

(19) World Intellectual Property
Organization
International Bureau



(43) International Publication Date
29 December 2005 (29.12.2005)

PCT

(10) International Publication Number
WO 2005/124547 A1

(51) International Patent Classification⁷: **G06F 9/46**

(21) International Application Number:
PCT/US2005/020076

(22) International Filing Date: 6 June 2005 (06.06.2005)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/866,334 10 June 2004 (10.06.2004) US

(71) Applicant (for all designated States except US): **ORACLE INTERNATIONAL CORPORATION** [US/US]; 500 Oracle Parkway, Redwood Shores, CA 94065 (US).

(72) Inventors; and

(75) Inventors/Applicants (for US only): **CHAN, Wilson,**

Wai, Shun [CA/US]; 129 Woodbridge Circle, San Mateo, CA 94403 (US). **YUREK, Tolga** [TR/US]; 101 Irene Court #302, Belmont, CA 94002 (US).

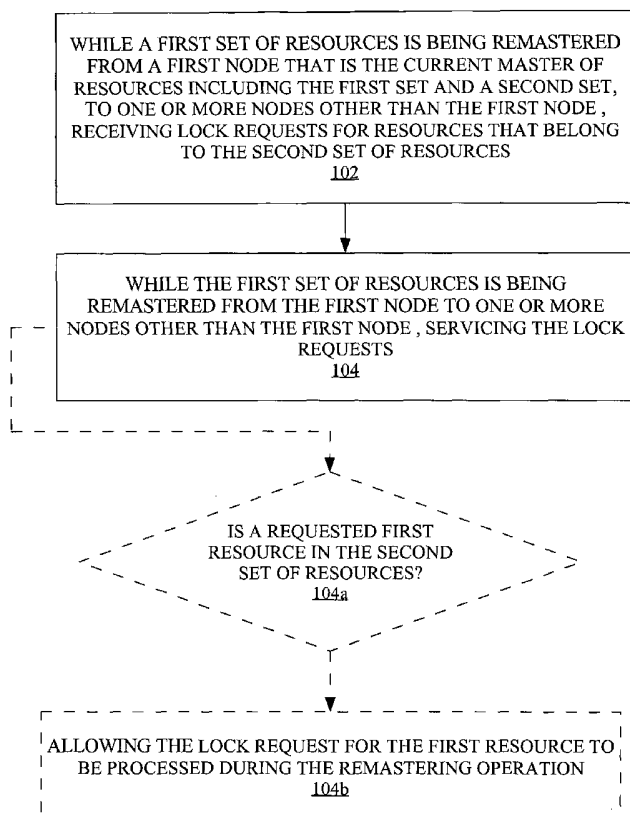
(74) Agent: **HENKHAUS, John**; Hickman Palermo Truong & Becker LLP, 2055 Gateway Place, Suite 550, San Jose, CA 95110-1089 (US).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, YU, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH,

[Continued on next page]

(54) Title: TECHNIQUES FOR ACHIEVING HIGHER AVAILABILITY OF RESOURCES DURING RECONFIGURATION OF A CLUSTER



(57) Abstract: Techniques are described for remastering shared resources among nodes in a cluster, while minimizing the number of resources that are unavailable for access ("frozen") during the remastering operation. Only the resources that are being remastered from one node to one or more other nodes are frozen, whereas the other resources that are not being remastered are not frozen and are, therefore, still accessible. For example, in the context of a database cluster, there is no longer a need to freeze the entire database during a reconfiguration remastering operation. Consequently, during the remastering operation requests for access to the resources that are not being remastered can be processed and locks granted thereon.



GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— with international search report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

TECHNIQUES FOR ACHIEVING HIGHER AVAILABILITY OF RESOURCES DURING RECONFIGURATION OF A CLUSTER

FIELD OF THE INVENTION

[0001] The present invention relates generally to clustered computing systems and, more specifically, to techniques for achieving higher availability of resources during reconfiguration of a cluster.

BACKGROUND OF THE INVENTION

[0002] A “cluster” is the result of “clustering” computing resources together in such a way that they behave like a single resource. Clustering is often used for purposes of parallel processing, load balancing and fault tolerance. One common example of a cluster is a set of computers, or “nodes”, that are configured so that they behave like a single computer. Each computer in the cluster has shared access to a set of resources. A resource is, generally, any item that can be shared by the computers in the cluster. A common example of a resource is a block of memory in which information is stored. The block of memory may be part of a node in the cluster or may be external to the cluster, such as a database block.

[0003] A cluster comprises multiple nodes that each executes an instance of a server that each facilitates access to a shared set of resources on behalf of clients of the cluster. One example of a cluster is a database cluster. A database cluster comprises multiple nodes that each executes an instance of a database server that each facilitates access to a shared database. Among other functions of database management, a database server governs and facilitates access to the particular database by processing requests by clients to access data in the database.

[0004] Typically, resources are assigned to masters, where each master coordinates the sharing of the resources assigned to it. A single node is the master of a given shared resource. A master has a global view of the state of the shared resources that it masters at any given time and acts as a coordinator for access to the shared resource. For example, a master coordinates and is aware of which node is currently granted a lock on the shared resource (and what type of lock) and which nodes are queued to obtain a lock on the shared resource. Typically, the master’s global view of the status of a shared resource is embodied in metadata associated with the resource.

[0005] Each shared resource is mapped to a master. Various mechanisms may be used to establish the resource-to-master mapping. Techniques for using hash tables to establish the resource-to-master mapping are described in detail, for example, in U.S. Patent No. 6,363,396.

The techniques described herein are not limited to any particular mechanism for establishing the resource-to-master mapping.

[0006] In order to ensure a balanced distribution of shared resource mastership among nodes in a cluster, resources should be remastered when a node membership change causes a reconfiguration in the cluster, such as when a node is added to or removed from a cluster. For example, if a node goes down and needs to be removed from the cluster, the resources that the node was mastering need to be remastered, i.e., the mastership of these resources needs to be redistributed to other nodes in the cluster. Likewise, if a node is added to the cluster, the new node should be assigned some resources to master, from the other nodes in the cluster.

Resource remastering generally entails message exchanges among the nodes regarding which resources need to be remastered as a result of the cluster reconfiguration. Once these messages are exchanged and the resources for remastering are agreed upon by the nodes, resource remastering generally entails freezing access operations (e.g., granting locks) on the particular resources being remastered, while transferring the new resource-to-master mapping and transferring the global view of the state of the resources (e.g., metadata) being remastered from the source master node to the target master node. Resource remastering associated with a cluster reconfiguration operation typically results in a better performing system.

[0007] However, with past approaches to cluster reconfiguration, while a cluster is being reconfigured, all the shared resources associated with the cluster are left in an inconsistent state until the remastering operation is completed. Lock operations on any of the resources are not allowed while the resources are in this inconsistent state, which lasts until the remastering operation is completed. This constraint results in a total freezing of access to all the shared resources associated with the cluster. In the context of a database cluster, all the resources associated with the database, i.e., the database itself, are frozen during a reconfiguration remastering operation. Hence, there is room for improvement in making resources shared among nodes in a cluster available while the cluster is being reconfigured.

BRIEF DESCRIPTION OF THE DRAWINGS

[0008] Embodiments of the present invention are illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings and in which like reference numerals refer to similar elements and in which:

[0010] FIG. 1 is a flow diagram that illustrates a process for performing a remastering operation on resources in a cluster, according to an embodiment of the invention; and

[0011] FIG. 2 is a block diagram that illustrates a computer system upon which an embodiment of the invention may be implemented.

DETAILED DESCRIPTION OF EMBODIMENTS OF THE INVENTION

[0012] In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of embodiments of the present invention. It will be apparent, however, that embodiments of the present invention may be practiced without these specific details. In other instances, well-known structures and devices are shown in block diagram form in order to avoid unnecessarily obscuring embodiments of the present invention.

FUNCTIONAL OVERVIEW OF EMBODIMENTS

[0013] Techniques are described for remastering shared resources among nodes in a cluster, while minimizing the number of resources that are unavailable for access (“frozen”) during the remastering operation. Only the resources that are being remastered from one node to one or more other nodes are frozen, whereas the other resources that are not being remastered are not frozen and are, therefore, still accessible. For example, in the context of a database cluster, there is no longer a need to freeze the entire database during a reconfiguration remastering operation. Consequently, during the remastering operation, requests for access to the resources that are not being remastered can be processed and locks granted thereon.

OPERATING ENVIRONMENT

[0014] An embodiment of the invention may be implemented in an operating environment referred to as a cluster. One type of cluster in which embodiments may be implemented is a database cluster. As previously described, a database cluster comprises multiple nodes that each executes an instance of a database server (“server”) that each facilitates access to a shared database. Each node comprises a combination of computing hardware, such as computing system 200 of FIG. 2, and a server instance. Each server instance comprises a combination of integrated software components and an allocation of computational resources (such as memory and processes) for executing the integrated software components on one or more processors of a node, where the combination of the software and computational resources are used to manage a particular common database on behalf of clients of the cluster.

[0015] The operating environment includes clients that are communicatively coupled to the cluster of database servers. The database servers are communicatively coupled to the shared database. The clients are applications or programs that cause execution of processes on the

database cluster via a network, for example. A client may execute, for non-limiting examples, on a desktop computer or on an application server.

[0016] In the context of a database cluster, the database is a repository for storing data and metadata on a persistent memory mechanism, such as a set of hard disks. Such data and metadata may be stored in the database logically, for example, according to relational database constructs, multidimensional database constructs, or a combination of relational and multidimensional database constructs. When a server accesses data from the shared database, it typically copies the data into the volatile memory of the node on which the server is executing. For example, the data is copied into the node's cache memory for manipulation purposes. Another server on another node may subsequently access the data, either from the other node's cache memory or from persistent storage, depending on the database architecture.

REMASTERING RESOURCES DURING A CLUSTER RECONFIGURATION

[0017] Partial freezing of resources during reconfiguration of a cluster allows client processes to enter a critical section of the server, in which the process can modify resource and lock structures if a particular resource of interest is not being remastered during this reconfiguration. Other attempts to enter this critical section are blocked if the resource of interest is being remastered. In a database cluster context, the average freeze time experienced by clients while trying to access the database during a cluster reconfiguration is reduced by only barring access to a portion of the database as compared to past approaches in which the entire database is frozen. Hence, a client that is working on a database transaction that only accesses portions of the database that are not being remastered during a cluster reconfiguration will be immune from the freeze during the remastering phase, which typically constitutes a majority of the reconfiguration time.

[0018] FIG. 1 is a flow diagram that illustrates a process for performing a remastering operation on resources in a cluster, according to an embodiment of the invention. A server that is executing on a node of the cluster performs the process illustrated in FIG. 1, such as a database server. The technique embodied in the process is presented in the context of a cluster reconfiguration, however, implementation of the process is not limited to reconfiguration remastering because this process can be beneficial to any remastering operation. One approach to remastering resources among nodes in a cluster is described in U.S. Patent Application No. 10/832,527 entitled "TECHNIQUES FOR MULTIPLE WINDOW RESOURCE REMASTERING AMONG NODES OF A CLUSTER", the disclosure of which is incorporated

by this reference in its entirety for all purposes as if fully set forth herein. However, embodiments of this invention are not limited to use of remastering techniques as described in the reference.

[0019] A first node in the cluster is the current master of resources that include a first set of resources and a second set of resources. While the first set of resources is undergoing a remastering operation from the first node to one or more nodes other than the first node, at block 102 lock requests are received for resources that belong to the second set of resources. A given lock request may come from a process executing on the first node, or may come from a process executing on another node in the cluster.

[0020] At block 104 the lock requests associated with the second set of resources are serviced, while the first set of resources is still undergoing the remastering operation. The process of servicing the lock request may use conventional lock management technology, which may vary from implementation to implementation. Servicing the lock requests includes allowing the requesting process to communicate with the lock management mechanism associated with the respective resource master(s), rather than blocking such an exchange. Servicing the lock requests may also include granting a lock on a resource from the second set of resources. Furthermore, servicing the lock requests may comprise (indicated by a dashed lines/blocks in FIG. 1), at block 104a, determining whether a requested resource is in the second set of resources and if it is, then allowing processing of the lock request for the requested resource during the remastering operation, block 104b.

[0021] As discussed, identification of the resources that are being remastered is provided to each node in the cluster as part of a remastering process through the exchange of messages among the nodes in the cluster. For example, upon a node terminating, another node may suggest to the other nodes a particular resource-to-master mapping for the resources being remastered from the terminating source node to one or more target nodes, based on execution of a hash function. The other nodes are supposed to agree to the new mapping, for example, based on an acknowledgement message. Similarly, upon a node initializing in the cluster, one of the nodes may suggest a particular resource-to-master mapping for the resources being remastered from the existing source nodes to the new target node, based on execution of the hash function, and the other nodes are supposed to agree to the new mapping.

[0022] Hence, all the nodes, when initially processing lock requests received from a local process for resources mastered by any node in the cluster, can determine whether or not a relevant resource is being remastered before further processing of the lock request. Further

processing of the lock request may include (1) allowing the request to be forwarded to the master node, in situations in which the node is not the master of that resource; or (2) processing the lock request according to conventional lock management techniques, in situations in which the node is the master of that resource. Alternatively, a lock request for a resource that is not mastered locally may be forwarded to the relevant master node, where the determination is made as to whether or not the resource is being remastered.

[0023] One way to determine whether or not a resource is being remastered is similar to determining the master of a resource. In one embodiment, a resource identifier associated with a given resource is hashed to a hash bucket using a hash function. From information associated with that hash bucket to which the resource is hashed, the state of the resource is determined. For example, if there is an ongoing remastering operation that involves the resource, then the new master node is identified; and if there is no ongoing remastering operation that involves the resource, then the current master node is identified. The state of each respective resource is updated in response to a remastering operation involving the resource, and every server instance in the cluster has access to the same state of each resource.

[0024] Often, with past approaches to reconfiguration of a cluster, a process desiring a lock on a shared resource is required to monitor the reconfiguration operation, for example, by communicating with a reconfiguration coordinator process, and is required to wait until a response is received that indicates that the reconfiguration operation is complete. With the techniques described herein, the caller process that is requesting a lock does not have to wait for the reconfiguration operation to complete because it is allowed to access shared resources, during the reconfiguration operation, that are not being remastered as part of the reconfiguration operation.

[0025] For example, management of requests for locks on resources may involve enforcement of "gate" logic toward the beginning of every lock operation. At this gate, a caller process that is requesting a lock on a resource is required to provide the resource identifier, which is used to determine whether a lock operation is currently allowed for that particular resource (e.g., as described above). If the resource is being remastered, then the caller may get a return message that indicates that the lock operation is not currently allowed. Subsequently, the caller is able to again request a lock on the resource. Alternatively, the caller process may simply be stalled at the gate until the lock operation is allowed. Once remastering of the resource is complete, the gate allows the caller to continue with the lock operation. By contrast,

if the resource is not being remastered when the caller arrives at the gate, then the caller is allowed to continue with the lock operation without interruption.

HARDWARE OVERVIEW

[0026] FIG. 2 is a block diagram that illustrates a computer system 200 upon which an embodiment of the invention may be implemented. Computer system 200 includes a bus 202 or other communication mechanism for communicating information, and a processor 204 coupled with bus 202 for processing information. Computer system 200 also includes a main memory 206, such as a random access memory (RAM) or other dynamic storage device, coupled to bus 202 for storing information and instructions to be executed by processor 204. Main memory 206 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 204. Computer system 200 further includes a read only memory (ROM) 208 or other static storage device coupled to bus 202 for storing static information and instructions for processor 204. A storage device 210, such as a magnetic disk, optical disk, or magneto-optical disk, is provided and coupled to bus 202 for storing information and instructions.

[0027] Computer system 200 may be coupled via bus 202 to a display 212, such as a cathode ray tube (CRT) or a liquid crystal display (LCD), for displaying information to a computer user. An input device 214, including alphanumeric and other keys, is coupled to bus 202 for communicating information and command selections to processor 204. Another type of user input device is cursor control 216, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor 204 and for controlling cursor movement on display 212. This input device typically has two degrees of freedom in two axes, a first axis (e.g., x) and a second axis (e.g., y), that allows the device to specify positions in a plane.

[0028] The invention is related to the use of computer system 200 for implementing the techniques described herein. According to one embodiment of the invention, those techniques are performed by computer system 200 in response to processor 204 executing one or more sequences of one or more instructions contained in main memory 206. Such instructions may be read into main memory 206 from another computer-readable medium, such as storage device 210. Execution of the sequences of instructions contained in main memory 206 causes processor 204 to perform the process steps described herein. In alternative embodiments, hard-wired circuitry may be used in place of or in combination with software instructions to

implement the invention. Thus, embodiments of the invention are not limited to any specific combination of hardware circuitry and software.

[0029] The term “computer-readable medium” as used herein refers to any medium that participates in providing instructions to processor 204 for execution. Such a medium may take many forms, including but not limited to, non-volatile media, volatile media, and transmission media. Non-volatile media includes, for example, optical, magnetic, or magneto-optical disks, such as storage device 210. Volatile media includes dynamic memory, such as main memory 206. Transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise bus 202. Transmission media can also take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0030] Common forms of computer-readable media include, for example, a floppy disk, a flexible disk, hard disk, magnetic tape, or any other magnetic medium, a CD-ROM, any other optical medium, punchcards, papertape, any other physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, any other memory chip or cartridge, a carrier wave as described hereinafter, or any other medium from which a computer can read.

[0031] Various forms of computer readable media may be involved in carrying one or more sequences of one or more instructions to processor 204 for execution. For example, the instructions may initially be carried on a magnetic disk of a remote computer. The remote computer can load the instructions into its dynamic memory and send the instructions over a telephone line using a modem. A modem local to computer system 200 can receive the data on the telephone line and use an infra-red transmitter to convert the data to an infra-red signal. An infra-red detector can receive the data carried in the infra-red signal and appropriate circuitry can place the data on bus 202. Bus 202 carries the data to main memory 206, from which processor 204 retrieves and executes the instructions. The instructions received by main memory 206 may optionally be stored on storage device 210 either before or after execution by processor 204.

[0032] Computer system 200 also includes a communication interface 218 coupled to bus 202. Communication interface 218 provides a two-way data communication coupling to a network link 220 that is connected to a local network 222. For example, communication interface 218 may be an integrated services digital network (ISDN) card or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, communication interface 218 may be a local area network (LAN) card to provide a data communication connection to a compatible LAN. Wireless links may also be implemented.

In any such implementation, communication interface 218 sends and receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

[0033] Network link 220 typically provides data communication through one or more networks to other data devices. For example, network link 220 may provide a connection through local network 222 to a host computer 224 or to data equipment operated by an Internet Service Provider (ISP) 226. ISP 226 in turn provides data communication services through the world wide packet data communication network now commonly referred to as the "Internet" 228. Local network 222 and Internet 228 both use electrical, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link 220 and through communication interface 218, which carry the digital data to and from computer system 200, are exemplary forms of carrier waves transporting the information.

[0034] Computer system 200 can send messages and receive data, including program code, through the network(s), network link 220 and communication interface 218. In the Internet example, a server 230 might transmit a requested code for an application program through Internet 228, ISP 226, local network 222 and communication interface 218.

[0035] The received code may be executed by processor 204 as it is received, and/or stored in storage device 210, or other non-volatile storage for later execution. In this manner, computer system 200 may obtain application code in the form of a carrier wave.

EXTENSIONS AND ALTERNATIVES

[0036] Alternative embodiments of the invention are described throughout the foregoing description, and in locations that best facilitate understanding the context of the embodiments. Furthermore, the invention has been described with reference to specific embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention. Therefore, the specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

[0037] In addition, in this description certain process steps are set forth in a particular order, and alphabetic and alphanumeric labels may be used to identify certain steps. Unless specifically stated in the description, embodiments of the invention are not necessarily limited to any particular order of carrying out such steps. In particular, the labels are used merely for

convenient identification of steps, and are not intended to specify or require a particular order of carrying out such steps.

CLAIMS

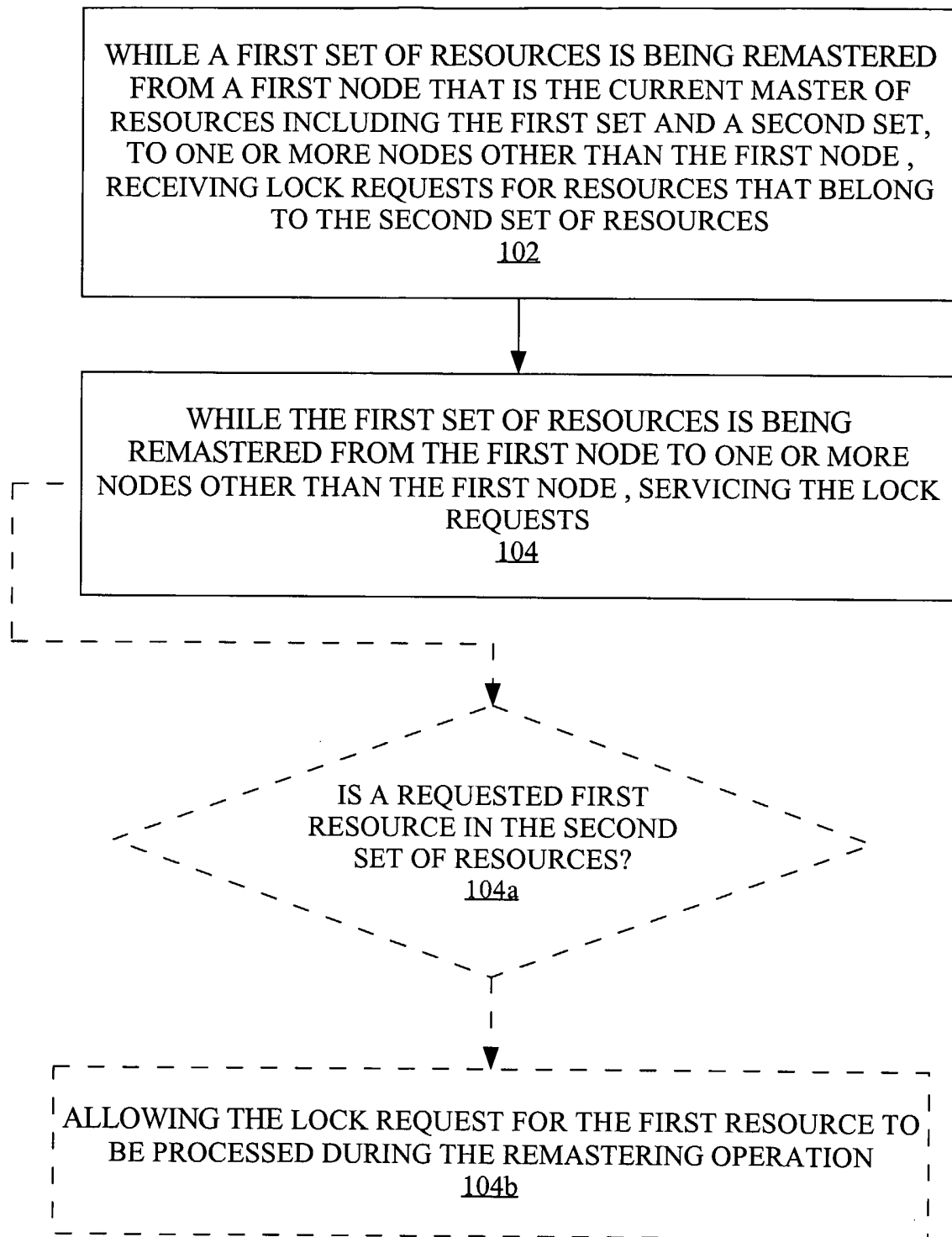
What is claimed is:

- 1 1. A method for performing a remastering operation on resources in a cluster having a
2 plurality of nodes, wherein a first node is the current master of resources including a first
3 set of resources and a second set of resources, the method comprising the computer-
4 implemented steps of:
5 while said first set of resources is undergoing a remastering operation from mastership
6 by said first node to mastership by one or more nodes other than said first node,
7 said first node
8 receiving lock requests for resources that belong to said second set of resources,
9 and
10 servicing said lock requests.
- 1 2. The method of Claim 1, wherein the step of servicing comprises granting a lock on a
2 resource from said second set of resources.
- 1 3. The method of Claim 1, wherein the step of servicing comprises
2 in response to receiving a lock request for a first resource, determining whether said first
3 resource is in said second set of resources.
- 1 4. The method of Claim 3, wherein the step of servicing comprises
2 if it is determined that said first resource is in said second set of resources, then allowing
3 said lock request for said first resource to be processed during said remastering
4 operation.
- 1 5. The method of Claim 1, wherein the step of receiving comprises receiving a lock request
2 from a process executing on said first node.
- 1 6. The method of Claim 1, wherein the step of receiving comprises receiving a lock request
2 from a process executing on a second node in said cluster other than said first node.
- 1 7. The method of Claim 6, further comprising the computer-implemented step of:
2 submitting said lock request to said first node from said second node, without waiting for
3 an indication that a cluster reconfiguration operation is complete.

- 1 8. The method of Claim 1, wherein said remastering operation is associated with
2 reconfiguring said cluster.
- 1 9. The method of Claim 1, further comprising the computer-implemented step of:
2 providing to each node of said plurality of nodes, identification of said first set of
3 resources to be remastered.
- 1 10. A computer-readable medium carrying one or more sequences of instructions which,
2 when executed by one or more processors, causes the one or more processors to perform
3 the method recited in Claim 1.
- 1 11. A computer-readable medium carrying one or more sequences of instructions which,
2 when executed by one or more processors, causes the one or more processors to perform
3 the method recited in Claim 2.
- 1 12. A computer-readable medium carrying one or more sequences of instructions which,
2 when executed by one or more processors, causes the one or more processors to perform
3 the method recited in Claim 3.
- 1 13. A computer-readable medium carrying one or more sequences of instructions which,
2 when executed by one or more processors, causes the one or more processors to perform
3 the method recited in Claim 4.
- 1 14. A computer-readable medium carrying one or more sequences of instructions which,
2 when executed by one or more processors, causes the one or more processors to perform
3 the method recited in Claim 5.
- 1 15. A computer-readable medium carrying one or more sequences of instructions which,
2 when executed by one or more processors, causes the one or more processors to perform
3 the method recited in Claim 6.
- 1 16. A computer-readable medium carrying one or more sequences of instructions which,
2 when executed by one or more processors, causes the one or more processors to perform
3 the method recited in Claim 7.

- 1 17. A computer-readable medium carrying one or more sequences of instructions which,
2 when executed by one or more processors, causes the one or more processors to perform
3 the method recited in Claim 8.
- 1 18. A computer-readable medium carrying one or more sequences of instructions which,
2 when executed by one or more processors, causes the one or more processors to perform
3 the method recited in Claim 9.

1/2

**FIG. 1**

2/2

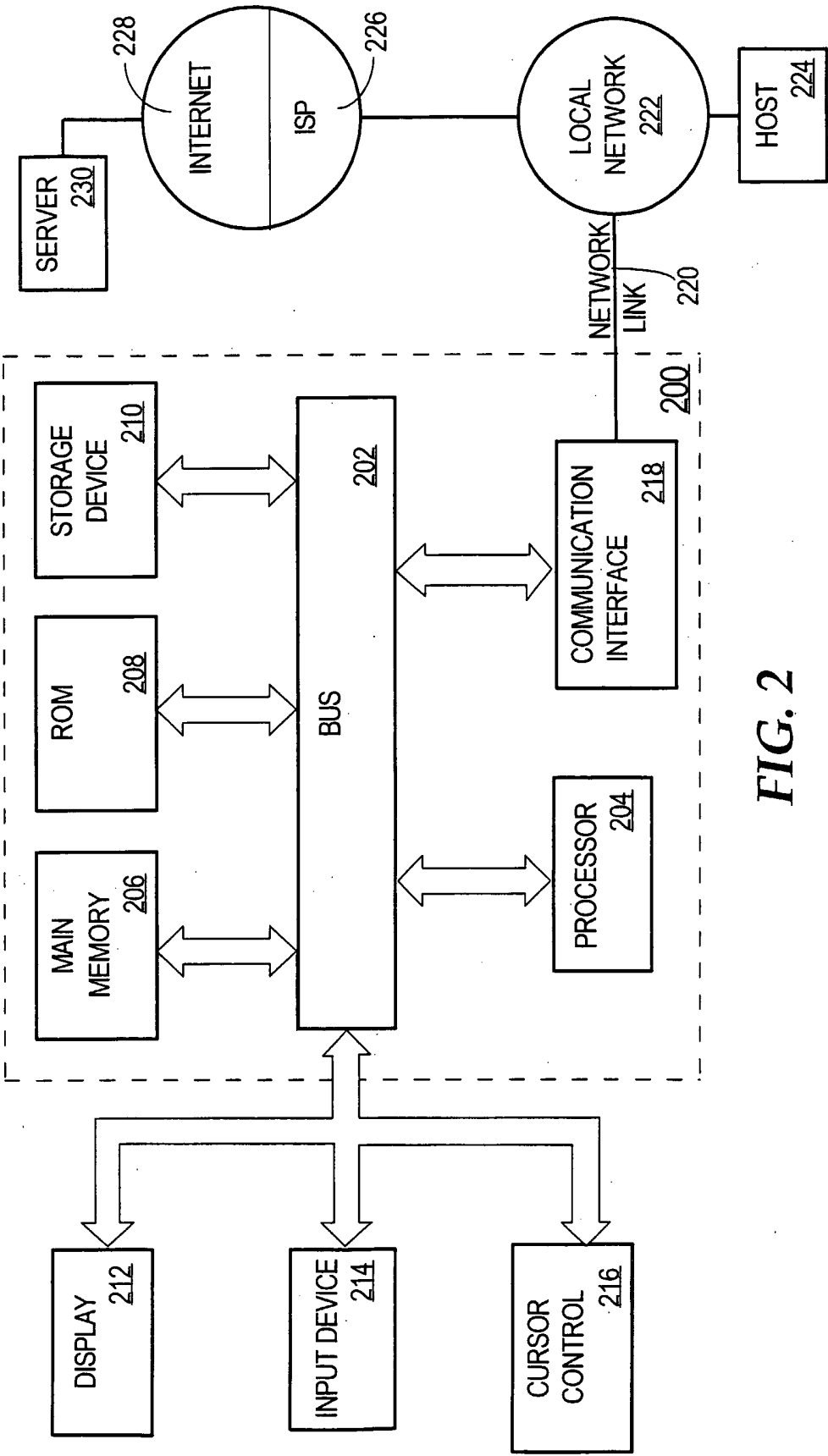


FIG. 2

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US2005/020076

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 G06F9/46

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practical, search terms used)

EPO-Internal, WPI Data, PAJ, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	<p>US 6 529 906 B1 (CHAN WILSON WAI SHUN) 4 March 2003 (2003-03-04) column 2, lines 43-59 column 3, lines 23-37 column 5, lines 41-67 column 6, lines 17-22 column 9, lines 51-67 column 10, lines 1-5 column 11, lines 5-10 column 12, lines 10-54 column 16, lines 5-9 column 19, lines 14-59 column 20, lines 14-19 column 22, line 57 - column 23, line 4 column 24, lines 31-67 figures 2-10C</p> <p>----- -/--</p>	1-18

☒ Further documents are listed in the continuation of box C.

☒ Patent family members are listed in annex.

* Special categories of cited documents:

- *A* document defining the general state of the art which is not considered to be of particular relevance
- *E* earlier document but published on or after the international filing date
- *L* document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
- *O* document referring to an oral disclosure, use, exhibition or other means
- *P* document published prior to the international filing date but later than the priority date claimed

- *T* later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
- *X* document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
- *Y* document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.
- * & * document member of the same patent family

Date of the actual completion of the international search

29 September 2005

Date of mailing of the international search report

10/10/2005

Name and mailing address of the ISA

European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HV Rijswijk
Tel. (+31-70) 340-2040, Tx. 31 651 epo nl,
Fax: (+31-70) 340-3016

Authorized officer

No11, J

INTERNATIONAL SEARCH REPORT

International Application No
PCT/US2005/020076

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category *	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2003/200398 A1 (HARRIS KENNETH M) 23 October 2003 (2003-10-23) paragraphs '0019! - '0046! paragraphs '0086! - '0133! figures 1-26 -----	1-18
A	US 6 249 879 B1 (WALKER BRUCE J ET AL) 19 June 2001 (2001-06-19) column 2, line 40 - column 3, line 47 column 7, line 30 - column 9, line 11 column 13, lines 35-64 figures 1-8 -----	1-18
P,A	US 2004/220933 A1 (WALKER MICHAEL LEO) 4 November 2004 (2004-11-04) paragraphs '0013! - '0016! paragraphs '0109! - '0130! figures 1-15 -----	1-18

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/US2005/020076

Patent document cited in search report		Publication date	Patent family member(s)	Publication date
US 6529906	B1	04-03-2003	NONE	
US 2003200398	A1	23-10-2003	NONE	
US 6249879	B1	19-06-2001	NONE	
US 2004220933	A1	04-11-2004	WO 2004097632 A2	11-11-2004