(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
22 February 2007 (22.02.2007)

PCT

(10) International Publication Number
WO 2007/021709 A1

(51) International Patent Classification:
G06F 11/20 (2006.01)

(21) International Application Number:
PCT/US2006/030961

(22) International Filing Date: 8 August 2006 (08.08.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
11/205,529 16 August 2005 (16.08.2005) US

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

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **CHAN, Wilson, Wai, Shun** [CA/US]; 129 Woodbridge Circle, San Mateo, California 94403 (US). **PRUSCINO, Angelo** [IT/US]; 436 Distel Drive, Los Altos, California 94022 (US). **ROESCH, Stefan** [AT/US]; 2205 Bridgepointe Parkway, Apt. #0312, San Mateo, California 94404 (US). **ZOLL, Michael** [DE/US]; 346 Meridian Drive, Rewood City, California 94065 (US). **YUREK, Tolga** [TR/US]; 852 Peary Lane, Foster City, California 94404 (US).

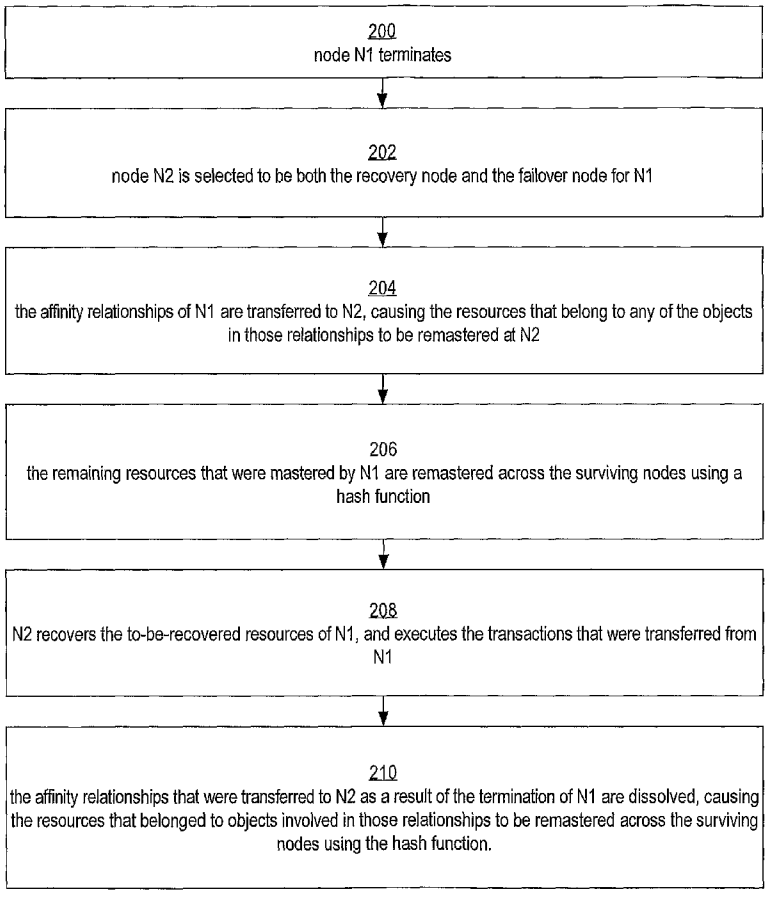
(74) Agent: **HICKMAN, Brian, D.**; HICKMAN PALERMO TRUONG & BECKER LLP, 2055 Gateway Place, Suite 550, San Jose, CA 95110, (US).

(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, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM,

[Continued on next page]

(54) Title: AFFINITY-BASED RECOVERY/FAILOVER IN A CLUSTER ENVIRONMENT



(57) Abstract: Techniques are provided for responding to the termination of a node by selecting another node, and assigning to the selected node the affinity relationships that existed between the terminated node and one or more objects. The resources that belong to the objects involved in the affinity relationships are remastered to the selected node. The selected node then performs recovery of the resources that had been opened by the terminated node and/or serves as a failover node to execute the transactions that had been executing on the terminated node.

WO 2007/021709 A1

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, LV, 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*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments*

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.

AFFINITY-BASED RECOVERY/FAILOVER IN A CLUSTER ENVIRONMENT

FIELD OF THE INVENTION

[0001] The present invention relates to database systems and, more specifically, to techniques for performing recover/failover operations in database systems where objects are mastered based on affinity to nodes.

BACKGROUND

[0002] Within the context of computer systems, many types of resources can be shared among processes. However, many resources, though sharable, may not be accessed in certain ways by more than one process at any given time. For example, resources such as data blocks of a storage medium or tables stored on a storage medium may be concurrently accessed in some ways (e.g. read) by multiple processes, but accessed in other ways (e.g. written to) by only one process at a time. Consequently, mechanisms have been developed which control access to resources.

[0003] One such mechanism is referred to as a lock. A lock is a data structure that indicates that a particular process has been granted certain rights with respect to a resource. There are many types of locks. Some types of locks may be shared on the same resource by many processes, while other types of locks prevent any other locks from being granted on the same resource.

[0004] The entity responsible for granting locks on resources is referred to as a lock manager. In a single node database system, a lock manager will typically consist of one or more processes on the node. In a multiple-node system, such as a multi-processing machine or a local area network, a lock manager may include processes distributed over numerous nodes. A lock manager that includes components that reside on two or more nodes is referred to as a distributed lock manager.

[0005] Figure 1 is a block diagram of a multiple-node computer system 100. Each node has stored therein a database server and a portion of a distributed lock management system 132. Specifically, the illustrated system includes three nodes 102, 112 and 122 on which reside database servers 104, 114 and 124, respectively, and lock manager units 106, 116 and 126, respectively. Database servers 104, 114 and 124 have access to the same database 120. The

database 120 resides on a disk 118 that contains multiple blocks of data. Disk 118 generally represents one or more persistent storage devices which may be on any number of machines, including but not limited to the machines that contain nodes 102, 112 and 122.

[0006] A communication mechanism allows processes on nodes 102, 112, and 122 to communicate with each other and with the disks that contain portions of database 120. The specific communication mechanism between the nodes and disk 118 will vary based on the nature of system 100. For example, if the nodes 102, 112 and 122 correspond to workstations on a network, the communication mechanism will be different than if the nodes 102, 112 and 122 correspond to clusters of processors and memory within a multi-processing machine.

[0007] Before any of database servers 104, 114 and 124 can access a resource shared with the other database servers, it must obtain the appropriate lock on the resource from the distributed lock management system 132. Such a resource may be, for example, one or more blocks of disk 118 on which data from database 120 is stored.

[0008] Lock management system 132 stores data structures that indicate the locks held by database servers 104, 114 and 124 on the resources shared by the database servers. If one database server requests a lock on a resource while another database server has a lock on the resource, then the distributed lock management system 132 must determine whether the requested lock is consistent with the granted lock. If the requested lock is not consistent with the granted lock, then the requester must wait until the database server holding the granted lock releases the granted lock.

[0009] According to one approach, lock management system 132 maintains one master resource object for every resource managed by lock management system 132, and includes one lock manager unit for each node that contains a database server. The master resource object for a particular resource stores, among other things, an indication of all locks that have been granted on or requested for the particular resource. The master resource object for each resource resides within only one of the lock manager units 106, 116 and 126.

[0010] The node on which a lock manager unit resides is referred to as the "master node" (or simply "master") of the resources whose master resource objects are managed by that lock manager unit. Thus, if the master resource object for a resource R1 is managed by lock manager unit 106, then node 102 is the master of resource R1.

[0011] In typical systems, a hash function is employed to select the particular node that acts as the master node for a given resource. For example, system 100 includes three nodes, and therefore may employ a hash function that produces three values: 0, 1 and 2. Each value is

associated with one of the three nodes. The node that will serve as the master for a particular resource in system 100 is determined by applying the hash function to the name of the resource. All resources that have names that hash to 0 are mastered on node 102. All resources that have names that hash to 1 are mastered on node 112. All resources that have names that hash to 2 are mastered on node 122.

[0012] When a process on a node wishes to access a resource, a hash function is applied to the name of the resource to determine the master of the resource, and a lock request is sent to the master node for that resource. The lock manager on the master node for the resource controls the allocation and deallocation of locks for the associated resource.

[0013] While the hashing technique described above tends to distribute the resource mastering responsibility evenly among existing nodes, it has some significant drawbacks. For example, it is sometimes desirable to be able to select the exact node that will function as master node to a lock resource. For example, consider the situation when a particular lock resource is to be accessed exclusively by processes residing on node 102. In this situation, it would be inefficient to have the lock resource and the request queue for that resource located on any node in the network other than node 102. However, the relatively random distribution of lock resource management responsibilities that results from the hash function assignment technique makes it unlikely that resources will be mastered at the most efficient locations.

[0014] To address the inefficiency associated with the randomness of assigning masters based on a hash function, techniques have been developed for establishing resource-to-master-node assignments based on the affinity between (1) nodes and (2) the objects to which the resources belong. In this context, an "object" may be any entity that includes resources that are protected by locks. The types of objects to which the techniques described herein may be applied may vary based on the type of system in which the techniques are used. For example, within a relational database system, "objects" could include tables, table partitions, segments, extents, indexes, Large Objects (LOBs), etc. Within a file system, "objects" could include files, sets of file system metadata, etc. Within a storage system, "objects" could include storage devices, disk sectors, etc.

[0015] The "affinity" between a node and an object refers to the degree of efficiency achieved by assigning the node to be the master of the resources that belong to the object. For example, a particular node that accesses a table much more frequently than any other node has a high degree of affinity to the table. Relative to that table, the degree of affinity for that particular node is high because, if that node is assigned to be the master of the resources within the table, a

high number of inter-node lock-related communications would be avoided. On the other hand, a node that accesses a table much less frequently than other nodes has a low degree of affinity to the table, because assigning that node to be the master of the table would avoid few inter-node lock-related communications.

[0016] The Related Applications describe various techniques related to mastering resources based on the affinity between nodes and the objects to which the resources belong. In general, once an affinity relationship has been established between an object and a node, the resources for the object cease to be randomly mastered across the nodes in the system. Instead, the node becomes master for all of the resources that belong to the object. On the other hand, when an affinity relationship is dissolved, the resources of the object are no longer mastered by the node with whom they had the affinity relationship. Instead, the resources are remastered across the nodes in the system.

[0017] The approaches described in this section are approaches that could be pursued, but not necessarily approaches that have been previously conceived or pursued. Therefore, unless otherwise indicated, it should not be assumed that any of the approaches described in this section qualify as prior art merely by virtue of their inclusion in this section.

BRIEF DESCRIPTION OF THE DRAWINGS

[0018] The present invention is 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:

[0019] Figure 1 is a block diagram of a computer system having a distributed lock manager;

[0020] Figure 2 is a block diagram that illustrates steps for responding to termination of a node, according to an embodiment of the invention; and

[0021] Figure 3 is a block diagram of a computer system upon which embodiments of the invention may be implemented.

DETAILED DESCRIPTION

[0022] In the following description, for the purposes of explanation, numerous specific details are set forth in order to provide a thorough understanding of the present invention. It will be apparent, however, that 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 the present invention.

NODE TERMINATION

[0023] A node may terminate for any number of reasons. For example, termination of a node may result from a hardware or software error. In addition, a node may be intentionally taken off line to be repaired or moved. When a node terminates for any reason, certain tasks have to be performed to ensure that the cluster to which the node belonged continues to operate correctly and efficiently. Typically, those tasks include (1) remastering resources that were mastered at the terminated node, (2) migrating transactions that were executing on the terminated node, and (3) recovering the resources that had been opened by the terminated node. Each of these tasks shall now be described in greater detail.

REMASTERING RESOURCES THAT WERE MASTERED AT A TERMINATED NODE

[0024] When a node fails, the resources that were mastered by that node have to be remastered by the remaining nodes. The resources that were mastered by a terminated node are referred to herein as the "to-be-remastered resources".

[0025] In systems that use affinity-based assignment mechanisms, the failure of a node in an affinity relationship may be an event that leads to the dissolution of the affinity relationship. Specifically, when a node in a cluster fails, any affinity relationships involving the node may be dissolved. After the affinity relationships of the terminated node are dissolved, none of the to-be-remastered resources will belong to an object that is in an affinity relationship. Since none of the to-be-remastered resources belong to objects that have affinity relationships with any of the remaining nodes, all of the to-be-remastered resources are randomly remastered across the remaining node using a hash function.

MIGRATING TRANSACTIONS OF A TERMINATED NODE

[0026] In addition to remastering the to-be-remastered resources of a terminated node, the failure of the node may also result in a failover operation, where transactions that were being handled by the terminated node at the time of the failure are transferred to one or more of the remaining nodes. Automatic failover techniques are described, for example, in U.S. Patent No. 6,490,610, entitled "Automatic Failover for Clients Accessing a Resource Through a Server", issued to Rizvi et al. on May 30, 1997, the contents of which are incorporated herein by reference. Planned failover techniques are described in U. S. Patent No. 6,199,110, entitled "Planned Session Termination for Clients Accessing a Resource Through a Server", issued to Rizvi et al. on March 6, 2001, the contents of which are incorporated herein by reference.

[0027] If the transactions that were executing on the terminated node are automatically migrated to a failover node, the failover node will have to obtain locks on the resources being used by those transactions. Obtaining those locks may also result in a significant amount of inter-node traffic.

RECOVERING RESOURCES HELD BY A TERMINATED NODE

[0028] When a node terminates unexpectedly, the resources that had been opened by the node may have been left in an inconsistent state. To return the resources to a consistent state, certain recovery operations need to be performed on the resources. Techniques for performing recovery operations on resources are described in U.S. Patent No. 6,182,241, entitled "Method and Apparatus for Improved Transaction Recovery", issued to Ngai et al., on January 20, 2001, the contents of which are incorporated herein by reference.

[0029] Typically, one of the remaining nodes is assigned to perform recovery operations on the resources that the terminated node had open at the time of the failure (the "to-be-recovered resources"). To perform recovery, the designated "recovery node" may have to obtain locks on the to-be-recovered resources. Obtaining those locks may result in a significant amount of inter-node traffic.

AFFINITY-BASED REMASTERY DURING RECOVERY

[0030] As mentioned above, upon the termination of a node, current systems randomly remaster the resources that were mastered by the terminated node. The random remastery of those resources makes sense in systems where the resources that were mastered by the terminated node had been randomly assigned to the terminated node. However, in systems that

use affinity-based assignment mechanisms, the random remastery of the to-be-remastered resources may lead to inefficiencies.

[0031] Specifically, the to-be-remastered resources may include many resources that belong to objects that had an affinity relationship with the terminated node. Such objects are referred to herein as "affinity objects". The resources that belong to affinity objects are referred to herein as "affinity resources".

[0032] Due to the affinity between the affinity objects and the terminated node, the terminated node may have had many open locks on affinity resources at the time the terminated node terminated. Consequently, many of the affinity resources may also be to-be-recovered resources. Because the recovery node will have to obtain locks on the to-be-recovered resources, and affinity resources are likely to be to-be-recovered resources, efficiency may be achieved by remastering the affinity resources at the recovery node.

[0033] According to one embodiment, affinity-based remastering is performed by transferring the affinity relationships of the terminated node to the recovery node. The transfer of the affinity relationships to the recovery node causes the affinity resources to be mastered at the recovery node. Resources that had been mastered at the terminated node that did not belong to objects involved in an affinity relationship could be remastered across all of the surviving nodes using a hash function.

[0034] After the affinity relationships of the terminated node have been transferred to the recovery node, the recovery of affinity resources will not require inter-node communication. If a high percentage of the to-be-recovered resources are affinity resources, then the amount of inter-node communication generated by the recovery operation may be dramatically reduced.

AFFINITY-BASED REMASTERY DURING FAILOVER

[0035] Due to the affinity between the affinity objects and the terminated node, the transactions that were being executed by the terminated node may be transactions that frequently access affinity resources. Consequently, there is a high likelihood that a failover node may heavily access the affinity resources after the transactions of the terminated node are transferred to the failover node. Because the failover node will have to obtain locks on the resources accessed by the transferred transactions, and the transferred transactions are likely to access affinity resources, efficiency may be achieved by remastering the affinity resources at the failover node.

[0036] According to one embodiment, affinity-based remastering is performed by transferring the affinity relationships of the terminated node to the failover node. The transfer of the affinity relationships to the failover node causes the affinity resources to be mastered at the failover node. Resources that had been mastered at the terminated node that did not belong to objects involved in an affinity relationship could be remastered across all of the surviving nodes using a hash function.

[0037] After the affinity relationships of the terminated node have been transferred to the failover node, operations in which the failover node accesses an affinity resource will not require inter-node communication. If a high percentage of the accesses performed by the transferred transactions involve affinity resources, then the amount of inter-node communication generated by the transferred transactions may be dramatically reduced.

COMBINED FAILOVER-RECOVERY

[0038] As explained above, inter-node traffic may be reduced by either (1) remastering the affinity resources at the recovery node, or (2) remastering the affinity resources at the failover node. According to one embodiment, the affinity resources may be remastered at both the recovery node and the failover node by assigning the same node to be both the recovery node and the failover node of the terminated node.

[0039] In such an embodiment, termination of a node causes a surviving node to be selected to be both the recovery node and the failover node for the terminated instance. This selection may be based, for example, on characteristics of the surviving nodes, such as the number of CPUs, the amount of memory available, etc. Once a combined recovery/failover node is selected, the affinity relationships of the terminated node are automatically transferred to the combined recovery/failover node.

[0040] Because of recovery/failover node inherits the affinity relationships, all of the affinity resources are remastered at the recovery/failover node. Because the recovery/failover node is the master of the affinity resources, recovery operations performed on affinity resources do not cause inter-node lock-related traffic. Similarly, operations performed by the transferred transactions on any affinity resources do not cause inter-node lock-related traffic.

DISSOLUTION OF AFFINITY RELATIONSHIPS

[0041] As explained above, the affinity relationships of a terminated node are not automatically dissolved upon the termination of the node. Instead, those relationships are

transferred to a recovery node, a failover node, or a combined recovery/failover node. Once transferred, those affinity relationships continue until dissolved. The conditions that result in dissolution may vary from implementation to implementation.

[0042] For example, in an embodiment in which the affinity relationships are transferred to a recovery node, the affinity relationships that are transferred to a recovery node may be automatically dissolved upon completion of the recovery operation.

[0043] Similarly, in an embodiment in which the affinity relationships are transferred to a failover node, the affinity relationships may be dissolved when the failover node completes the execution of the transactions that were transferred from the terminated node.

[0044] In an alternative embodiment, the affinity relationships are not automatically dissolved upon completion of any specific task. Instead, the affinity relationships continue until affinity end conditions have been satisfied. Affinity end conditions may vary from implementation to implementation. Affinity end conditions are described in greater detail in the Related Applications.

EXAMPLE PROCESS FLOW

[0045] FIG. 2 is a flowchart illustrating steps for responding to termination of a node in a system that implements an embodiment of the techniques described above. Referring to FIG. 2, at step 200, a node N1 terminates. As mentioned above, the termination may be planned or unplanned.

[0046] At step 202, a node N2 is selected to be both the recovery node and the failover node for N1. N2 may be based on a variety of factors, including memory capacity, processing capacity, and current workload.

[0047] At step 204, the affinity relationships of N1 are transferred to N2, causing the resources that belong to any of the objects in those relationships to be remastered at N2. At step 206, the remaining resources that were mastered by N1 are remastered across the surviving nodes using a hash function.

[0048] At step 208, N2 recovers the to-be-recovered resources of N1, and executes the transactions that were transferred from N1.

[0049] At step 210, the affinity relationships that were transferred to N2 as a result of the termination of N1 are dissolved, causing the resources that belonged to objects involved in those relationships to be remastered across the surviving nodes using the hash function.

HARDWARE OVERVIEW

[0050] Figure 3 is a block diagram that illustrates a computer system 300 upon which an embodiment of the invention may be implemented. Computer system 300 includes a bus 302 or other communication mechanism for communicating information, and a processor 304 coupled with bus 302 for processing information. Computer system 300 also includes a main memory 306, such as a random access memory (RAM) or other dynamic storage device, coupled to bus 302 for storing information and instructions to be executed by processor 304. Main memory 306 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 304. Computer system 300 further includes a read only memory (ROM) 308 or other static storage device coupled to bus 302 for storing static information and instructions for processor 304. A storage device 310, such as a magnetic disk or optical disk, is provided and coupled to bus 302 for storing information and instructions.

[0051] Computer system 300 may be coupled via bus 302 to a display 312, such as a cathode ray tube (CRT), for displaying information to a computer user. An input device 314, including alphanumeric and other keys, is coupled to bus 302 for communicating information and command selections to processor 304. Another type of user input device is cursor control 316, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor 304 and for controlling cursor movement on display 312. 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.

[0052] The invention is related to the use of computer system 300 for implementing the techniques described herein. According to one embodiment of the invention, those techniques are performed by computer system 300 in response to processor 304 executing one or more sequences of one or more instructions contained in main memory 306. Such instructions may be read into main memory 306 from another machine-readable medium, such as storage device 310. Execution of the sequences of instructions contained in main memory 306 causes processor 304 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.

[0053] The term "machine-readable medium" as used herein refers to any medium that participates in providing data that causes a machine to operation in a specific fashion. In an embodiment implemented using computer system 300, various machine-readable media are involved, for example, in providing instructions to processor 304 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 or magnetic disks, such as storage device 310. Volatile media includes dynamic memory, such as main memory 306. Transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise bus 302. Transmission media can also take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0054] Common forms of machine-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.

[0055] Various forms of machine-readable media may be involved in carrying one or more sequences of one or more instructions to processor 304 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 300 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 302. Bus 302 carries the data to main memory 306, from which processor

304 retrieves and executes the instructions. The instructions received by main memory 306 may optionally be stored on storage device 310 either before or after execution by processor 304.

[0056] Computer system 300 also includes a communication interface 318 coupled to bus 302. Communication interface 318 provides a two-way data communication coupling to a network link 320 that is connected to a local network 322. For example, communication interface 318 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 318 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 318 sends and receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

[0057] Network link 320 typically provides data communication through one or more networks to other data devices. For example, network link 320 may provide a connection through local network 322 to a host computer 324 or to data equipment operated by an Internet Service Provider (ISP) 326. ISP 326 in turn provides data communication services through the world wide packet data communication network now commonly referred to as the "Internet" 328. Local network 322 and Internet 328 both use electrical, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link 320 and through communication interface 318, which carry the digital data to and from computer system 300, are exemplary forms of carrier waves transporting the information.

[0058] Computer system 300 can send messages and receive data, including program code, through the network(s), network link 320 and communication interface 318. In the Internet example, a server 330 might transmit a requested code for an application program through Internet 328, ISP 326, local network 322 and communication interface 318.

[0059] The received code may be executed by processor 304 as it is received, and/or stored in storage device 310, or other non-volatile storage for later execution. In this manner, computer system 300 may obtain application code in the form of a carrier wave.

[0060] In the foregoing specification, embodiments of the invention have been described with reference to numerous specific details that may vary from implementation to implementation. Thus, the sole and exclusive indicator of what is the invention, and is intended by the applicants to be the invention, is the set of claims that issue from this application, in the specific form in which such claims issue, including any subsequent correction. Any definitions

expressly set forth herein for terms contained in such claims shall govern the meaning of such terms as used in the claims. Hence, no limitation, element, property, feature, advantage or attribute that is not expressly recited in a claim should limit the scope of such claim in any way. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

CLAIMS

What is claimed is:

1. A method for responding to termination of a terminated node, the method comprising:

causing a particular node to master all resources that belong to a particular object by transferring to the particular node an affinity relationship that was between the particular object and the terminated node;

causing the particular node to perform at least one of:

recovering resources that were opened by the terminated node; and

serving as a failover node to execute one or more transactions that had been executing on the terminated node.

2. The method of Claim 1 wherein the step of causing the particular node to perform includes causing the particular node to perform both of:

recovering resources that were opened by the terminated node; and

serving as a failover node to execute one or more transactions that had been executing on the terminated node.

3. The method of Claim 1 wherein the step of causing the particular node to perform includes causing the particular node to recover resources that were opened by the terminated node.

4. The method of Claim 1 wherein the step of causing the particular node to perform includes causing the particular node to serve as a failover node to execute one or more transactions that had been executing on the terminated node.

5. The method of Claim 3 further comprising dissolving said affinity relationship in response to the particular node completing the recovery of resources that were opened by the terminated node.

6. The method of Claim 4 further comprising dissolving said affinity relationship in response to the particular node completing execution of the one or more transactions that had been executing on the terminated node.

7. The method of Claim 1 wherein:

at the time of termination, the terminated node was master of a set of resources that did not belong to the particular object;

in response to termination of the terminated node, resources within said set of resources are randomly remastered among a plurality of remaining nodes.

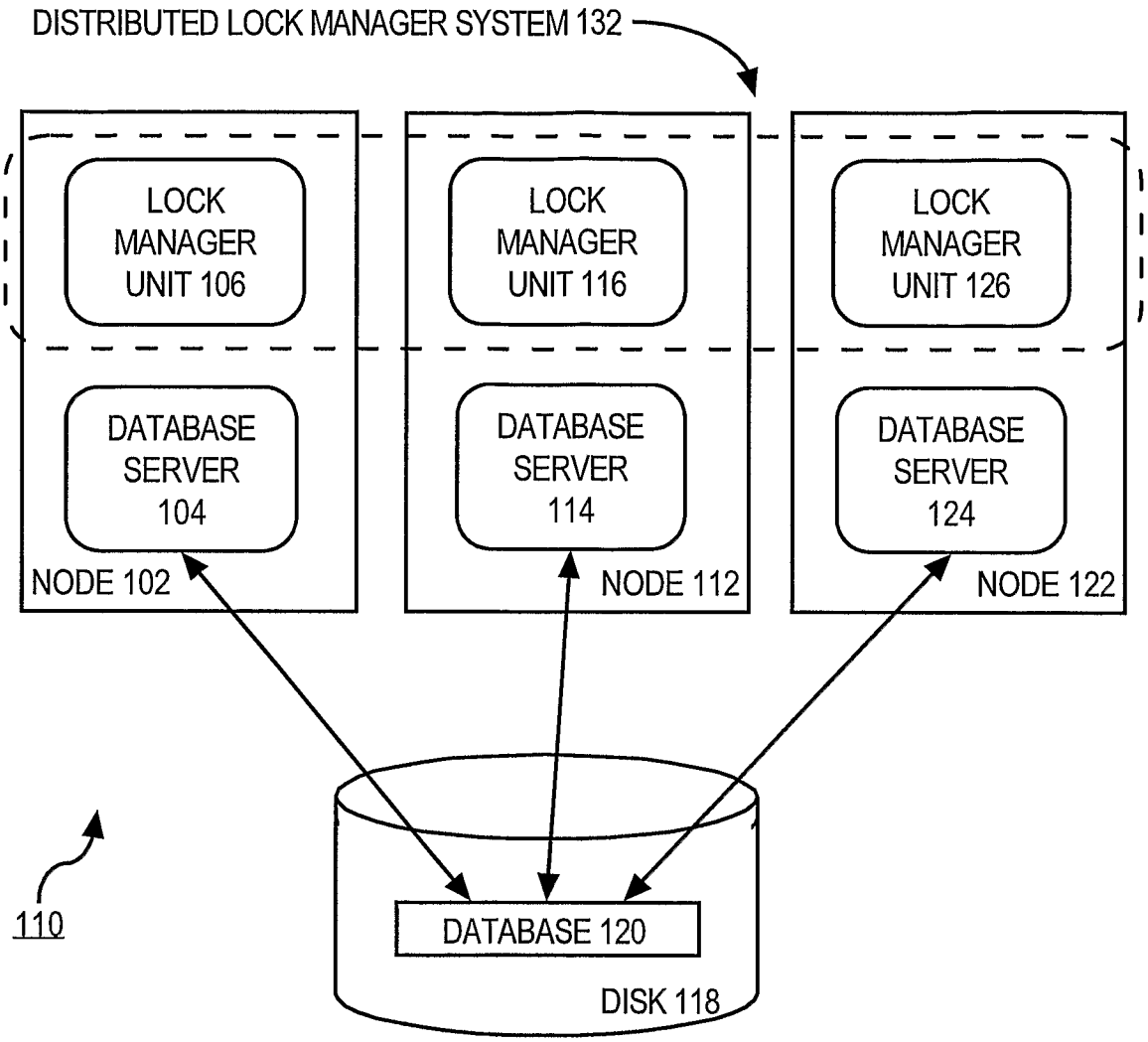
8. The method of Claim 1 wherein termination of the terminated node is an unplanned termination caused by a failure.

9. The method of Claim 1 wherein termination of the terminated node is a planned termination.

10. The method of Claim 3 further comprising selecting the particular node to be the recovery node for the terminated node based on one or more characteristics of the particular node.

11. The method of Claim 4 further comprising selecting the particular node to be the failover node for the terminated node based on one or more characteristics of the particular node.

12. A computer-readable medium carrying one or more sequences of instructions which, when executed by one or more processors, causes the one or more processors to perform the method recited in any one of Claims 1-11.



(Prior Art)

FIG. 1

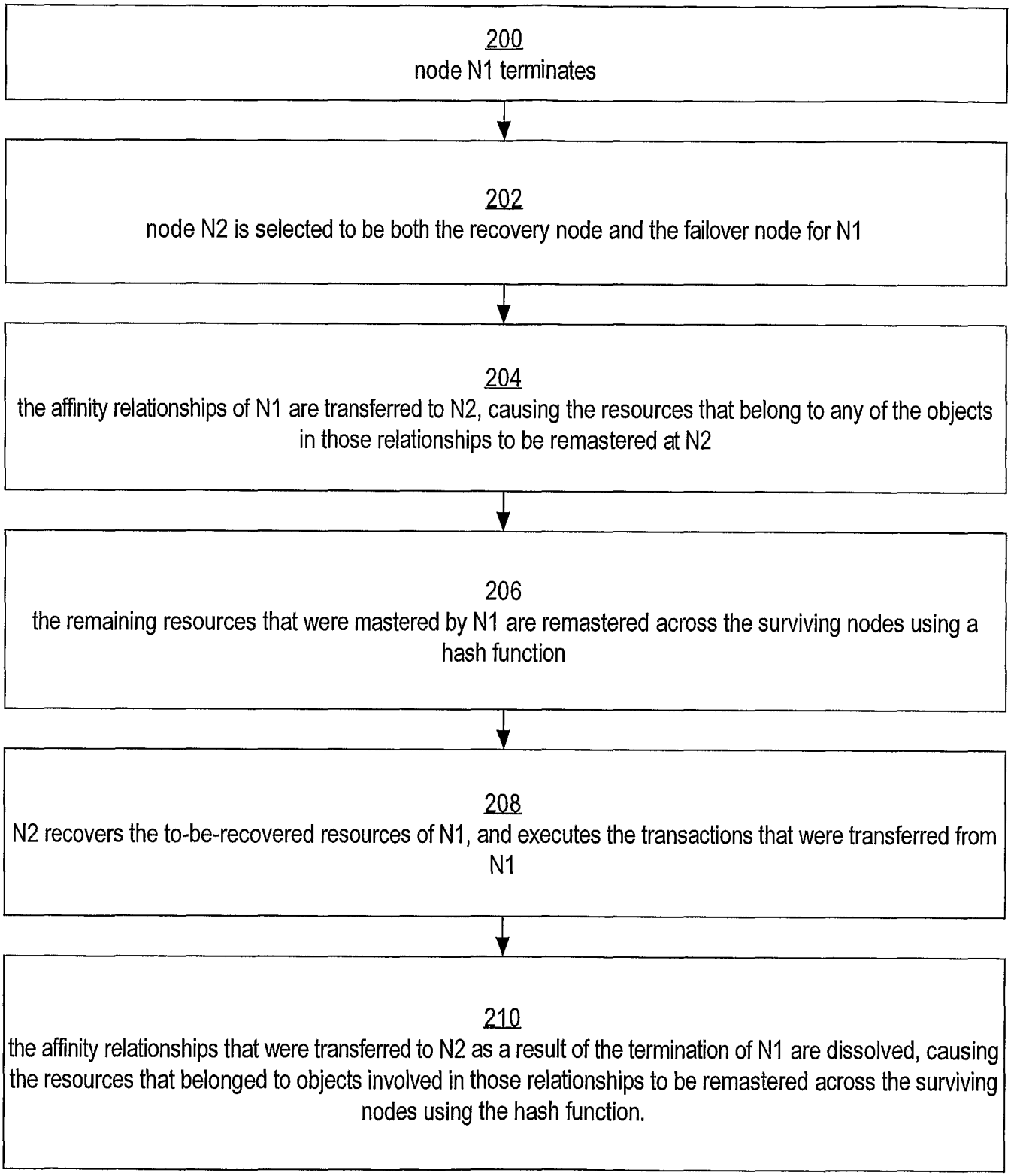

200
node N1 terminates
202
node N2 is selected to be both the recovery node and the failover node for N1
204
the affinity relationships of N1 are transferred to N2, causing the resources that belong to any of the objects in those relationships to be remastered at N2
206
the remaining resources that were mastered by N1 are remastered across the surviving nodes using a hash function
208
N2 recovers the to-be-recovered resources of N1, and executes the transactions that were transferred from N1
210
the affinity relationships that were transferred to N2 as a result of the termination of N1 are dissolved, causing the resources that belonged to objects involved in those relationships to be remastered across the surviving nodes using the hash function.

FIG. 2

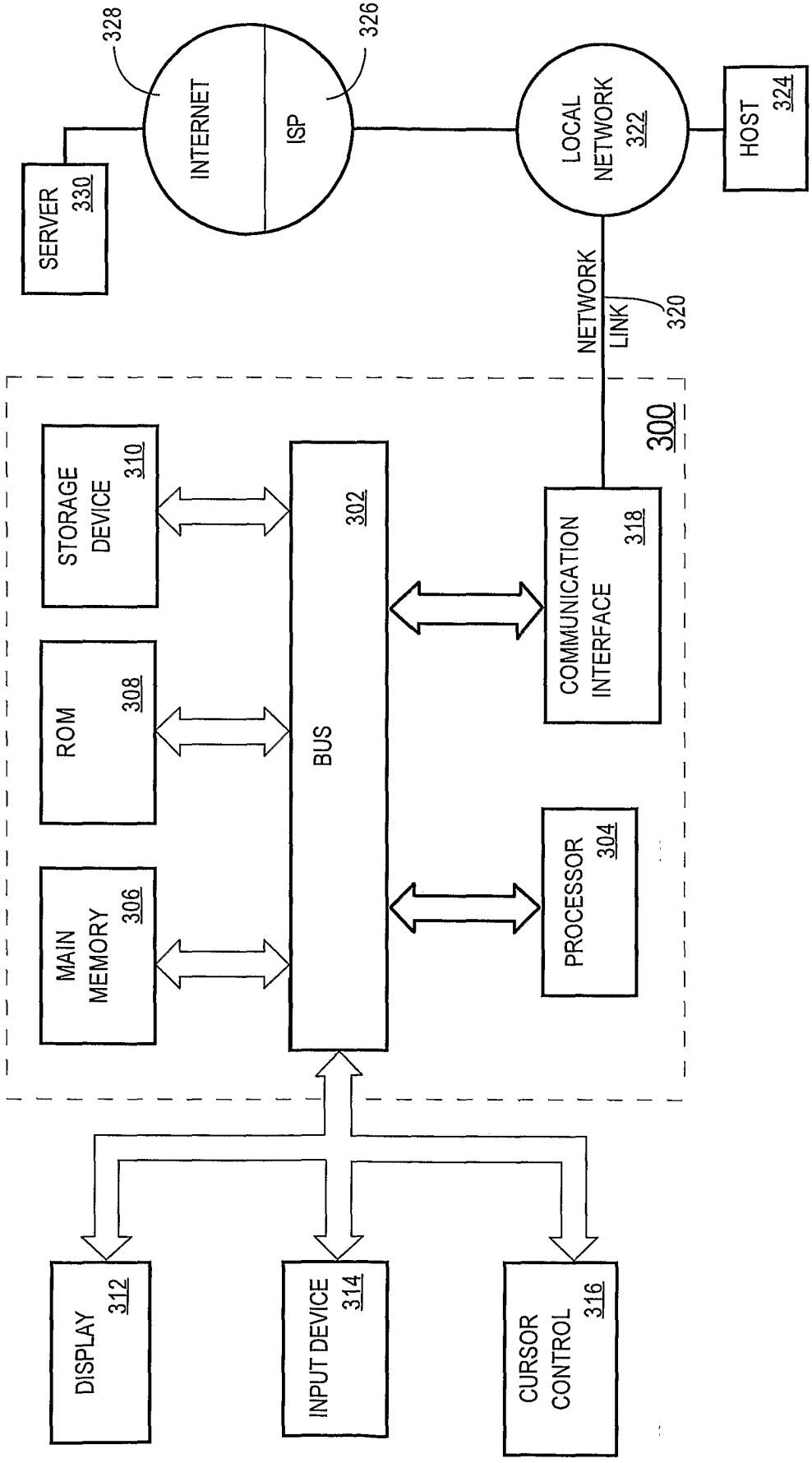

328
INTERNET
ISP
326
SERVER 330
LOCAL NETWORK 322
HOST 324
NETWORK LINK 320
300
STORAGE DEVICE 310
ROM 308
MAIN MEMORY 306
BUS 302
COMMUNICATION INTERFACE 318
PROCESSOR 304
DISPLAY 312
INPUT DEVICE 314
CURSOR CONTROL 316

FIG. 3

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2006/030961

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F11/20

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)
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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6 529 906 B1 (CHAN WILSON WAI SHUN [US]) 4 March 2003 (2003-03-04) abstract column 1, lines 18-21 column 2, line 8 - column 5, line 67 column 9, line 50 - column 10, line 3	1-12
X	US 6 115 830 A (ZABARSKY JEFFREY A [US] ET AL) 5 September 2000 (2000-09-05) abstract figures 3,4 column 1, line 21 - column 2, line 2 column 2, line 36 - column 3, line 19 column 5, line 6 - column 6, line 59 column 12, line 60 - column 13, line 30	1-12

-/--

[X] Further documents are listed in the continuation of Box C.

[X] See patent family 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	Date of mailing of the international search report
9 January 2007	18/01/2007

Name and mailing address of the ISA/	Authorized officer
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	Weber, Vincent

Form PCT/ISA/210 (second sheet) (April 2005)

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2006/030961

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6 249 879 B1 (WALKER BRUCE J [US] ET AL) 19 June 2001 (2001-06-19) abstract figures 2,3,5 column 1, line 23 - column 3, line 40 column 5, line 25 - column 6, line 61 column 9, line 33 - column 10, line 5	1-12
X	US 6 178 529 B1 (SHORT ROBERT T [US] ET AL) 23 January 2001 (2001-01-23) abstract figures 2,3 column 1, lines 41-46,62-65 column 4, line 18 - column 6, line 10 column 7, lines 33-45 column 9, line 60 - column 10, line 9	1-12
X	US 2002/073354 A1 (SCHROIFF KLAUS [DE] ET AL) 13 June 2002 (2002-06-13) abstract paragraphs [0014], [0015], [0044] - [0067]; figures 2,3	1-12

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2006/030961

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6529906 B1	04-03-2003	NONE	
US 6115830 A	05-09-2000	NONE	
US 6249879 B1	19-06-2001	NONE	
US 6178529 B1	23-01-2001	NONE	
US 2002073354 A1	13-06-2002	CN 1336589 A	20-02-2002
		DE 10134492 A1	21-02-2002
		JP 2002091938 A	29-03-2002
		KR 20020010490 A	04-02-2002
		SG 99917 A1	27-11-2003
		US 2006010338 A1	12-01-2006