Trials@uspto.gov
571-272-7822

Paper 8
Date: May 16, 2013

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

ORACLE CORPORATION
Petitioner

v.

CLOUDING IP, LLC
Patent Owner

Case IPR2013-00098
Patent 6,918,014

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

KIM, *Administrative Patent Judge.*

DECISION
Institution of *Inter Partes* Review
37 C.F.R. § 42.108

I. INTRODUCTION

Background

Oracle Corporation ("Oracle") petitioned for *inter partes* review of claims 1, 3, 4, 6, 8-11, 18, 22, and 24-27 of US Patent 6,918,014 (" '014 Patent") (Ex. 1001.) pursuant to 35 U.S.C. §§ 311 et seq. The patent owner, Clouding IP, LLC ("Clouding IP"), filed a preliminary response. We have jurisdiction under 35 U.S.C. § 314.

The standard for instituting an *inter partes* review is set forth in 35 U.S.C. § 314(a) which provides as follows:

> THRESHOLD -- The Director may not authorize an inter partes review to be instituted unless the Director determines that the information presented in the petition filed under section 311 and any response filed under section 313 shows that there is a reasonable likelihood that the petitioner would prevail with respect to at least 1 of the claims challenged in the petition.

The petition is granted and we institute an *inter partes* review of all the challenged claims.

Summary of the Invention

The '014 Patent sets forth (Ex. 1001, 1:15-22):

> 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 '014 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. (Ex. 1001, 1:28-30.) In multiprocessor configurations, each processor cache can have its own copy of the same data. (Ex. 1001, 35-37.)

However, when the data is updated in the main memory, 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. (Ex. 1001, 1:37-38.) Accordingly, inconsistencies can result across processor caches for the same data. (Ex. 1001, 1:34-38.)

One prior solution is to use a snoopy cache coherence protocol, where each processor keeps track of the data on other processors. (Ex. 1001, 1:48-53.) However, performance typically is limited to an optimum number of processors connected to the shared interconnect, making it difficult to scale or extend. (Ex. 1001, 2:11-21.)

Another prior solution is 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. (Ex. 1001, 2:22-26.) However, a directory-based scheme is not scalable or extensible, 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. (Ex. 1001, 2:34-37.)

Illustrative Claim

Claims 1 and 18 are independent claims, both of which are reproduced below:

> 1. A system comprising:
> a plurality of connected nodes; and
> a storage object,
> wherein each node of the plurality of connected nodes is configured to maintain a storage object routing table;
> wherein a first node of the plurality of connected nodes is further configured to:
> identify one or more neighbor nodes of the first node;

send a message to each neighbor node of the one or more neighbor nodes, indicative of an availability of the storage object at the first node; and

wherein each neighbor node of the one or more neighbor nodes is further configured to:

create an entry for the storage object within a storage object routing table at the neighbor node in response to the message;

store an indication of a path to the first node within the entry.

18. A method comprising:

maintaining a storage object routing table at each node of a plurality of connected nodes;

identifying one or more neighbor nodes of a first node of the plurality of connected nodes;

sending a message to each neighbor node of the one or more neighbor nodes from the first node, indicative of

an availability of the storage object at the first node;

creating an entry for the storage object within a storage object routing table at each neighbor node of the one or more neighbor nodes; and

storing an indication of a path to the first node within the entry.

<u>Prior Art References Applied by Petitioner</u>

Oracle challenges the patentability of claims 1, 3-4, 6, 8-11, 18, 22, and 24-27 on the basis of the following items of prior art:

Borrill, P.L.; Onufryk, P.Z., "Extending Snoopy Cache Consistency to General Interconnection Networks," published in the Proceedings of the IEEE International Conference on Industrial Technology ("Borrill Article")	Dec. 1994	Ex. 1004
Demmer, M. and Herlihy, M., The Arrow Distributed Directory Protocol ("Herlihy")	1998	Ex. 1005
Hennessy, J. and Patterson, D., Computer Architecture: A Quantitative Approach, 2nd Ed. ("Hennessy")	1996	Ex. 1006

Raymond, K., "A Tree-Based Algorithm for Distributed Mutual Exclusion," published in ACM Transactions on Computer Systems, Vol. 7, No. 1, p. 61-77 ("Raymond") Feb. 1989 Ex. 1007

Further, Oracle relies upon declaration testimony of its witness, Dr. Paul F. Reynolds, Jr. ("Reynolds Decl.")[1]. (Ex. 1008.)

<u>The Alleged Grounds of Unpatentability</u>

Oracle alleges the following grounds of unpatentability:

a. Claims 1, 3-4, 8-10, 18, and 24-26 are unpatentable under 35 U.S.C. § 102(b) as anticipated by the Borrill Article.

b. Claims 6 and 22 are unpatentable under 35 U.S.C. § 103(a) as obvious over the Borrill Article and Herlihy.

c. Claims 11 and 27 are unpatentable under 35 U.S.C. § 103(a) as obvious over the Borrill Article and Hennessy.

d. Claims 1, 3-4, 6, 9, 18, 22, and 25 are unpatentable under 35 U.S.C. § 103(a) as obvious over Herlihy and Raymond.

e. Claims 8, 10, 24, and 26 are unpatentable under 35 U.S.C. § 103(a) as obvious over Herlihy, Raymond, and the Borrill Article.

f. Claims 11 and 27 are unpatentable under 35 U.S.C. § 103(a) as obvious over Herlihy, Raymond, and Hennessy.

[1] Dr. Reynolds holds a Ph.D. in computer science from the University of Texas at Austin and since the mid-1970s conducted research in the field of parallel and distributed systems. (Ex. 1008, ¶¶ 1-12.) From 1980 until August 2012, Dr. Reynolds was employed by the University of Virginia's School of Engineering and Applied Science as a Professor of Computer Science. (*Id*.) We conclude that Dr. Reynolds is qualified to testify as to the understanding of one skill in the art.

II. ANALYSIS

A. Findings of Fact

The following findings of facts and those in our Analysis are supported by a preponderance of the evidence.

1. Borrill Article

1. The Borrill Article relates to a cache consistency scheme in multiprocessor system. (Ex. 1004, p. 752, § II.)

2. The Borrill Articles discloses that processors are partitioned into clusters. (Ex. 1004, p. 754, § VI.)

3. The Borrill Articles discloses that a cluster may be a PC, workstation, server, or any system which uses a shared interconnect, such as a bus. (Ex. 1004, p. 754, § VI.)

4. As shown below, Figure 3 of the Borrill Article discloses a network spanning tree consisting of all clusters in a system. (Ex. 1004, p. 754-755, § VI.)

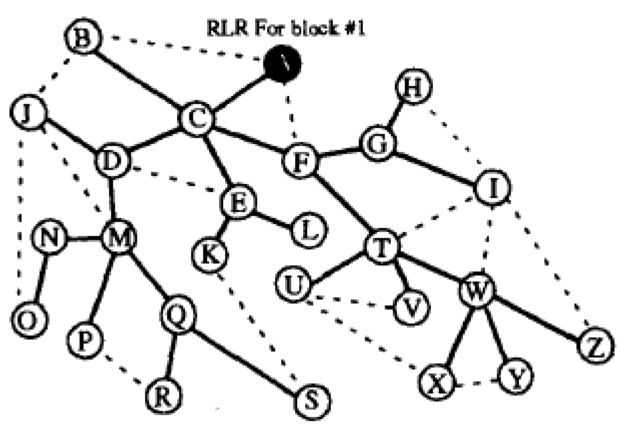

Fig. 3. Example System

5. The Borrill Article discloses that point-to-point links are used to interconnect clusters. (Ex. 1004, p. 754-755, § VI.)

6. The Borrill Article discloses that these links may be implemented as point-to-point transmission media or as virtual channels in general purpose networks. (Ex. 1004, p. 754, § VI.)

7. The Borrill Article discloses that a cluster may cache data associated with a resource. (Ex. 1004, p. 754, § VI.A.)

8. The Borrill Article discloses that a cluster may possess an "ownership" attribute for a resource if it assumes all responsibility for accuracy of the resource in the entire system. (Ex. 1004, p. 752, § II.A.)

9. The Borrill Article discloses that one of the clusters is a Repository of Last Resort ("RLR"). (Ex. 1004, p. 754, § VI.A.)

10. The Borrill Article discloses that the RLR is a cluster which is responsible for ownership of a resource. (Ex. 1004, p. 754, § VI.A.)

11. The Borrill Article discloses that a Multicast Controller serves as an interface between a cluster and its links. (Ex. 1004, p. 755, § VI.C.)

12. The Borrill Article discloses that a Multicast Controller receives requests for a resource. (Ex. 1004, p. 755, § VI.D.)

13. According to the Borrill Article, if a cluster associated with a particular Multicast Controller is an RLR for the resource, the resource is provided to the requester. (Ex. 1004, p. 754, § VI.A.)

14. According to the Borrill Article, if a cluster associated with a particular Multicast Controller does not possess the resource, the Multicast Controller forwards the request to another Multicast Controller via one of the links connected to the Multicast Controller. (Ex. 1004, p. 755, § VI.C.)

15. The Borrill Article discloses that the decision as to which Multicast Controller to send the request to, is accomplished utilizing two parts of the

Multicast Controller: an RLR Routing Table and a Routing Cache. (Ex. 1004, p. 755, § VI.C.)

16. The Borrill Article discloses that the RLR Routing Table maintains forward path information for every published resource in the system. (Ex. 1004, p. 755, § VI.C.)

17. The Borrill Article discloses that the Routing Cache examines the request for a resource and, if necessary, relays the request only in the direction of the RLR for the resource. (Ex. 1004, p. 755, § VI.C.)

18. Using Figure 3 as an example, the Borrill Article discloses that if Multicast Controller C receives a request for a resource stored at a cluster associated with Multicast Controller A, the Routing Cache of Multicast Controller C will examine the request, determine by consulting the RLR Routing Table of Multicast Controller C the forward path information of the resource, and forward the request to Multicast Controller A. (Ex. 1004, p. 755, §§ VI.C, VI.D.)

19. The Borrill Article discloses that a publication process is used by an RLR to announce an availability of a resource to other Multicast Controllers in the system. (Ex. 1004, p. 754, § VI.A.)

20. The Borrill Article discloses that during the publication process, forward path information concerning the newly available resource is established in the RLR Routing Table of each Multicast Controller. (Ex. 1004, p. 754, § VI.C.)

2. Herlihy

21. Herlihy discloses a distributed multiprocessor system including mobile objects and nodes. (Ex, 1005, p. 119, § 1.)

22. According to Herlihy, mobile objects may be a file, a process, or any other data structure. (Ex, 1005, p. 119, § 1.)

23. Herlihy discloses that a node can store mobile objects and perform functions related to storage of mobile objects. (Ex, 1005, p. 119, § 1.)

24. Herlihy discloses that a given mobile object is stored only on one node at a time. (Ex, 1005, p. 119, § 1.)

25. According to Herlihy, the mobile object may move from one node to another. (Ex, 1005, p. 119, § 1.)

26. Herlihy discloses a directory service stored on each node. (Ex, 1005, p. 119, § 1.)

27. According to Herlihy, the directory service allows each node to keep track of mobile objects stored on itself, and on other nodes in the system. (Ex, 1005, p. 119, § 1.)

28. As shown below, Figure 2 of Herlihy discloses a tree system *T* that contains linked nodes *u, v, w,* u_1, u_2, u_3. (Ex, 1005, p. 122, § 3.)

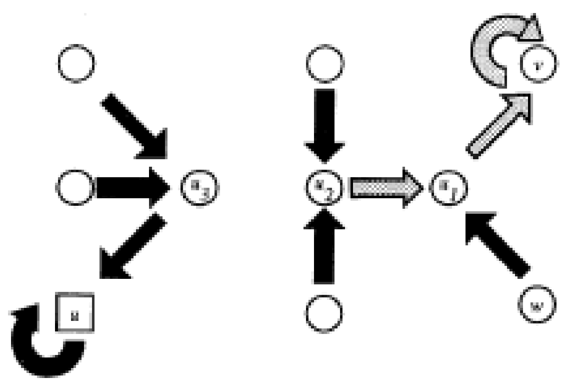

Fig. 2. Threee steps after u issues *find*(u) message

29. Herlihy discloses that node *v* stores directory entry *link(v)* for a particular mobile object that may or may not be stored on node *v*. (Ex, 1005, p. 121, § 3.)

30. Herlihy discloses that *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. (Ex, 1005, p. 121, § 3.)

31. Thus, in Figure 2 of Herlihy shown above, *link(v)* can reference either node *v* or node u_1. (Ex, 1005, p. 121, § 3.)

32. According to Herlihy, when *link(v)* references node *v*, the particular mobile object is stored on node *v*. (Ex. 1005, p. 121, § 3.)

33. According to Herlihy, when *link(v)* references a node other than node *v*, the particular mobile object is not stored on node *v*. (Ex. 1005, p. 121, § 3.)

34. Herlihy discloses that when *link(v)* references a node other than node *v*, that does not necessarily indicate that the particular mobile object is stored on the node referenced in *link (v)*. (Ex. 1005, p. 121, § 3.)

35. Instead, Herlihy discloses that 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)*. (Ex. 1005, p. 121, § 3.)

36. For example, in Figure 2 of Herlihy shown above, when *link(v)* references node u_1, that indicates that the particular mobile object is stored on any one of nodes *u*, *w*, u_1, u_2, u_3. (Ex. 1005, p. 121, § 3.)

37. Herlihy discloses a network that provides a routing service that allows node *v* to send a message to node *u*. (Ex, 1005, p. 121, § 3.)

3. Raymond

38. Raymond discloses an algorithm for distributed mutual exclusion in a computer network of *N* nodes that communicate by messages. (Ex. 1007, p. 61.)

39. Raymond discloses that the network of nodes are arranged in a tree structure. (Ex. 1007, p. 62, § 2.)

40. Raymond discloses that one node in the tree may be designated as a PRIVILEGE node. (Ex. 1007, p. 62, § 2.)

41. Raymond discloses that each node has a variable HOLDER which indicates a location of a PRIVILEGE node relative to itself. (Ex. 1007, p. 62, § 2.)

42. Raymond discloses that to initialize a network, one node is chosen as an initial PRIVILEGE node. (Ex. 1007, p. 73, § 7.)

43. Raymond discloses that the initial PRIVILEGE node sends an INITIALIZE message to each of its neighbor nodes. (Ex. 1007, p. 73, § 7.)

44. Raymond discloses that once a node has received the INITIALIZE message, it may request the PRIVILEGE from the PRIVILEGE node. (Ex. 1007, p. 73, § 7.)

45. Raymond discloses that in an exemplary system of connected nodes A and B, when node A receives an INITIALIZE message from neighboring node B, A assigns $HOLDER_A$ to B. (Ex. 1007, p. 73, § 7.)

B. Claim Construction

In assessing the merit of Oracle's petition, we have construed the claim term "a storage object" and "a storage object routing table" in light of the specification of the '014 Patent.

The Board construes a claim in an *inter partes* review using the "broadest reasonable construction in light of the specification of the patent in which it appears." 37 C.F.R. § 42.100(b); *see Office Patent Trial Practice Guides*, 77 *Fed. Reg*. 48756, 48766 (Aug. 14, 2012). Claims terms usually are given their ordinary and customary meaning as would be understood by one of ordinary skill in the art in the context of the underlying patent disclosure. *Phillips v. AWH Corp*., 415 F.3d 1303, 1313 (Fed. Cir. 2005) (en banc). Indeed, the construction that stays true to the claim language and most naturally aligns with the inventor's description is likely to be the correct construction. *Renishaw PLC v. Marposs Societa per Azioni*, 158 F.3d 1243, 1250 (Fed. Cir. 1998).

1. "Storage Object"

Independent claims 1 and 18 each recite "a storage object." Clouding IP contends a storage object is "a consistent storage entity that is persisted within a storage network (*e.g.*, through the facility of an associated RLR). (Prelim. Resp. 3-4.) We agree to an extent. The Specification discloses a storage object is "any consistent storage entity such as a file, a block or an extent which may need to be updated atomically." (Ex. 1001, 5:20-22). The Specification also discloses an RLR as an example of a storage object. (Ex. 1001, 5:62-64.) Thus, we construe a storage object as "any consistent storage entity such as a file, a block or an extent which may need to be updated atomically." This is essentially no different than Clouding IP's definition, but is more consistent with the specification language.

To further clarify the meaning of "storage object," we elaborate on additional terms. Wiley Electrical Engineering Dictionary defines "extent" as "[i]n a direct-access storage device, such as a hard disk, a contiguous block reversed for a specific data set or program." Wiley Electrical and Electronics Engineering Dictionary 270 (2004). An entity is said to be consistent if a transaction involving the entity either creates a new and valid state of data, or, if any failure occurs, returns all data to its state before the transaction was started. ISO/IEC 10026-1:1992 Section 4. Similarly, atomicity is a concept where in a transaction involving two or more discrete pieces of information, either all of the pieces are committed or none are. ISO/IEC 10026-1:1992 Section 4.

3. "Storage Object Routing Table"

Independent claims 1 and 18 each recite "a storage object routing table." Clouding IP contends a storage object routing table is "a table including path information for a storage object." (Prelim. Resp. 4-5.) We agree. The Specification discloses that "[t]he Routing Table provides a forward path, that is, a path for transactions destined to the RLR." (Ex. 1001, 5:5-7.) The Specification discloses an RLR as an example of a storage object. (Ex. 1001, 5:62-64.) "The Routing Table provides a pointer indicating a direction to the RLR." (Ex. 1001, 5:14-15.) "Each node **210** maintains a direction or forward path for all published resources in the associated Routing Table." (Ex. 1001, 5:67-6:2.) Accordingly, we construe a storage object routing table as a table including forward path, pointer, or direction information for a storage object. This is essentially no different than Clouding IP's definition, but is more consistent with the specification language.

C. 35 U.S.C. § 102(b) Grounds of Unpatentability—Claims 1, 3-4, 8-10, 18, and 24-26 as Unpatentable based on the Borrill Article

A claim is anticipated only if each and every element as set forth in the claim is described, either expressly or inherently, in a single prior art reference. *Verdegaal Bros., Inc. v. Union Oil Co. of Cal.*, 814 F.2d 628, 631 (Fed. Cir. 1987). Oracle contends that claims 1, 3-4, 8-10, 18, and 24-26 are unpatentable as anticipated by the Borrill Article. (Pet. 16-24.) We have considered Oracle's contentions and supporting evidence, and they have merit.

Taking independent claim 1 as an example, independent claim 1 recites a system comprising a plurality of connected nodes and a storage object. The Borrill Article discloses a cluster, which may be a PC, workstation, server, or any system which uses a shared interconnect, such as a bus. (Ex. 1004, p. 754, § VI.) Figure 3 of the Borrill Article discloses a network spanning tree consisting of all clusters in a system. (Ex. 1004, p. 754-755, § VI.) The Borrill Article discloses that point-to-point links are used to interconnect clusters. (Ex. 1004, p. 754-755, § VI.) The Borrill Article discloses that these links may be implemented as point-to-point transmission media or as virtual channels in general purpose networks. (Ex. 1004, p. 754, § VI.) The Borrill Article discloses that a cluster may cache data associated with a resource. (Ex. 1004, p. 754, § VI.A.). The cluster meets the definition of a storage object, which is "any consistent storage entity such as a file, a block or an extent which may need to be updated atomically."

Independent claim 1 then recites "wherein each node of the plurality of connected nodes is configured to maintain a storage object routing table; wherein a first node of the plurality of connected nodes is further configured to: identify one or more neighbor nodes of the first node." As defined herein, a storage object routing table is a table including forward path, pointer, or direction information for a storage object. The Borrill Article discloses that a Multicast Controller serves as an interface between a cluster and its links. (Ex. 1004, p. 755, § VI.C.) The Borrill Article discloses that the Multicast Controller includes an RLR Routing Table, which maintains forward path information for every published resource in the system. (Ex. 1004, p. 755, § VI.C.)

Independent claim 1 further recites "wherein a first node of the plurality of connected nodes is further configured to. . . : send a message to each neighbor node of the one or more neighbor nodes, indicative of an availability of the storage object at the first node." The Borrill Article discloses that a publication process is used by an RLR to announce an availability of a resource to other clusters in the system. (Ex. 1004, p. 754, § VI.A.)

Independent claim 1 additionally recites "wherein each neighbor node of the one or more neighbor nodes is further configured to: create an entry for the storage object within a storage object routing table at the neighbor node in response to the message" and "store an indication of a path to the first node within the entry." The Borrill Article discloses that during the publication process, forward path information concerning the newly available resource is established in the RLR Routing Table of each Multicast Controller. (Ex. 1004, p. 754, § VI.C.)

We similarly are persuaded that Oracle has demonstrated a reasonable likelihood that claims 3, 4, 8-10, 18, and 24-26 are unpatentable based upon the teachings of the Borrill Article. (Pet. 20-24.)

For the foregoing reasons, we conclude that there is a reasonable likelihood that Oracle would prevail on the grounds that claims 1, 3-4, 8-10, 18, and 24-26 are unpatentable as anticipated by the Borrill Article.

D. 35 U.S.C. § 103(a) Grounds of Unpatentability—Claims 6 and 22 are Unpatentable based on the Borrill Article and Herlihy

A patent 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 the invention was made to a person having ordinary skill in the art to which said subject matter pertains." *KSR Int'l Co. v. Teleflex Inc.*, 550 U.S. 398, 406 (2007) (citing 35 U.S.C. § 103(a)). 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 skill in the art; and (4) where in evidence, so-called secondary considerations. *Graham v. John Deere Co. of Kansas City*, 383 U.S. 1, 17-18 (1966). Oracle contends that claims 6 and 22 are unpatentable for obviousness over the Borrill Article and Herlihy. (Pet. 25-26.) We have considered Oracle's contentions and supporting evidence, and they have merit.

Both claims 6 and 22 recite "wherein the storage object is a file." As discussed above, a cluster of the Borrill Article meets the definition of a storage object. However, the Borrill Article does not disclose that the cluster is a file. Herlihy discloses mobile object are stored on nodes. (Ex, 1005, p. 119, § 1.) Herlihy's nodes are analogous to the Borrill Article's clusters. Herlihy discloses that mobile objects may be files. (Ex, 1005, p. 119, § 1.) We are persuaded by the testimony of Dr. Reynolds that a combination of the Borrill Article and Herlihy reasonably would have suggested files stored on clusters, which would meet the claim limitation. (Ex, 1008, ¶¶ 45-47.)

For the foregoing reasons, we conclude that there is a reasonable likelihood that Oracle would prevail on showing that claims 6 and 22 are unpatentable for obviousness over the Borrill Article and Herlihy.

E. 35 U.S.C. § 103(a) Grounds of Unpatentability—Claims 11 and 27 as Unpatentable based on the Borrill Article and Hennessy

Oracle contends that claims 11 and 27 are unpatentable for obviousness over the Borrill Article and Hennessy. (Pet. 27-29.) We have considered Oracle's contentions and supporting evidence, and they have merit.

Both claims 11 and 27 recite "wherein the message is sent over a TCP/IP packet-switched network." The Borrill Article discloses that a publication process is used by an RLR to announce an availability of a resource to other Multicast Controllers in the system. (Ex. 1004, p. 754, § VI.A.) Hennessy discloses that TCP/IP is the most popular communication standard between networks. (Ex. 1006, p. 609, § 7.9.) We are persuaded by the testimony of Dr. Reynolds that a combination of the Borrill Article and Hennessy reasonably would have suggested announcing an availability of a resource using TCP/IP, which would meet the claim limitation. (Ex, 1008, ¶¶ 50-51.)

For the foregoing reasons, we conclude that there is a reasonable likelihood that Oracle would prevail on showing that claims 11 and 27 are unpatentable for obviousness over the Borrill Article and Hennessy.

F. 35 U.S.C. § 103(a) Grounds of Unpatentability—Claims 1, 3-4, 6, 9, 18, 22, and 25 as Unpatentable based on Herlihy and Raymond

Oracle contends that claims 1, 3-4, 6, 9, 18, 22, and 25 are unpatentable for obviousness over Herlihy and Raymond. (Pet. 30-40.) We have considered Oracle's contentions and supporting evidence, and they have merit.

Taking independent claim 1 as an example, independent claim 1 recites a system comprising a plurality of connected nodes and a storage object. Herlihy discloses a distributed multiprocessor system including mobile objects and nodes. (Ex, 1008, p. 119, § 1.) According to Herlihy, mobile objects may be a file, a process, or any other data structure. (Ex, 1008, p. 119, § 1.) Herlihy discloses that a node can store mobile objects and perform functions related to storage of mobile objects. (Ex, 1008, p. 119, § 1.) Figure 2 of Herlihy discloses a tree system *T* that contains linked nodes u, v, w, u_1, u_2, u_3. (Ex, 1008, p. 122, § 3.) The mobile object meets the definition of a storage object, which is "any consistent storage entity such as a file, a block or an extent which may need to be updated atomically."

Independent claim 1 then recites "wherein each node of the plurality of connected nodes is configured to maintain a storage object routing table." As defined herein, a storage object routing table is a table including forward path, pointer, or direction information for a storage object. Herlihy discloses a directory service stored on each node. (Ex, 1008, p. 119, § 1.) According to Herlihy, the directory service allows each node to keep track of mobile objects stored on itself, and on other nodes in the system. (Ex, 1008, p. 119, § 1.)

Independent claim 1 also recites "wherein a first node of the plurality of connected nodes is further configured to: identify one or more neighbor nodes of the first node." Herlihy discloses that node *v* stores directory entry *link(v)* for a particular mobile object that may or may not be stored on node *v*. (Ex, 1008, p.

121, § 3.) Herlihy discloses that *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. (Ex, 1008, p. 121, § 3.)

Independent claim 1 further recites "wherein a first node of the plurality of connected nodes is further configured to…: send a message to each neighbor node of the one or more neighbor nodes, indicative of an availability of the storage object at the first node." Herlihy discloses a network that provides a routing service that allows node *v* to send a message to node *u*. (Ex, 1005, p. 121, § 3.) Raymond discloses that to initialize a network, one node is chosen as an initial PRIVILEGE node. (Ex. 1006, p. 73, § 7.) Raymond discloses that the initial PRIVILEGE node sends an INITIALIZE message to each of its neighbor nodes. (Ex. 1006, p. 73, § 7.) Raymond discloses that once a node has received the INITIALIZE message, it may request the PRIVILEGE from the PRIVILEGE node. (Ex. 1006, p. 73, § 7.) The INITIALIZE message of Raymond corresponds to the recited "message… indicative of an availability of the storage object at the first node."

Independent claim 1 additionally recites "wherein each neighbor node of the one or more neighbor nodes is further configured to: create an entry for the storage object within a storage object routing table at the neighbor node in response to the message; store an indication of a path to the first node within the entry." Raymond discloses that each node has a variable HOLDER which indicates a location of a PRIVILEGE node relative to itself. (Ex. 1006, p. 62, § 2.) Raymond discloses that in an exemplary system of connected nodes A and B, when node A receives an INITIALIZE message from neighboring node B, A assigns $HOLDER_A$ to B. (Ex. 1006, p. 73, § 7.)

We are similarly persuaded that Oracle has demonstrated a reasonable likelihood that claims 3, 4, 6, 9, 18, 22, and 25 are unpatentable based upon the teachings of Herlihy and Raymond. (Pet. 36-40.)

For the foregoing reasons, we conclude that there is a reasonable likelihood that Oracle would prevail on showing that claims 1, 3-4, 6, 9, 18, 22, and 25 are unpatentable for obviousness over Herlihy and Raymond.

G. 35 U.S.C. § 103(a) Grounds of Unpatentability—Claims 8, 10, 24, and 26 are Unpatentable based on Herlihy, Raymond, and the Borrill Article

Oracle contends that claims 8, 10, 24, and 26 are unpatentable for obviousness over Herlihy, Raymond, and the Borrill Article. (Pet. 41-43.) This alleged ground of unpatentability is redundant in light of the grounds on the basis of which we institute review for the same claims.

H. 35 U.S.C. § 103(a) Grounds of Unpatentability—Claims 11 and 27 are Unpatentable based on Herlihy, Raymond, and Hennessy

Oracle contends that claims 11 and 27 are unpatentable for obviousness over Herlihy, Raymond, and Hennessy. (Pet. 44-45.) We have considered Oracle's contentions and supporting evidence, and they have merit.

Both claims 11 and 27 recite "wherein the message is sent over a TCP/IP packet-switched network." Herlihy discloses a network that provides a routing service that allows node *v* to send a message to node *u*. (Ex, 1005, p. 121, § 3.) Hennessy discloses that TCP/IP is the most popular communication standard between networks. (Ex. 1006, p. 609, § 7.9.) We are persuaded by the testimony of Dr. Reynolds that a combination of Herlihy, Raymond, and Hennessy reasonably would have suggested announcing an availability of a resource using TCP/IP, which would meet the claim limitation. (Ex, 1008, ¶¶ 83-84.)

For the foregoing reasons, we conclude that there is a reasonable likelihood that Oracle would prevail on showing that claims 11 and 27 are unpatentable for obviousness over Herlihy, Raymond, and Hennessy.

III. ORDERS

After due consideration of the record before us, and for the foregoing reasons it is:

ORDERED that pursuant to 35 U.S.C. § 314, an *inter partes* review is hereby instituted as to claims 1, 3-4, 8-10, 18, and 24-26 of the ’014 patent on the ground of unpatentability under 35 U.S.C. § 102 as anticipated by the Borrill Article; claims 6 and 22 on the ground of unpatentability under 35 U.S.C. § 103 for obviousness over the Borrill Article and Herlihy; claims 11 and 27 on the ground of obviousness over the Borrill Article and Hennessy; claims 1, 3-4, 6, 9, 18, 22, and 25 on the ground of obviousness over Herlihy and Raymond; and claims 11 and 27 on the ground of obviousness over Herlihy, Raymond, and Hennessy.

FURTHER ORDERED that *inter partes* review is not instituted with respect to claims 8, 10, 24, and 36 on the ground of obviousness over Herlihy, Raymond, and the Borrill Article;

FURTHER ORDERED that pursuant to 35 U.S.C. § 314(c) and 37 C.F.R. § 42.4, notice is hereby given of the institution of a trial; and

FURTHER ORDERED that an initial conference call with the Board is scheduled for 2:00 PM Eastern Time on June 6, 2013. The parties are directed to the Office Patent Trial Practice Guide, 77 *Fed. Reg.* 48756, 48765-66 (Aug. 14, 2012), for guidance in preparing for the initial conference call, and should come prepared to discuss any proposed changes to the Scheduling Order entered herewith and any motions the parties anticipate filing during the trial.

For PETITIONER:

Greg Gardella
cpdocketgardella@oblon.com

Scott McKeown
cpdocketmckeown@oblon.com

For PATENT OWNER:

Tarek Fahmi
tarek.fahmi@tnfip.com

Amy Embert
amy.embert@fseip.com

Extended Graphics Array An enhanced VGA standard supporting resolutions up to 1024 x 768. Its abbreviation is **XGA**.

Extended Industry Standard Architecture Same as **EISA**.

extended memory In certain computer systems, memory beyond 1 megabyte.

extended play Playback time which is longer than ordinarily available. Also refers to storage capacity which is greater than usual. In the case of videocassette recorders, it also refers to recording time which is longer than ordinarily available. Its abbreviation is **EP**.

extended-range communication Communication by means of radio waves that are propagated well beyond line-of-sight distances. This is usually due to scattering by the ionosphere or troposphere. Also called **over-the-horizon communication, forward-scatter communication, scatter communication,** or **beyond-the-horizon communication.**

extended-range propagation Propagation of radio waves well beyond line-of-sight distances. This is usually due to scattering by the ionosphere or troposphere. Also called **over-the-horizon propagation, forward-scatter propagation, scatter propagation,** or **beyond-the-horizon propagation.**

extended-range transmission Transmission of radio waves well beyond line-of-sight distances. This is usually due to scattering by the ionosphere or troposphere. Also called **over-the-horizon transmission, forward-scatter transmission, scatter transmission,** or **beyond-the-horizon transmission.**

extended superframe A T-carrier framing format or standard with enhanced features, such as less-frequent synchronization, and real-time monitoring of the line. It encompasses 24 DS1 frames, while a **superframe** assembles 12. Its abbreviation is **ESF**. Also called **extended superframe format.**

extended superframe format Same as **extended superframe**. Its abbreviation is **ESF format**.

Extended TACACS Abbreviation of **Extended** Terminal Access Controller Access Control System. An enhanced version of **TACACS**, which provides additional support for auditing and accounting. Its abbreviation is **XTACACS**.

Extended Terminal Access Controller Access Control System Same as **Extended TACACS**.

Extended VGA Abbreviation of **Extended** Video Graphics Array. Any of various graphics standards which provide for higher resolution than VGA. SVGA resolutions range from 800 x 600 pixels, to 1600 x 1200 pixels, with support for over 16.7 million colors. Also called **SVGA**.

Extended Video Graphics Array Same as **Extended VGA**.

extensible **1.** That which can be extended or enhanced. For example, a computer system which provides expansion slots, enabling it to incorporate additional peripherals. **2.** That which can be extended or protruded. For instance, a panel which can project out, facilitating its use.

Extensible HTML Abbreviation of **Extensible** Hypertext Markup Language. A markup language that combines HTML and XML, and which features greater portability and ease of extension and enhancement. Its own abbreviation is **XHTML**.

Extensible Hypertext Markup Language Same as **Extensible HTML**.

Extensible Markup Language A specification for the format of documents and data to be used on the Web. It is a scaled-down version of SGML, and seeks to retain the comparative simplicity of HTML, yet offer greater flexibility in areas such as organization and presentation, while being fully compatible with both. Its abbreviation is **XML**.

Extensible Stylesheet Language Within **Extensible Markup Language**, a standard defining stylesheets. Its abbreviation is **XSL**.

Extensible Stylesheet Language Transformations A language utilized for converting XML documents into other XML documents with different structures. Its abbreviation is **XSLT**.

extension **1.** An enlargement in scope, or in length of time. For instance, a plug-in which adds functionality to an application. **2.** A set of characters appearing at the end of a filename, indicating the file type. For example, in the filename *notepen.exe*, the **.exe** portion is the extension, and in this case specifies an executable program. Also called **filename extension**. **3.** A telephone, modem, or similar device connected to a main line, such as a PBX.

extensometer An instrument which measures small variations in the dimensions of a solid. For instance, it may measure deformation due to stress. An example is a laser extensometer.

extent In a direct-access storage device, such as a hard disk, a contiguous block reserved for a specific data set or program.

Exterior Gateway Protocol A routing protocol used between autonomous systems on the Internet. Its abbreviation is **EGP**.

external bus A bus between a CPU and peripherals. A PCI bus is an example. An **internal bus** runs between a CPU and memory.

external cache A memory cache which utilizes a memory bank between the CPU and main memory. This contrasts with **internal cache**, which is built right into the CPU. Internal cache is faster, but smaller than external cache, while external cache is still faster than main memory. Also called **level 2 cache**, or **L2 cache**.

external capacitor A capacitor which is externally connected to an oscillator, to vary its frequency.

external circuit A circuit, or part of a circuit, which is outside of a main circuit, or any other circuit in question. For instance, that which is externally connected to a battery, including its load.

external device A device which is not part of a central system, but which is utilized with it. For instance, a peripheral, such as a mouse or printer, used with a computer system.

external drive A drive, such as a disk drive or tape drive, that is not located within the system unit of a computer, as opposed to an **internal drive**, which is.

external electric field **1.** An electric field other than that being considered. **2.** An electric field affecting a given particle or body situated in the medium surrounding said field.

external error An error caused by an **external device**.

external feedback Feedback provided by an **external circuit**.

external field An **external electric field**, or an **external magnetic field**.

external interrupt In a computer, an interrupt generated by an **external device**. This contrasts with an **internal interrupt**, which is generated by the CPU.

external magnetic field **1.** A magnetic field other than that being considered. **2.** A magnetic field affecting a given particle or body situated in the medium surrounding said field.

external memory A deprecated term for **external storage**.

external modem A modem which is self-contained, and which usually connects to a computer via a cable to a serial port. This contrasts with an **internal modem**, which is plugged into an expansion slot.

external photoelectric effect The ejection of electrons from a surface through the absorption of sufficient incident electromagnetic radiation, such as infrared, visible, or ultraviolet