UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
13/074,916	03/29/2011	Satyam B. VAGHANI	A516.CIP	1683

36378 7590 12/08/2016

VMWARE, INC.
DARRYL SMITH
3401 Hillview Ave.
PALO ALTO, CA 94304

EXAMINER
CHOI, YUK TING

ART UNIT	PAPER NUMBER
2153	

NOTIFICATION DATE	DELIVERY MODE
12/08/2016	ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

ipteam@vmware.com
ipadmin@vmware.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte SATYAM B. VAGHANI and MANJUNATH RAJASHEKHAR

Appeal 2016-001613
Application 13/074,916[1]
Technology Center 2100

Before CAROLYN D. THOMAS, JEFFREY S. SMITH, and TERRENCE W. MCMILLIN, *Administrative Patent Judges*.

MCMILLIN, *Administrative Patent Judge*.

DECISION ON APPEAL

This is a decision on appeal under 35 U.S.C. § 134(a) of the final rejection of claims 1–13 and 21–26. Final Act. 1. We have jurisdiction under 35 U.S.C. § 6(b).

We AFFIRM.

---

[1] According to Appellants, the real party in interest is VMware, Inc. (Appeal Br. 3).

REJECTIONS ON APPEAL

Claims 1–13, 22, 23, 25, and 26 stand rejected under 35 U.S.C. § 103(a) over Adam et al. (US 2008/0184249 A1, published July 31, 2008) and Chan (US 6,920,454 B1, issued July 19, 2005). Final Act. 4–9.

Claims 21 and 24 stand rejected under 35 U.S.C. § 103(a) over Adam, Chan, and Govindaraju et al. (US 6,105,050, issued Aug. 15, 2000). Final Act. 9.

THE CLAIMED INVENTION

The present invention generally relates to lock operations to gain access to system resources, and more particularly to atomic test and set operations used in performing lock operations that allow a node to acquire or release a lock to a resource of a shared file system that is stored in a data storage unit without preventing other nodes from performing input/output with the data storage unit. *See* Spec. ¶¶ 5–6, 8. Independent claim 1 is directed to a method; and independent claim 9 is directed to a non-transitory computer-readable storage medium. App. Br. 13–14.

Claim 1 recites

> 1. A method of managing accesses to resources of a shared file system that are stored in a data storage unit (DSU) using locks for the resources, wherein contents of the locks are also stored in the DSU, comprising the steps of:
>
> reading contents of a lock associated with a resource of the shared file system to obtain a current state of the lock, wherein the contents of the lock and the resource of shared file system are stored in the DSU and the contents of the lock are read from a storage location for the lock in the DSU, and wherein the shared file system is a file system that is shared among a plurality of host computers including a first host computer and a second host computer and the storage location

of the lock in the DSU is accessed by the first host computer when the first host computer is examining the current state of the lock and by the second host computer when the second host computer is examining the current state of the lock;

determining that the lock is available based on the current state;

transmitting a command to the DSU, the command including an old image of the contents, a new image of the contents, and a request to perform an atomic update to the contents of the lock comprising a first operation to confirm that the current state of the lock represented by the old image of the contents has not changed since the reading and a second operation to acquire the lock by writing the new image of the contents into the storage location for the lock in the DSU, wherein no other operation can be performed on the lock between the first operation and the second operation; and

acquiring access to the resource upon receiving confirmation of successful completion of the atomic update, wherein no exclusive reservation of the DSU is requested to update the contents of the lock and acquire access to the resource.

ANALYSIS

We have reviewed the Examiner's rejections in light of Appellants' arguments that the Examiner erred. We are not persuaded that Appellants identify reversible error. Upon consideration of the arguments presented in the Appeal Brief and Reply Brief, we agree with the Examiner that all the pending claims are unpatentable over the cited combination of references. We adopt as our own the findings and reasons set forth in the rejection from which this appeal is taken and in the Examiner's Answer. We provide the following explanation to highlight and address specific arguments and findings primarily for emphasis.

Claims 1–13, 22, 23, 25, and 26

Appellants contend Chan does not teach or suggest "*contents of a lock associated with a resource of the shared file system* to obtain a current state of the lock, wherein the *contents of the lock and the resource of shared file system are stored in the DSU* and the *contents of the lock are read from a storage location for the lock in the DSU*," as recited in claim 1. App. Br. 7; Rep. Br. 2–3 (emphasis added).

Appellants' argument against Chan separately from Adams does not persuasively rebut the combination made by the Examiner. One cannot show non-obviousness by attacking references individually, where the rejections are based on combinations of references. *In re Merck & Co., Inc.*, 800 F.2d 1091, 1097 (Fed. Cir. 1986); *In re Keller*, 642 F.2d 413, 425 (CCPA 1981).

Specifically, the Examiner finds Adams teaches reading contents of a lock associated with a resource, and resources of a shared file system that are stored in a data storage unit using locks for the resources. Final Act. 4. We agree with the Examiner.

As cited by the Examiner, Adams discloses:

> FIG. 1 illustrates a *computer system 10 including multiple processors (or CPUs) 12a, b, c*, each executing one or more programs… a single operating system 13 and *multiple applications 20, 21, and 22, all executed by processors 12a, b, c*… System 10 also includes *a known type of resource 14*, such as a data structure, a work queue, a device, an adapter, a file, a database, or a directory, *accessible in either an exclusive manner or shared manner with corresponding type of hold of the associated lock*…

Adams ¶ 21 (emphasis added).

> S*ystem 10 also includes a lock manager program 15*, lock acquisition program macro 35 and lock release program macro 45 . . . . *The lock manager 15 can support one of more locks 29 for resources 14* by allowing either an exclusive hold of a lock 29 or more concurrent, shared holds of that same lock…

Adams ¶ 22 (emphasis added).

> A *program* or program function (*such as application 20, 21, or 22*, scheduler 23 or dispatcher 24 of operating system 13 *executing on one of the processors 12a, b or c*) *invokes* the processing of FIG. 2 *when the program or program function needs an exclusive hold on a resource and calls the lock acquisition macro 35* beginning at step 200. In response, the lock acquisition macro 35 assumes that the *lock is available in state Av or AvX and atomically attempts to change a lock state of Av or AvX to X* (step 200).

Adams ¶ 33 (emphasis added).

In other words, Adams describes a system including lock contents that are managed, as well as the resources for which the lock manages access, and multiple processors; and Adams describes the processors running programs requiring resource access and reading the lock contents. As such, Adams teaches or suggests lock contents and the associated resources being stored on a system or unit, and reading the lock contents from their storage location for the purposes of accessing the associated resources.

Appellants further contend Chan does not teach or suggest "the shared file system is *a file system that is shared among a plurality of host computers* including a *first host computer and a second host computer* and the *storage location of the lock in the DSU is accessed by the first host computer* when the first host computer is examining the current state of the lock *and by the second host computer* when the second host computer is examining the current state of the lock," as recited in claim 1. App. Br. 8–9

(emphasis added). Specifically, Appellants argue that in Chan, no two nodes ever access the same lock object and neither the master RLO nor the shadow RLO of Chan describe the claimed lock. Rep. Br. 3–4. In response, the Examiner finds Chan teaches shadow RLOs for a resource are spread over multiple nodes and servers, and each shadow RLO may be used to perform the lock operation at that node related to the resource associated with the shadow RLO, and thereby that a lock contents can be read by different nodes from the lock's shadow RLOs. Ans. 5–6. We agree with the Examiner.

As cited by the Examiner, Chan discloses:

> One or more *shadow RLOs for any given resource may be spread over one or more nodes*, effectively turning the master resource locking object (MRLO) into a distributed locking object. For example, resource 261, which has a master RLO 236 on node 232, has shadow SLOs 209, 219, and 229 on nodes 202, 212, and 222, *respectively to handle lock requests for resource 261 by the corresponding data base servers on those same nodes. Each of the nodes that has a shadow RLO may be used to perform lock operations* at that node related to the resource associated with the shadow RLO. For example, node 202 can be used to perform lock operations on node 202 related to resource 261 using shadow RLO 209, even though the master RLO for resource 261 is master RLO 238 on node 232.

Chan col. 3, l. 61–col. 4, l. 8 (emphasis added). In other words, Chan describes multiple nodes reading contents of a lock from the storage location for the contents of the lock. Furthermore, Adams teaches or suggests lock contents and the associated resources being stored on a system or unit, and reading the lock contents from the lock contents storage location for the purposes of accessing the associated resources. *See* Final Act. 4, citing Adams ¶¶ 21, 22, and 33.

Again, one cannot show non-obviousness by attacking references individually, where the rejections are based on combinations of references. *In re Merck* at 1097; *In re Keller* at 425. The combination of Chan and Adams teaches or suggests lock contents and the associated resources being stored on a system or unit, and reading the lock contents from their storage locations, as described in Adams; as well as reading contents using a first host computer and a second host computer in a shared file system as described in Chan.

Appellants further contend the combination of Adams and Chan does not teach or suggest "a request to *perform an atomic update to the contents of the lock* comprising a first operation to confirm that the current state of the lock represented by the old image of the contents has not changed since the reading and a second *operation to acquire the lock by writing the new image of the contents into the storage location for the lock* in the DSU," as recited in claim 1. App. Br. 9–10; Rep. Br. 5 (emphasis added). In response, the Examiner finds Adams teaches changing or updating a previous lock state Av or Avx to a new lock state X, and thereby teaches an atomic update to the contents of the lock and writing the new image. Ans. 7–8. We agree with the Examiner.

As cited by the Examiner, Adams discloses:

> A *program* or program function (*such as application 20, 21, or 22*, scheduler 23 or dispatcher 24 of operating system 13 *executing on one of the processors 12a, b or c*) *invokes* the processing of FIG. 2 *when the program or program function needs an exclusive hold on a resource and calls the lock acquisition macro 35* beginning at step 200. In response, the lock acquisition macro 35 assumes that the *lock is available in state Av or AvX and atomically attempts to change a lock state of Av or AvX to X* (step 200).

Adams ¶ 33 (emphasis added). In other words, Adams describes analyzing the current state of a lock and atomically changing the lock state. As such, Adams teaches or suggests performing an atomic update and, by writing a new lock state, Adams teaches or suggests writing a new image for the contents of the lock.

Appellants have not provided persuasive argument or evidence that: (1) "reading contents of a lock associated with a resource of the shared file system to obtain a current state of the lock, wherein the contents of the lock and the resource of shared file system are stored in the DSU and the contents of the lock are read from a storage location for the lock in the DSU," as recited in claim 1, is not taught or otherwise suggested by Adams' lock contents and associated resources being stored on a system or unit, and reading the lock contents from their storage location for the purposes of accessing the associated resources; (2) "the storage location of the lock in the DSU is accessed by the first host computer when the first host computer is examining the current state of the lock and by the second host computer when the second host computer is examining the current state of the lock," as recited in claim 1, is not taught or otherwise suggested by Adams' lock contents and the associated resources being stored on a system or unit, and reading the lock contents from their storage locations, as described in Adams, in combination with Chan's reading contents using a first host computer and a second host computer in a shared file system; and (3) "perform an atomic update to the contents of the lock comprising a first operation to confirm that the current state of the lock represented by the old image of the contents has not changed since the reading and a second operation to acquire the lock by writing the new image of the contents into

the storage location for the lock in the DSU," as recited in claim 1, is not taught or otherwise suggested by Adams' performing an atomic update and writing a new image by atomically writing a new lock state for the contents of the lock.

Accordingly, we sustain the § 103 rejection of independent claim 1, as well as the rejection of commensurate independent claim 9 and dependent claims 2–8, 10–13, 22, 23, 25, and 26, not separately argued. *See* App. Br. 7, 12.

Claims 21 and 24

Appellants have provided no separate arguments towards patentability for claims 21 and 24. Therefore, the Examiner's § 103 rejection of claims 21 and 24 is sustained for similar reasons as noted *supra*.

DECISION

The rejection of claims 1–13 and 21–26 is affirmed.

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED