first read operation, Node D is informed that Node B is the dynamic home node and updates its information. *Id.* at 6:22–24. Node B subsequently negotiates

with Nodes A and C and transfers the dynamic home node of the global page to Node C and drops the page from its memory. *Id.* at 6:25–27. Node D requests a line in the global page and sends the request to Node B as its memory indicates Node B to be the dynamic home node. *Id.* at 6:29–30. But node B has dropped the information on this page and, thus, forwards the line request to the static home node, Node A, contained in the message. *Id.* at 6:31–33. Node A forwards the request to the dynamic home node, Node C. *Id.* at 6:34–35. Node C sends the line data and the current dynamic home node information to Node D. *Id.* at 6:36–37.

2. Herlihy

Herlihy describes a distributed multiprocessor system including mobile objects and nodes, as follows. Ex. 1009, p. 19, § 1. A node can store mobile objects and perform functions related to storage of mobile objects. *Id.* Mobile objects may be a file, a process, or any other data structure, and a given mobile object is stored only on one node at a time. *Id.* The mobile object may move from one node to another. *Id.* Each node stores a directory service, which allows each node to keep track of mobile objects stored on itself, and other nodes in the system. *Id.*

Figure 2 from Herlihy is reproduced below.

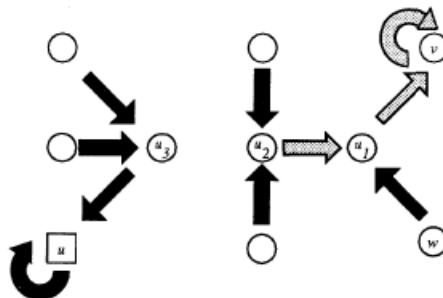


Figure 2, reproduced above, illustrates a tree system T that contains linked nodes u, v, w, u_1, u_2, u_3 . *Id.* at p. 122, § 3. A node v stores directory entry *link*

(*v*) for a particular mobile object that may or not be stored on node *v*. *Id.* at p. 121, § 3. *Link(v)* can either reference a node on which *link(v)* is stored, or a node directly linked to the node on which *link(v)* is stored. *Id.* Thus, in Figure 2, *link(v)* can reference either node *v* or node *u₁*. *Id.* When *link(v)* references node *v*, the particular mobile object is stored on node *v*. *Id.* When *link(v)* references a node other than node *v*, that does not necessarily indicate that the particular object is stored on the node referenced in *link(v)*. *Id.* Instead, when *link(v)* references a node other than node *v*, that indicates that the particular mobile object is stored in a node in the direction of the node referenced in *link(v)*. *Id.* For example, in Figure 2, when *link(v)* references node *u₁*, that indicates that the particular mobile object is stored on any one of nodes *u*, *w*, *u₁*, *u₂*, *u₃*. *Id.*

Herlihy further discloses that an embodiment of the directory service supports read-only replication. *Id.* at p. 130, § 5. When multiple read-only copies of a particular mobile object exist in a system, one copy is designated as a primary copy. *Id.* And each directory entry consists of a primary link pointing toward a node storing the primary copy, and a set of secondary links, each pointing toward a node storing a copy other than the primary copy. *Id.* When a node storing a primary copy requests exclusive access, it modifies each primary link on each directory entry of every node to reference itself. *Id.* When the node has acquired exclusive access to the primary copy, it invalidates all of the other read-only copies in the system. *Id.* And when all invalidations are complete, the node has acquired exclusive access to the object. *Id.*

3. *Whether Claims 1 and 5 Would Have Been Obvious Over the Combination of Baylor and Herlihy*

Baylor describes its invention as providing a mechanism to dynamically migrate the “home node” of a global page to a more suitable

node. Baylor splits its “home node” into a static and dynamic home nodes. As discussed further below, we find that Baylor’s static and dynamic home nodes together (i.e., “split home”) satisfies the recited “repository of last resort,” the only limitation at dispute. Furthermore, based on Herlihy’s teaching of a migration method that utilizes a link reversal system, we determine that a skilled artisan would have had reason to modify Baylor’s split-home repository of last resort by modifying the split home, the static and dynamic home nodes, into a single node, for the reasons set forth in the Petition and below. Furthermore, we find that Baylor teaches monitoring requests to the dynamic home node and determining whether to migrate its dynamic home node to a more suitable node based on network traffic. We determine that a skilled artisan would have had reason to monitor requests to Baylor’s modified repository of last resort having a single node, and decide whether to migrate that RLR for the same reasons Baylor teaches doing so with respect to its dynamic home node, i.e., to minimize network traffic.

a. Claims 1 and 5

“monitoring storage object requests to the repository of last resort”

The preamble of claim 1 states “[a] method of maintaining storage object consistency among nodes in a distributed storage architecture including a repository of last resort.” The preamble of claim 5 includes similar language. As explained in the discussion of the specific limitation below, we find that Baylor’s split home, its static and dynamic home nodes together, satisfies the claimed “repository of last resort.” Independent claim 1’s first limitation recites “monitoring storage object requests to the repository of last resort.” Independent claim 5 similarly recites “the processor operable to monitor data requests to the repository of last resort.”

Patent Owner does not specifically dispute that this limitation is taught by the combination of Baylor and Herlihy.

For the monitoring and deciding whether to migrate limitations, Petitioner refers to disclosure from Baylor. Pet. 30–31. Baylor states that client nodes, including the dynamic home node itself, maintain “[t]he number of requests sent to the dynamic home thus far,” called the “Consultation Count” of the client. Ex. 1010, 4:11–16; 5:29–30. “The consultation count . . . reflects the amount of traffic between the client and the dynamic home.” *Id.* at 5:27–29. If the number of requests from a certain client node for the global page exceeds the number of requests from other nodes, the system can migrate the dynamic home node, e.g., to the node making the most requests. *Id.* at 6:29–44.

Based on this disclosure, we determine that a skilled artisan would have had reason to monitor storage object requests, including those sent to a “repository of last resort,” specifically to monitor network traffic in order to minimize it.

“deciding whether to migrate the repository of last resort”

Claim 1 next recites “deciding whether to migrate the repository of last resort.” Claim 5 similarly recites “the processor operable to . . . decide whether to move the repository of last resort.” Patent Owner does not specifically dispute that this limitation is taught by the combination of Baylor and Herlihy.

For the deciding limitations, Petitioner refers to disclosure from Baylor. Pet. 31–32. As discussed above, Baylor disclosing using a “Consultation Count” to decide whether to migrate the dynamic home of a global page. Pet 31–32 (citing Ex. 1010, 5:28–58). As noted above, the

parties agree that Baylor's dynamic home node is part of a "repository of last resort," but does not constitute a RLR by itself. Baylor specifically states:

The [consultation] counts are maintained by all client nodes, including the dynamic home nodes. Thus, for instance, suppose there are 4 nodes, N1,N2,N3,N4, where N4 is the dynamic home. Suppose all the nodes access lines from the page and the consultation counts for them are 5, 10, 2, 3, respectively. This implies that N2 is consulting the directory more heavily than other nodes. If the home is moved to N2, it would reduce messages between nodes N2 and N4. Thus, a criterion for causing a change in the dynamic node is that the consultation count of some client is greater than the consultation count of the dynamic home itself by a predefined margin. When the dynamic home observes this situation, it communicates with the static home, supplying the node number of the client that has the maximum consultation count. The static home then arranges for the migration as described later.

Ex. 1010, 5:29–45.

Based on this disclosure, we determine that a skilled artisan would have had reason to decide whether to migrate an RLR in order to minimize network traffic, e.g., migrating the RLR to the node that has been making the most number of requests to it.

"migrating the repository of last resort"

Claim 1 recites "*migrating* the repository of last resort" (emphasis added). Claim 5 similarly recites "the processor operable to . . . *migrate* the repository of last resort" (emphasis added). Patent Owner disputes that the combination of Baylor and Herlihy teaches a "repository of last resort," which we construe as a "facility for storing a last or only remaining data replica that may not be deleted."

We agree with Petitioner that Baylor's split "home" is a "repository of last resort," as recited in the challenged claims. Pet. 22; Tr.7:4–9. Petitioner

avers that other than the static home's function of providing linkage to the backing store and knowing the location of the dynamic home so that other nodes can always find the dynamic home node, the dynamic home retains all the remaining of functions of a home, i.e., it is the default owner. Pet. 21–22 (citations omitted); Pet. Reply 13 (citing Ex. 1025 ¶ 29). According to Petitioner, “Baylor’s dynamic home node is also . . . a default owner of a copy of data that cannot be deleted until responsibility for storing the data is transferred to another default owner.” *Id.* at 22.

Petitioner’s witness, Dr. Reynolds, opines that the term “home” means “the backing store, the ultimate or **default owner** of the data block.” Ex. 1022 ¶ 37. “Home,” according to Dr. Reynolds, is the location where there is a guaranteed space for a data block, and, thus, when a data block is evicted by the “owner,” the data blocks’ “home” is able to store it. Ex. 1022 ¶¶ 26–27, 37 (according to Dr. Reynolds, RLR and “owner” can refer to either a copy of a data object or the node storing the copy). Dr. Reynolds further states that an “owner” would have been known to a skilled artisan at the time of the invention “to typically refer to a node that holds a copy of data for the purpose of writing.” *Id.* ¶ 34. According to Dr. Reynolds, “[t]he home or default owner is important as the place where a last or only copy of a data block can be stored when it is evicted from all the caches.” *Id.* at ¶ 27. In addition, according to Dr. Reynolds, the home cannot delete the data until after the data is either written up the memory hierarchy to another backing store (the memory location allocated for the data object in the next larger and slower level of the memory hierarchy) or to some other safe haven (a human owner, another system, etc.). *Id.* ¶ 39; see Pet. 22.

Based on Baylor's description of its split "home" node and Dr. Reynold's testimony on the function of a home node, we find that Baylor's split home is a "repository of last resort." Specifically, Baylor teaches that the dynamic home node stores a global page and can retain exclusive access to a line in that page when a particular client requests exclusive access by invalidating copies of the line in other clients. Ex. 1010, 3:21–51. In addition, the static home node, or other node, is responsible for bringing the page in and out of long term storage. *Id.* at 3:16–20; Tr. 1:11. Thus, we find that Baylor's split home, static and dynamic home nodes together, for a global page is a repository of last resort, i.e., a facility for storing a last or only remaining data replica that may not be deleted.² Indeed, Patent Owner's expert, Dr. Nettles agrees that Baylor's static and dynamic nodes combined constitute a "repository of last resort." Dr. Nettles opines that "it is the static node that (in conjunction with the dynamic home node) serves as the ultimate repository for storing a last or only remaining replica of that data item." Ex. 2001 ¶ 15. In support of this opinion, Dr. Nettles refers to Baylor's description of the static home node's role of bringing pages in and out of the backing store. *Id.*; *id.* ¶ 47 (citing Ex. 1010, 3:15–23). According to Dr. Nettles, "[b]oth are needed." *Id.*

With respect to the "migrating" element of the limitations of claims 1 and 5, Petitioner argues that a skilled artisan, in light of Herlihy's teachings, would have had reason to modify Baylor's split "home" to eliminate the static portion of Baylor's split home, i.e., the static home node. Pet. 25

² Given that we determine Baylor's split home is a RLR, we need not determine whether Herlihy also discloses a RLR. *See* Dec. 15.

(citations omitted). Baylor's method of migrating its dynamic home node includes the static home that always remembers the location of the dynamic home node. As Petitioner contends, migration is "expensive" if all nodes must update their records of the location of a home node whenever it moves. Pet. 23 (citing Ex. 1022 ¶¶ 50, 54). Baylor's solution to this expense is to use a "static" home. Ex. 1002 ¶¶ 54, 56. The static home node's function is to identify the location of the dynamic home node of a global page so that other nodes can always find the dynamic home node. Ex. 1010, 4:17–20, 4:58–59; Ex. 1022 ¶¶ 54, 60. Thus, a client node that wants access to a global page just needs to identify the static home node of that page (static home node never moves) without updating its directory for the migratable dynamic home node for that global page. See Ex. 1010, 4:48–5:8; 6:5–37. Petitioner contends that "[i]t would be obvious to anyone skilled in the art to substitute the single-home method of Herlihy for the split-home method of Baylor as a solution to the problem of reducing the expense of migrating [Baylor's] home." Pet. 25. According to Petitioner, Herlihy teaches a link-reversal system that provide an alternative to Baylor's split home method for minimizing migration expense. Pet. Reply 10. We determine that a skilled artisan would have had reason to modify Baylor's split home based on Herlihy because Herlihy teaches an alternative way to minimize the cost of migrating a home node.

Herlihy teaches a method of migrating a single node wherein the cost of migration is low because it uses a link-reversal system. As Petitioner argues, although all nodes in Herlihy's system contain a directory with pointers in the direction of the node with the copy of data, when migration occurs, only the nodes along the migration path need to update their

information—pointing towards the new node with the data. *Id.* at 26–27 (citing Ex. 1009, 120–21; Ex. 1022 ¶¶ 58–59). Herlihy describes a method of “moving a copy in a distributed storage architecture where only the nodes on the path along which the copy is routed need to update the records they use to find the copy.” *Id.* at 24–25 (citing Ex. 1022 ¶¶ 55, 57, 61).

Therefore, Herlihy’s link-reversal system would provide a skilled artisan a reason to modify Baylor’s split home. The rationale would have been reducing the cost of migrating the home node by eliminating the need for a separate static home node, and its function of remembering the location of the dynamic home node whenever it migrates. Pet. 23–25; Pet. Reply 10–12. The result is a single (i.e., not split) migratable RLR that is not deleted until responsibility for preserving the data is transferred to another default owner (e.g., by copying the data to more secure storage). Pet. Reply 11.

Patent Owner argues that the combination of Baylor and Herlihy does not teach a migratable RLR, and a skilled artisan would not have made the proposed modification because Baylor’s static home node serves, at least in part, as the RLR, and Baylor teaches that the static home does not migrate. PO Resp. 22–24. Thus, argues Patent Owner, in any combination of Baylor and Herlihy, a skilled artisan “would conclude that at least some component of the node responsible for storing a last or only remaining replica of data must remain static (i.e., not migratable).” *Id.* (citing *W.L. Gore & Assoc., Inc. v. Garlock, Inc.*, 721 F.2d 1540 (Fed. Cir. 1993) (recognizing that a prior art reference must be considered in its entirety, including portions that would lead away from the claimed invention); Ex. 2003, 39:25–40:5 (Dr. Reynolds agreeing that in Baylor, a data item must have a static home); Ex. 2001 ¶¶ 47–48. Patent Owner further asserts that a skilled artisan would not

have made the proposed modification because “[m]aintaining the immutability of the static home node is a critical aspect of Baylor” because that is the only way for other nodes to find the dynamic home node of the global page. PO Resp. 24. Patent Owner’s argument is misplaced.

Patent Owner misstates Petitioner’s proposed modification of Baylor’s split home. Petitioner contends that a skilled artisan would have had reason to modify Baylor’s system by utilizing Herlihy’s tree-based distributed directory/link-reversal system as an alternative to Baylor’s split home method for minimizing migration expense. Pet. Reply 10 (citations omitted). Indeed, the crux of Petitioner’s argument is that a skilled artisan would have had reason to modify Baylor’s static and dynamic home nodes into a single node in order to obviate the need for a static home node. Patent Owner further argues that a skilled artisan would not have eliminated Baylor’s static home node because Baylor’s describes one of its functions as bringing pages in and out of the backing store. Prelim. Resp. 22–23. Baylor, however, states that the function can be performed by other nodes and that the function is described as being restricted to the static home node for simplicity. Ex. 1010, 3:18–20. For example, Dr. Reynolds explains that this function could be performed by the dynamic home node. Ex. 1025 ¶ 29. Thus, Patent Owner has not sufficiently persuaded us that the combination of Baylor and Herlihy does not teach the claimed RLR or that a skilled artisan would not have had reason to make the proposed modifications.

Patent Owner also argues that a skilled artisan would not have modified Baylor based on Herlihy, because Herlihy’s method lacks fault tolerance. PO Resp. 24 (citing Ex. 2002, 1 (stating the “[Herlihy’s] Arrow protocol is not fault-tolerant, because it assumes that nodes and links never

fail”)); Ex. 2001 ¶ 49. Specifically, Patent Owner contends that Herlihy’s method assumes the presence of reliable links, which leads to a lack of fault tolerance. *Id.* at 25. Herlihy expressly states “[w]e have not addressed the issue of fault-tolerance.” Ex. 1009, p. 12, § 5. Patent Owner contends that in Herlihy, “pointers are reversed as a find() message is propagated; that is, before the object itself is relocated.” PO Resp. 25. Thus, contends Patent Owner, “if a link failure occurs, there is no assurance that the location of the object can later be determined.” *Id.* In contrast, avers Patent Owner, in the ’449 patent, RLR pointers are revised only when the actual storage object traverses the network. *Id.* (citing 1001, 7:45–53). Patent Owner also contrasts Herlihy’s method to Baylor method. Patent Owner argues that although the lack of fault tolerance in Herlihy could prevent a node from later determining the correction location of a migrated node, Baylor avoids the problem by keeping a static home node (i.e., that always knows the location of the dynamic home node of the global page). *Id.* Thus, concludes Patent Owner, a skilled artisan would not make the proposed combination and “instead would maintain at least a portion of a home node static, as taught by Baylor.” *Id.* at 25–26 (citing Ex. 2001 ¶¶ 49–50).

We are not persuaded by Patent Owner’s argument for the following reasons. Foremost, Patent Owner has not shown that the claims require fault tolerance. Indeed, Patent Owner’s expert testifies that the ’449 patent does not address fault tolerance. Ex. 1026, 77:10–12 (Dr. Nettles stating that “[t]he ’449 patent doesn’t really address the fault-tolerant issue -- the detail of fault-tolerant issues very extensively.”). In addition, as Petitioner argues (Pet. Reply 16), the relevant question is not whether the “particular combination” of Baylor and Herlihy is “the preferred, or the most desirable,

combination described in the prior art in order to provide motivation for the current invention.” *In re Fulton*, 391 F.3d 1195, 1200 (Fed. Cir. 2004).

Therefore, we are persuaded that a skilled artisan would have modified Baylor based on Herlihy’s link-reversal method for the reasons articulated by Petitioner, even if it lacks fault tolerance, as urged by Patent Owner.

Lastly, Patent Owner’s argument, with respect to Petitioner’s proposed combination of Baylor and Herlihy, summarily alleges that “the combination of Herlihy and Baylor does not suggest an RLR, which ‘provides a facility for storing a last or only remaining data replica that may not be deleted,’ as required by the challenged claims.” PO Resp. 26 (citing Ex. 2001 ¶¶ 51–52). Dr. Nettles asserts that the combined teachings of Baylor and Herlihy do not disclose an RLR that “‘may not be deleted.’” Ex. 2001 ¶ 51. According to Dr. Nettles, Baylor does not disclose that either the static or dynamic home nodes may not be deleted. *Id.*

We are not persuaded by Patent Owner’s argument because neither the claims, nor the ’449 patent specification, requires that a “repository of last resort” be non-deletable. Patent Owner did not argue for such a construction (PO Resp. 14), but applies such a construction when discussing the prior art. *See id.* at 16, 20 (citing Ex. 2001 ¶¶ 42–44). As Petitioner argues, however, the broadest reasonable interpretation of RLR does not require such non-deletable characteristic or persistence. Pet. Reply 3. Indeed, the ’449 patent specification expressly discloses that the “data replica” within the RLR is not deletable, not the RLR itself. *Id.* at 14. The ’449 patent specification provides the following:

The integrated protocol utilizes a facility called the repository of last resort (RLR). The RLR provides a facility for storing a last or only remaining copy of a data replica that may not be deleted.

In this manner the RLR ensures *the persistence of the data replica*.

Ex. 1001, 4:56–61 (emphasis added). In addition, the '449 patent specification expressly provides that the RLR may be deleted. The specification states that the RLR “may end its life by an explicit instruction to delete the data by an owner who has the authority, by migration to another physical location in another node . . . or by transfer of ownership through some input/output device to another storage system not participating in the system.” *Id.* at 4:65–5:3. Petitioner has not sufficiently persuaded us that the challenged claims do not require a RLR that is not deletable. Thus, we are persuaded that the combination of Baylor and Herlihy teaches a “repository of last resort.”

Accordingly, we determine Petitioner has proven by a preponderance of the evidence that the combination of Baylor and Herlihy teaches “migrating the repository of last resort,” and “migrate the repository of last resort,” as recited in claims 1 and 5, respectively.

b. Claim 2

Claim 2 recites “[t]he method of claim 1 further comprising the step of updating repository of last resort state bits.” Patent Owner does not make a separate argument with respect to claim 2. *See* PO Resp. 21–26. Based on Petitioner’s argument and evidence (Pet. 33–34), we are persuaded that the combination of Baylor and Herlihy teaches claim 2. As Petitioner asserts (Pet. 33–34), Herlihy teaches that “when the location of a primary copy is migrated, the directory entries containing the primary links – pointers toward the location of the primary copy – are updated to reflect the new location.” *See* Ex. 1022 ¶¶ 67–68. Dr. Reynolds opines that a skilled artisan would

have understood that the flipping of links (pointers or bits represents neighboring nodes) in the directory entries is updating pointers of state bits in a table. *Id.* ¶ 67. Dr. Reynolds further explains that given that Baylor’s modified RLR, based on Herlihy, uses Herlihy’s tree structure, links, and method of migrating a mode, “it would have been obvious to one of ordinary skill in the art that this system should also update the home/RLR links during migration to indicate the forwarding path to the home/RLR.” *Id.* ¶ 68.

4. Conclusion

We determine Petitioner has proven, by a preponderance of the evidence, that the combination of Baylor and Herlihy renders obvious claims 1, 2, and 5.

C. Borrill, Futurebus, and Baylor

As noted above, we instituted review of claims 1, 2, and 5, as having been obvious under 35 U.S.C. § 103 over the combination of Borrill, Futurebus, and Baylor. Dec. 23.

1. Borrill

Borrill discloses “an extension to snoopy cache consistency that allows it to be implemented over general interconnection networks,” referred to as the “Unified Solution.” Ex. 1008, p. 752, 754, §§ I, VI. Borrill explains that shared memory multiprocessors use local caches (“cache”). *Id.* at p. 752, § I.

In the Unified Solution, the “Repository of Last Resort” is “the storage location which is responsible for ultimate ownership of shareable data.” *Id.* at p. 754, § VI.A. Borrill recognizes that in traditional architectures, the RLR is a home memory module which contains a region of physical address space.

Id. Borrill further explains that “[i]n a cache only memory architecture (i.e., one where even main memory is designed as a cache), the RLR is the physical location in one of the caches to which a logical address is bound.” *Id.*

In Borrill, the component responsible for keeping track of routing transactions is the “Multicast Controller,” which “consists of two parts, the RLR Routing Table and the Routing Cache.” *Id.* at p. 755, § VI.C. The RLR Routing Table provides the forward path for transactions destined to the RLR. *Id.* The Routing Cache, which includes “owner bits,” provides routing information for the “owner” of the data. *Id.*; Fig. 2; Ex. 1022 ¶ 73 (Borrill “describes tracking the owner by storing the forward routing path in the owner field of the Routing Cache.”).

As shown below, Figure 3 of Borrill discloses an illustrative system.

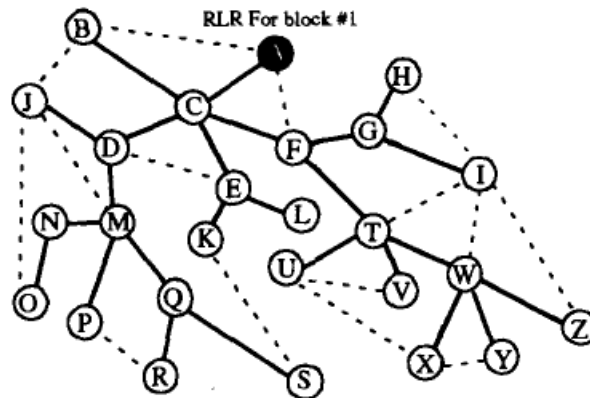


Fig. 3. Example System

Figure 3, reproduced above, is a Unified Solution with Multicast Controllers, labeled A through Z, and spanning tree links, shown as solid lines, for block #1. *Id.* at p. 755, § VI.D. Dotted lines represent links that are present in the network, but are not part of the spanning tree for block #1. *Id.* Assume cache D requests ownership of block #1 from cache A. *Id.* “A transaction is

routed along the paths DC and CA to the RLR following the paths stores in D's and C's RLR Routing Table.” *Id.* Once ownership is transferred, caches A and C mark the paths AC and CD as their owner direction in their Routing Caches, causing future transactions to be routed toward the owner rather the RLR. *Id.* If cache C receives a transaction requesting block #1 (for example, from caches B, E, or F), the request will be routed using the owner link, CD, rather than the RLR link, CA. *Id.*

2. *Futurebus*

Futurebus is a specification for a logical protocol in a microprocessor system. Ex. 1011, p.1. Futurebus describes a RLR in a hierarchical memory (or cache based) environment as “a storage location that ‘owns’ the only, or last remaining copy of shareable data.” *Id.* at p. 17, § 2.2.50. Futurebus explains that the data in a cache-only system may migrate around the system, and be owned by different caches at different times, provided no less than one copy of that data is maintained somewhere—i.e., the RLR. *Id.* Futurebus states that “[a] repository of last resort may end its life by an explicit instruction to ‘destroy’ the data by migration to a higher level in the memory (or cache hierarchy), or by transfer of ownership through some I/O device to another system, storage device, or display.” *Id.*

3. *Whether Claims 1, 2, and 5 Would Have Been Obvious Over the Combination of Borrill, Futurebus, and Baylor*

a. *Claims 1 and 5*

The preamble of claim 1 states “[a] method of maintaining storage object consistency among nodes in a distributed storage architecture including a repository of last resort.” The preamble of claim 5 includes similar language. As explained in the discussion of specific limitations

below, we find, as asserted by Petitioner, that the combination of Borrill and Futurebus teaches a “repository of last resort.”

For the limitations “monitoring storage object requests to the repository of last resort” and “deciding whether to migrate the repository of last resort, Petitioner refers to Baylor. Pet. 46–47. As discussed above, with respect to the ground based on Baylor and Herlihy, Baylor teaching monitoring requests to, and deciding whether to, migrate a dynamic home node based on a Consultation Count of requests for a global page from a client node to the dynamic home node. We find that a skilled artisan would have had reason to apply these teachings to a “repository of last resort” that results from the combined teachings of Borrill and Futurebus, to minimize network traffic, as discussed below.

“migrating the repository of last resort”

For the limitation “repository of last resort,” Petitioner refers to a combination of Borrill and Futurebus. Pet. 37, n.3. As Petitioner contends, Borrill refers to a “repository of last resort,” which is the storage location responsible for the “ultimate ownership³” of shareable data. Pet. 36 (citing Ex. 1008, p. 754). Even though Borrill refers to a “repository of last resort,”

³ As noted above, according to Dr. Reynolds, an “owner” would have been known to a skilled artisan to refer to a node that holds a copy of data for the purposes of writing. Ex. 1022 ¶ 34. Dr. Reynolds further explains the following: An owner is responsible for preserving data, and thus, the “owner copy,” if the data has been modified, is the last or only valid copy of the data in the system. *Id.* ¶ 35. An owner, however, may at any time remove, from its cache, the owned data to make room for other data. *Id.* The owner’s responsibility to the evicted data is met by transferring it back to the location that acts as the data’s backing store. *Id.*

it does not meet our above construction of “repository of last resort” as “a facility for *storing* a last or only remaining data replica that may not be deleted.” Petitioner, however, asserts that a skilled artisan would have modified Borrill’s RLR to include the storage functionality of Futurebus’s RLR. Pet. 41–42.

Futurebus also uses the term “repository of last resort.” Futurebus provides the following definition of its “repository of last resort:”

2.2.50 repository of last resort: In a hierarchical memory (or cache based) environment, *a storage location that “owns” the only, or last remaining, copy of sharable data.*

NOTE—It may be a unique source, an ultimate destination, or simply *a “safe” repository of data that may not be invalidated, unless action is taken to preserve a copy of that data at some higher level in the memory (or cache) hierarchy . . .* This data may migrate around the system, and be owned by different caches at different times, provided no less than one copy of that data is maintained somewhere. A repository of last resort may end its life by an explicit instruction to “destroy” the data by migration to a higher level in the memory (or cache hierarchy), or by transfer of ownership through some I/O device to another system, storage device, or display.

Id. at 7 (emphasis added). Petitioner concludes that a skilled artisan would have understood the RLR, resulting from modifying Borrill’s RLR to include the storage functionality of Futurebus’s RLR, to be “a default owner for storing a last or only remaining copy of data that may not be deleted (the data may be transferred to another location, but most not be lost).” Pet. 41–42. With respect to the “migrating” limitation, Petitioner argues a skilled artisan would have understood that the cost-saving rationale for migrating Baylor’s dynamic home node would apply equally to the RLR resulting from modifying Borrill’s RLR based on Futurebus, especially in a system that has

changing access patterns. *Id.* at 42–43 (citing Ex. 1022 ¶¶ 90–96). Specifically, Petitioner contends that a skilled artisan would have recognized the utility of allowing for the migration of the RLR, when, for example, a distant area of the system was known to be frequently accessing the data located in the RLR. *Id.* at 43. Petitioner further avers that because Borrill discloses a way of migrating a copy of data in a distributed storage, where only the nodes on the path along which the copy is routed need to update the records they use to find the copy (tracking the movement of the copy), a skilled artisan would understand that that this method could also be used to migrate the modified RLR. *Id.* at 42–43 (citing Ex. 1002 ¶ 91).

Based on Petitioner’s argument and the supporting evidence, including Dr. Reynold’s testimony, we determine that the combination of Borrill and Futurebus teaches a “repository of last resort.” *Id.* Futurebus expressly describes its RLR as owning, and also as a “‘safe’ repository,” of “the only, or last remaining, copy of shareable data” that may not be invalidated unless action is taken to preserve a copy of that data at some higher level in the memory/cache hierarchy.⁴ Ex. 1011, p. 7. We determine that modifying Borrill’s RLR to include this functionality would result in “a facility for storing a last or only remaining data replica that may not be deleted.”

⁴ Because we determine that the combination of Borrill and Futurebus teaches the recited “repository of last resort,” as Petitioner asserts (Pet. 37 n.3, 41–42), we need not determine whether the “repository of last resort” described by Futurebus by itself satisfies the RLR recited in the challenged claims.

We further determine that a skilled artisan would have had reason to modify Borrill's RLR to include the storage functionality of Futurebus's RLR in Borrill's RLR. In particular, as Dr. Reynolds asserts, a skilled artisan would have understood the importance of having a safe repository (default owner) to guarantee space for a data block, so that if no other node is interested in caching the data, there is space for the last or only copy of the data block at the default owner. *Id.* ¶ 85. Dr. Reynolds opines that, accordingly, to the extent it was not clear from a review of Borrill, a skilled artisan, upon reading Futurebus, would have understood that the responsibility for storage of a last or only remaining copy of data would require little or no change, indeed, would have been beneficial to Borrill's RLR. *Id.* According to Dr. Reynolds, providing this functionality to Borrill's ultimate owner would ensure that as long as the data is needed in the system, at least one node will retain a valid copy of it. *Id.* In view of this evidence, Patent Owner has not sufficiently persuaded us otherwise.

Patent Owner argues that the prior art, including Borrill and Futurebus, does not teach a "repository of last resort," but only teaches migration of "ownership" of data, i.e., writing access to data. PO Resp. 15–21. As discussed above with respect to the construction of "repository of last resort," however, the claimed RLR does not exclude "ownership."

Patent Owner contends the location of Borrill's repository of last resort is not migratable. *Id.* at 15–16. This argument is misplaced because Patent Owner argues against the references separately when the ground of unpatentability is based on a combination of prior art references. Petitioner is not arguing that Borrill alone teaches the claimed RLR, but rather asserts

that the combination of Baylor, Borrill, and Futurebus teaches a migratable RLR.

Patent Owner also argues that Borrill does not teach a RLR that “‘may not be deleted.’” PO Resp. 16. As we discussed with respect to the first ground of unpatentability based on Baylor and Herlihy, however, Patent Owner has not persuaded us that the claimed “repository of last resort” requires a non-deletable characteristic or “may not be deleted.”

With respect to Futurebus, Patent Owner argues that Futurebus teaches that “ownership” may be migrated among nodes, but not an RLR. PO Resp. 19–20. We disagree. Futurebus does describe its RLR as a storage location that “‘owns’ the only, or last remaining, copy of shareable data” and that the “data may migrate around the system, and be *owned* by different caches at different times.”

Futurebus also, however, teaches that the RLR “may be . . . an ultimate destination or simply a ‘safe’ repository of data that may not be invalidated, unless action is taken to preserve a copy of that data at some higher level in the memory (or cache) hierarchy.” Ex. 1011, p. 17, § 2.2.50 (emphasis added). Futurebus further states that its RLR “may end its life by an explicit instruction to ‘destroy’ the data by migration to a higher level in the memory (or cache hierarchy), or by transfer of ownership through some I/O device to another system, storage device, or display.” *Id.* This description of the storage function Futurebus’s RLR is consistent with the recited RLR, i.e., a facility for storing a last or only remaining data replica that may not be deleted.

Having determined that the combination of Borrill and Futurebus teaches a “repository of last resort,” we further determine that a skilled artisan would have had reason to modify this RLR to make it migratable, based on Baylor. As Dr. Reynolds states, Baylor describes the importance of migrating the home node, especially when used in applications where data access patterns are difficult to predict. Ex. 1022 ¶ 78; Ex. 1010, Abstract, 1:14–19, 2:7–31, 5:27–44. Dr. Reynolds opines that given Baylor’s disclosed motivation and policy for migrating a home node, a skilled artisan would understand the importance, and particularly the cost-savings, of having a migratable RLR in a system that may have changing access patterns. Ex. 1022 ¶ 90. Baylor describes monitoring the number of the requests its home has received for specific data and keeping track of a Consultation Count to determine whether to migrate the home. Ex. 1010, 2:19–34, 5:27–46. According to Dr. Reynolds, a skilled artisan would have further recognized the utility of using Baylor’s method to monitor requests to a RLR based on a Consultation Count, and deciding whether to migrate a RLR based on the Consultation Count when, for example, a distant node in the system is known to be frequently accessing a home node. *Id.* ¶¶ 78, 89–90. Thus, we determine that the combination of Baylor, Borrill, and Futurebus teaches the limitations “monitoring storage object requests to the repository of last resort” and “deciding whether to migrate the repository of last resort” of claim 1 and the similar limitations of claim 5.

In addition, as Petitioner notes (Pet. Reply 19), Borrill itself recognizes migration of a RLR by inclusion of a routing table with RLR Pointers. See Ex. 1025 ¶¶ 24, 26–27; Ex. 1008, pp. 754–55. As Dr. Reynolds explains, Borrill also describes migrating an owner where only the

nodes along the path, from the node requesting ownership to the node from which ownership is transferred from, update an owner pointer in their routing cache. Ex. 1022 ¶¶ 93–94. According to Dr. Reynolds, Borrill’s infrastructure functions in the same manner as the RLR locating infrastructure disclosed in the ’449 patent. Ex. 1025 ¶¶ 26, 56–67. We determine also that because Borrill already discloses infrastructure for migrating data, a skilled artisan would have found no challenge in migrating the RLR resulting from the teachings of Borrill and Futurebus. Ex. 1025 ¶ 26.

With respect to migrating the RLR and Petitioner’s reference to Baylor’s disclosure of migration of a dynamic home node, Patent Owner argues that Petitioner’s reliance on Baylor’s teachings is insufficient. Patent Owner argues that part of Baylor’s home, the static home node, is not migratable. PO Resp. 18–19. In particular, Patent Owner argues that “Petitioner seeks to selectively excise Baylor’s discussion of the dynamic home node from the overall teachings of the reference and apply it in the context of the Borrill ’94 repository of last resort (i.e., RLR-94).” PO Resp. 18 (citing Pet. 53–54). According to Patent Owner, “[t]his attempt fails, however, because even Dr. Reynolds recognizes that Baylor requires that a data object must have at least part of its home node that is (and remains) static.” *Id.* (citing Ex. 2003, 39:25–40:5). Patent Owner’s argument is misplaced. That Baylor’s static home node remains static is not the point of Petitioner’s argument. Rather, Petitioner relies on Baylor’s disclosure of a motivation and method for having a migratable home node that uses a Consultation Count. We determine that a skilled would have found Baylor’s reason for migrating the dynamic home node, i.e., minimizing network

traffic, equally relevant to migrating the RLR resulting from modifying Borrill's RLR to including the storage functionality of Futurebus's RLR.

Accordingly, we determine Petitioner has proven by a preponderance of the evidence that the combination of Baylor, Borrill and Futurebus teaches "migrating the repository of last resort," and "migrate the repository of last resort," as recited in claims 1 and 5.

b. Claim 2

Patent Owner does not make a separate argument with respect to claim 2. *See* PO Resp. 15–21. Based on Petitioner's argument and evidence (Pet. 48–49), we are persuaded that the combination of Borrill, Futurebus, and Baylor teaches the limitation of claim 2, which recites "[t]he method of claim 1 further comprising the step of updating repository of last resort state bits." As Dr. Reynolds explains, Borrill describes migrating an owner where only the nodes along the path, from the node requesting ownership to the node from which ownership is transferred from, update an owner pointer in their routing cache. Ex. 1022 ¶¶ 93–94. Moreover, as Petitioner asserts (Pet. 48–49), Borrill teaches maintaining RLR pointers in a Routing Table that provides the forward path for transactions destined to its RLR to use. Ex. 1008, p. 755. We find, that, as Dr. Reynolds explains, a skilled artisan would have had reason to update the RLR pointer to the modified Borrill RLR in the same manner that Borrill updates owner bits during the transfer of ownership in order to locate the RLR for a data item near the nodes most frequently accessing the data. Ex. 1022 ¶ 94.

4. *Conclusion*

We determine Petitioner has proven, by a preponderance of the evidence, that the combination of Baylor, Borrill, and Futurebus renders obvious claims 1, 2, and 5.

III. CONCLUSION

For the foregoing reasons, we conclude that Petitioner has shown by a preponderance of the evidence that each of claims 1, 2, and 5 of the '449 patent is unpatentable over the combination of Baylor and Herlihy and the combination of Baylor, Borrill, and Futurebus.

IV. ORDER

In consideration of the foregoing, it is hereby:

ORDERED that claims 1, 2, and 5 of the '449 patent are unpatentable; and

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

IPR2015-01923
Patent 6,631,449 B1

PETITIONER:

Bas de Blank
Donald Daybell
James Maune
ORRICK, HERRINGTON, &
SUTCLIFFE LLP
d2dptabdocket@orrick.com
m2bptabdocket@orrick.com
jjmptabdocket@orrick.com

PATENT OWNER:

Tarek Fahmi
Holly Atkinson
ASCENDA LAW GROUP, PC
tarek.fahmi@ascendalaw.com
holly.atkinson@ascendalaw.com