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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
25 March 2010 (25.03.2010)

PCT

(10) International Publication Number
WO 2010/033877 A1

(51) **International Patent Classification**:
G06F 17/30 (2006.01)

(21) **International Application Number**:
PCT/US2009/057590

(22) **International Filing Date**:
18 September 2009 (18.09.2009)

(25) **Filing Language**: English

(26) **Publication Language**: English

(30) **Priority Data**:
61/192,668 19 September 2008 (19.09.2008) US
61/099,872 24 September 2008 (24.09.2008) US

(71) **Applicant** *(for all designated States except US)*: **ORACLE INTERNATIONAL CORPORATION** [US/US]; 500 Oracle Parkway, Mail Stop 5OP7, Redwood Shores, California 94065 (US).

(72) **Inventors; and**

(75) **Inventors/Applicants** *(for US only)*: **LEE, Sue, K.** [US/US]; 1023 Rudder Lane, Foster City, California 94404 (US). **KOLLA, Vivekananda, C.** [IN/US]; 7844 McClellan Road, Cupertino, California 95014 (US). **SHAH, Akshay, D.** [IN/US]; 792 Mayten Tree Court, Sunnyvale, California 94086 (US). **CHATTERJEE, Sumanta** [US/US]; 3183 Belmont Terrace, Fremont, California 94539 (US). **SUSAIRAJ, Margaret** [IN/US]; 1502 Kennewick Drive, Sunnyvale, California 94087 (US). **LOAIZA, Juan, R.** [US/US]; 470 Las Pulgas Drive, Woodside, California 94062 (US). **TSUKERMAN, Alexander** [US/US]; 30 Port Royal, Foster City, California 94404 (US). **SUBRAMANIAM, Sridhar** [IN/US]; 10281 Torre Avenue, Unit 822, Cupertino, California 95014 (US).

(74) **Agents**: **HICKMAN, Brian, D.** et al.; 2055 Gateway Place, Suite 550, San Jose, California 95110 (US).

(81) **Designated States** *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, ST, SV, 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, 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, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

[Continued on next page]

(54) **Title**: STORAGE-SIDE STORAGE REQUEST MANAGEMENT

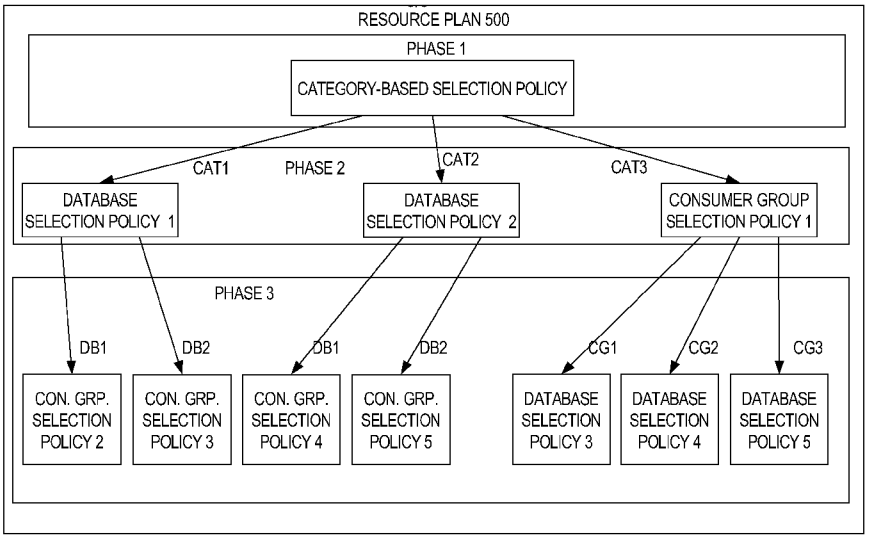

FIG. 5

(57) **Abstract**: Techniques are provided for managing, within a storage system, the sequence in which I/O requests are processed by the storage system based, at least in part, on a one or more logical characteristics of the I/O requests. The logical characteristics may include, for example, the identity of the user for whom the I/O request was submitted, the service that submitted the I/O request, the database targeted by the I/O request, an indication of a consumer group to which the I/O request maps, the reason why the I/O request was issued, a priority category of the I/O request, etc. Techniques are also provided for automatically establishing a scheduling policy within a storage system, and for dynamically changing the scheduling policy in response to changes in workload.

WO 2010/033877 A1

Published:

— *with international search report (Art. 21(3))*

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

STORAGE-SIDE STORAGE REQUEST MANAGEMENT

FIELD OF THE INVENTION

[0001] The present invention relates to storage request management and, more specifically, to storage-side management of I/O requests.

BACKGROUND

[0002] The primary role of database systems is to translate high-level, abstract declarations of data objects, and manipulations and searches for those objects. Ideally, those declarations and manipulations are translated efficiently by database systems into I/O requests sent to simple, linearly addressed, blocked, persistent storage devices. Often, database systems must perform such translations for multiple applications. Those applications may use multiple databases across different user and schema security levels, application types, database session types, and priorities and classes of I/O requests.

[0003] In many situations, the order in which I/O requests are sent to a storage system impacts the efficiency of the system. For example, when both a small, high-priority I/O request and a large, low-priority I/O requests need to be sent to a storage system, it would not be efficient for a database server to send the large, low-priority I/O request ahead of the small high-priority I/O request.

[0004] Various techniques have been developed to ensure that I/O requests are sent to storage systems in an intelligent manner. For example, U.S. Patent Application No. 11/716,364, entitled "Management Of Shared Storage I/O Resources", which is incorporated herein by this reference, describes techniques for queuing I/O requests within a database server, and issuing those requests to a storage system in an intelligent manner. However, having database servers issue I/O requests in an intelligent manner does not ensure the optimal handling of the I/O requests by the storage system.

[0005] Specifically, storage systems typically handle I/O requests on a First-In-First-Out (FIFO) basis. To the extent that storage systems deviate from FIFO processing, the deviation involves reordering I/O requests to improve disk efficiency. Such reordering is performed without any regard for the purpose behind the I/O requests. Consequently, when multiple

database servers are sending I/O requests for multiple databases to the same storage system (or the same storage device within a storage system), the storage system may end up processing lower priority I/O requests before processing higher priority I/O, even though each individual database server is independently issuing its I/O requests to the storage system in an optimal sequence.

[0006] Multiple types of workloads and databases often share storage. Unfortunately, running multiple types of workloads and databases on shared storage often leads to performance and response time problems between applications and mixed workloads. For example, large parallel queries on one production data warehouse can impact the performance of critical queries on another production data warehouse. Also, a data load on a data warehouse can impact the performance of critical queries also running on the same data warehouse. It is possible to mitigate these problems by over-provisioning the storage system, but this diminishes the cost-savings benefit of shared storage. It is also possible to schedule non-critical tasks at off-peak hours, but this manual process is laborious. It becomes impossible when databases have different administrators who do not coordinate their activities.

[0007] 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

[0008] 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:

[0009] FIG. 1 is a block diagram of a storage server configured to received requests from, and intelligently order, I/O requests from multiple I/O requestors;

[0010] FIG. 2 is a block diagram illustrating how various request metadata values may be mapped to consumer groups, according to an embodiment of the invention;

[0011] FIG. 3 is a block diagram of a three-phase selection policy, according to an embodiment of the invention;

[0012] FIG. 4 is a block diagram illustrating per-survivor-group policies, according to an embodiment of the invention;

[0013] FIG. 5 is a block diagram illustrating two phases of per-survivor-group policies, according to an embodiment of the invention; and

[0014] FIG. 6 is a block diagram of a computing device upon which embodiments of the invention may be implemented.

DETAILED DESCRIPTION

[0015] 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.

GENERAL OVERVIEW

[0016] Techniques are described herein for managing, within a storage system, the sequence in which I/O requests are processed by the storage system based, at least in part, on one or more logical characteristics of the I/O requests. The logical characteristics may include, for example, the identity of the user for whom the I/O request was submitted, the service that submitted the I/O request, the database targeted by the I/O request, an indication of a consumer group to which the I/O request maps, the reason why the I/O request was issued, a priority category of the I/O request, etc. Such logical characteristics differ fundamentally from the physical characteristics of the I/O requests, such as the storage location of the data targeted by the I/O request and the amount of data involved in the I/O operation specified by the I/O request.

[0017] The in-storage I/O management techniques described herein may be employed instead of or in addition to any management performed within I/O requestors that are issuing the I/O requests to the storage system. As used herein, "I/O requestor" refers to any entity that issues an I/O request to a storage system. I/O requestors may be, for example, database servers that issue the I/O requests in response to database commands received from database applications. However, the techniques described herein are not limited to any particular type of I/O requestor.

[0018] Because the storage system handles all I/O requests that are directed the storage devices within the storage system, the storage system is better situated than the I/O requestors to make intelligent scheduling decisions about how to schedule the I/O requests

that have been directed to those storage devices. For example, the storage system is able to determine that a high-priority I/O request that targets a particular storage device should be handled before a low-priority I/O request that targets the same storage device, even though the two requests came from different I/O requestors and target two different databases.

[0019] The logic, contained within the storage system, for managing the I/O requests is referred to herein as the "storage server" of the storage system. A storage server may be implemented by software executing on processors that are part of the storage system, by hard-wired logic, or by any combination of software and hardware. The I/O requests that have been received by the storage server, but have not yet been issued (e.g. added to an output queue of the storage system), are referred to herein as "pending I/O requests". I/O requests that have been issued to a storage device, but have not yet been completed, are referred to herein as "outstanding I/O requests".

[0020] The logical characteristics of the work associated with an I/O request are not conventionally available to a storage system. Therefore, in one embodiment, the logical characteristics of I/O requests are communicated to the storage system from other entities in the form of "request metadata". For example, in one embodiment, the I/O requestors provide the storage server with request metadata along with the I/O requests. As explained above, the request metadata specifies logical characteristics about the work represented by the I/O requests. Based on those logical characteristics, a scheduling policy, and a selection policy, the storage server determines when to issue each I/O request. As shall be described in greater detail below, the storage server may immediately issue an I/O request, or may queue the request for deferred issuance.

[0021] In one embodiment, when more than one I/O request is pending and the scheduling policy indicates that an I/O request should be issued, which pending I/O request to issue is determined by one or more selection policies. The various policies specified in a selection policy may be applied in phases. During each phase, the number of pending I/O requests that are considered candidates for selection is reduced based on the policies that belong to that phase. When only one non-empty waiting queue of pending I/O requests remains as a candidate, the storage server selects the I/O request that resides at the head of that waiting queue for processing.

[0022] Examples of policies that may be specified in a selection policy include:

- 70% of storage performance capacity should be allocated to data warehouse finance, and 30% of storage performance capacity should be allocated to data warehouse sales
- Production databases should have priority over test and development databases
- OLTP workloads should have priority over maintenance workloads

EXAMPLE SYSTEM

[0023] FIG. 1 is a block diagram of a system on which the I/O request management techniques described herein may be implemented. Referring to FIG. 1, a storage system 100 provides storage for data used by several different applications 130, 132, 134, 136 and 138. Those applications make use of storage system 100 indirectly, by sending commands to I/O requestors 120, 122 and 124. For example, applications 130, 132, 134, 136 and 138 may be various database applications, and I/O requestors 120, 122 and 124 may be database servers that respond to the database commands by storing, retrieving, and manipulating data stored on databases spread over storage devices 104 and 106.

[0024] In the illustrated embodiment, applications 130 and 132 interact with I/O requestor 120, application 134 interacts with I/O requestor 122, and applications 136 and 138 interact with I/O requestor 124. In actual implementations, the number and type of applications that concurrently interact with an I/O requestor may vary.

[0025] As mentioned above, I/O requestors 120, 122 and 124 may implement some form of internal I/O request scheduling policies. However, those internal policies do not ensure intelligent scheduling when, for example, both I/O requestor 120 and I/O requestor 122 submit I/O requests that target storage device 104. To intelligently manage I/O requests under such circumstances, policies are registered with storage server 102, and storage server 102 applies those policies to incoming I/O requests.

SCHEDULING AND SELECTION POLICIES

[0026] According to one embodiment, there are two separate policies within the storage server 102: a scheduling policy and a selection policy. The scheduling policy determines when an I/O should be issued. For example, the scheduling policy might be aimed at keeping throughput reasonably high. In this case, the storage server 102 would allow a relatively high number of outstanding I/Os. I/Os would only be queued once this limit is hit. Another example would be a scheduling policy that aims to keeping the latencies reasonably low. In

this case, the storage server 102 would allow a relatively low number of outstanding I/Os. In either case, once this limit is hit, any new I/Os would be queued. When the number of outstanding I/Os (the current I/O load on the disks) has reached a certain level, "saturation" occurs.

[0027] When, based on the scheduling policy, the storage server 102 decides to issue an I/O request, the storage server 102 decides which I/O to issue based on the selection policy. The techniques described herein may be used with any selection policy. For the purpose of explanation, various types of selection policies will be described hereafter. However, the techniques described herein are not dependent on the type of selection policy used. Typically, the selection policy implemented by the storage server 102 will hinge, for example, on the way the storage is to be used.

[0028] Typically, in situations where the storage is shared by multiple databases, the selection policy will include a policy for picking an I/O request based on the database targeted by the I/O request. As shall be described in greater detail hereafter, the policy for picking the database can be ratio-based, priority-based, a hybrid of the two.

[0029] On the other hand, if the storage is used by a single database, then the selection policy may include a policy for picking an I/O request based on workload associated with the I/O request. The workload-based selection policy can also be ratio-based, priority-based, or a hybrid of the two.

[0030] Finally, if the storage is shared by multiple databases and the databases have multiple workloads within them, then the selection policy may involve a two-phased plan, where the first phase of the plan selects a database, and the second phase of the plan selects a workload within that database.

[0031] According to one embodiment, the techniques described herein, as well as the scheduling policies and selection policies, are applied independently to each storage device within the storage system. Thus, if there are twelve disks in the storage system, then the load settings, queues, etc. are specified per disk. Per-disk load settings, queues and policies are useful because, while all disks are managed by the same storage system, each disk operates independent of the others.

THE SCHEDULING POLICY

[0032] As mentioned above, the storage server 102 issues I/O requests to the storage system up to the point that the storage system reaches a target, optimal load. As used herein, the term "load" refers to the total cost of the outstanding, incomplete I/O requests. The cost of an individual I/O request is determined by the amount of time the I/O request will utilize the storage system resources. Therefore, an I/O request to read a small amount of data has a much lower cost than an I/O request to read a large amount of data, since it occupies the disk I/O resources for a much shorter amount of time.

[0033] According to one embodiment, the cost of an I/O request is pre-determined by calibrating the storage system. Once the target load has been reached, the storage system queues all subsequent I/O requests. When enough I/O requests have completed so that the storage system is under its target load, the storage server uses the selection policy to choose enough I/O requests to bring the storage system back to its target load.

[0034] The setting of the target load determines the performance characteristics of the storage system. If the target load is low, then the latency of the I/O requests will be low. As the target load is increased, then the throughput of the storage will increase, leading to better I/O throughput. This is because storage systems work more efficiently at higher loads. However, the latency of the I/O requests will decrease, due to the increased load.

[0035] The size of the target load is referred to herein as the “load setting”. The load setting that is implemented by the scheduling policy can be established in many different ways. For example, the load setting could be hard-coded at a level that should work well for most people. Alternatively, the load setting could be set by the storage administrator specifying whether the storage system should be optimized for latency, throughput, or a compromise between the two. As another example, the load setting can be automatically determined by the storage server by looking at each workload. Using the meta-data and other workload characteristics, the storage server can determine whether the workload would be preferred to be optimized for latency or throughput. The storage server can use the selection policy to weigh the preference of each workload to determine an overall load setting. For example, the preference of a high priority database would be weighed more than the preference of a lower priority database.

[0036] In a system where the load setting is automatically determined by the storage server, the storage system may select the scheduling policy based on the usage characteristics

of the storage system, as reflected in the registered selection policy(s) and/or the request metadata. Thus, if the I/O requests received by the storage system are primarily for data warehousing operations, which tend to be throughput intensive, the storage system may select scheduling policy that that has a higher load setting. For example, the storage system may be configured to allow the storage system's output I/O queue to include up to eight one-Megabyte I/Os requests at any given time. A higher load setting allows a greater volume of I/Os to be issued to the storage devices within the storage system, keeping the storage devices as busy as possible.

[0037] On the other hand, if the majority of the I/O requests are for OLTP workloads, a scheduling policy with a lower load setting may be selected, to optimize for latency rather than throughput. Because of the lower load setting, I/O requests will tie up storage devices for shorter periods of time. Because the storage devices are tied up for shorter periods of time, the storage devices will be more available to process newly-arrived high-priority I/O requests. Thus, users that are performing the operations that produced those high-priority I/O requests will experience less storage system latency.

[0038] According to one embodiment, the storage system determines the characteristics of the overall workload based on the selection policies that are registered with the storage system and/or the request metadata that accompanies the requests. For example, a selection policy may indicate that applications for which the storage system is being used are data warehousing applications. Alternatively, the selection policy may not indicate the type of applications, but the request metadata received with the I/O requests may indicate that the vast majority of I/O requests are from data warehousing applications.

[0039] Even when selection policies indicate the nature of workloads that are using the storage system, the storage server may use the request metadata to determine how much each of the workloads is actually using the storage. For example, a selection policy may indicate that the storage system is being used by two OLTP applications and two warehousing applications. Based on this information alone, it may be unclear what the optimal load setting would be. However, if 90% of the I/O requests that are actually received by the storage server are from the two OLTP applications, then the storage system may select an load setting that is optimized for OLTP applications.

[0040] According to one embodiment, the load setting selection process is repeated periodically, so that the load settings can dynamically change as the workload changes. For

example, in one embodiment, the selection process is made based on the I/O requests that have been received by the storage system within a particular time period, such as the last five seconds. The workload represented by the I/O requests received during that period may differ significantly from the workload represented in I/O requests that were received in prior periods. The load setting is then changed to reflect the result of the most recent load setting selection operation. Thus, during periods where a storage server is used primarily for data warehouse operations, the load setting will automatically be optimized for throughput, and during periods in which the same storage system is used primarily for OLTP operations, the load setting will automatically be optimized for reduced latency.

REQUEST METADATA

[0041] To implement a selection policy, the storage server 102 needs to know information about the I/O requests that it receives. For example, if the selection policy is based on the workload associated with I/O requests, then storage server 102 needs to know the workloads to which the I/O requests belong. According to one embodiment, the information required by storage server 102 to implement the selection policy is provided to storage server 102 as "request metadata" that accompanies the requests. The metadata that accompanies the I/O requests, and how that metadata is used by storage server 102 to make intelligent decisions about I/O scheduling, shall be described in greater detail hereafter.

WAITING QUEUES

[0042] When the storage system 100 is operating below the capacity dictated by the scheduling policy, storage server 102 will not queue I/O requests. In particular, as long as the storage system's capacity is not saturated, I/O requests received by storage server 102 are immediately processed. The I/O requests may be processed, for example, by placing the I/O requests into an output I/O queue. However, whenever the I/O requests start to saturate storage system 100, storage server 102 will defer the execution of incoming I/O requests by placing the received I/O requests into waiting queues.

[0043] For example, if production and test databases are sharing storage system 100, selection policies can be configured that give priority to I/O requests that target the production database. In this case, whenever the test database load would affect the production database performance, storage server 102 will schedule the I/O requests such that

the production database I/O performance is not impacted. This means that the test database I/O requests will be placed in a waiting queue until they can be processed without disturbing the production database I/O performance.

[0044] The waiting queue into which a deferred I/O request is placed is based on the logical characteristics indicated by the request metadata that accompanies the I/O request. When the scheduling policy indicates that additional I/O requests should be issued, storage server 102 selects which deferred I/O requests to process based on the registered selection policies, and the waiting queue in which the I/O requests reside.

RATIO-BASED SELECTION POLICIES

[0045] According to one embodiment, the selection policies that are registered with storage server 102 may include ratio-based policies, priority-based policies, and hybrid policies. Ratio-based policies are policies that allocate the load between requests having different logical characteristics based on ratios that are assigned to those logical characteristics. For example, assume that the logical characteristic that is to be used as the basis for a ratio-based policy is the database that is targeted by the I/O requests (the "targeted-database characteristic"). Assume further that five databases DB1, DB2, DB3, DB4 and DB5 are stored on the storage devices 104 and 106. A ratio-based policy may specify that storage server 102 should allocate load among the five databases as follows:

[0046] DB1=50%, DB2=20%, DB3=20%, DB4=5% and DB5=5%.

[0047] Based on the ratio-based policy, storage server 102 processes pending I/O requests in a sequence to ensure that the I/O requests that target databases that are below their allocation ratios are processed before I/O requests that target databases that are above their allocation ratios.

[0048] In one embodiment, storage server 102 makes a "furthest behind" determination based on the ratios assigned to the various databases. For example, assume that, when storage server 102 is ready to select another I/O request to process, DB2, DB4 and DB5 have pending I/O requests. Under these circumstances, storage server 102 determines which of DB2, DB4 and DB5 is furthest behind in its respective allocation. For the purpose of explanation, assume that DB2 has already used 22% of the load, DB4 has used 4% and DB5 has used 3%. Under these circumstances, storage server 102 would select a pending I/O request that targets DB5.

[0049] Ratio-based policies are useful to ensure that higher priority entities receive a greater amount of bandwidth, while still ensuring that no entity is left out completely. In the above example, DB2 is generally more important than DB5 because DB2 has been allocated 20% of the load (while DB5 has been allocated only 5%). However, because DB2 had already used more than its allocated portion of the load, an I/O request from DB5 was processed ahead of an I/O request from DB2.

RATIO-BASED POLICIES USING PROBABILITIES

[0050] In the embodiment described above, a ratio-based policy is implemented by keeping track of which database that is "furthest behind" relative to its load allocation. Unfortunately, a "furthest behind" determination incurs the overhead of keeping track of which I/O requests have been previously issued. To avoid this overhead, the storage server 102 can simply treat the load allocations that are specified by the ratio-based policy as "probabilities" that I/O requests will be selected.

[0051] For example, assume that each of databases DB1, DB2, DB3, DB4 and DB5 has at least one pending I/O request. Under these conditions, all of the databases would be selection candidates. Consequently, the probability that a given database will be selected will equal the percentage of the load that the database was allocated. Specifically, there would be a 50% probability that the storage server 102 would select an I/O request that targets DB1, a 20% probability that the storage server 102 would select an I/O request that targets DB2, etc.

[0052] On the other hand, if only DB2 and DB5 have pending I/O requests, then DB2 and DB5 would be the only selection candidates. DB2 has a 20% load allocation and DB5 has a 5% load allocation. Under these circumstances, the relative load allocations of DB2 and DB5 would result in an 80% probability that the I/O request that targets DB2 would be selected, and a 20% probability that the I/O request that targets DB5 would be selected.

[0053] Once the probabilities have been determined for the various selection candidates, storage server 102 performs the I/O request selection based on the probabilities. One way of selecting an I/O request based on the probabilities involves assigning a sub-range of a range to each of the selection candidates, where the size of the sub-range that is assigned to each selection candidate is determined by the probability that the selection candidate will be selected.

[0054] For example, assume that the range is 1 to 100. Further assume that the only I/O request candidates are an I/O request that targets DB2, and an I/O request that targets DB5. Under these circumstances, the DB2 would have a selection probability of 80%. Therefore, the I/O request that targets DB2 may be assigned sub-range 1-80 (i.e. 80% of the entire range). On the other hand, DB5 would have a selection probability of 20%, and therefore may be assigned sub-range 81-100 (i.e. 20% of the entire range).

[0055] After each selection candidate has been assigned a sub-range within the range, storage server 102 may generate a random number within the range. The sub-range into which the random number falls determines which selection candidate the storage server 102 selects. Thus, in the present example, if the random number falls between 1 and 80, the I/O request that targets DB2 will be selected. On the other hand, if the random number falls between 81 and 100, then the I/O request that targets DB5 will be selected.

PRIORITY-BASED POLICIES

[0056] Priority-based policies are policies that assign a relative importance to the values of a logical characteristic of the candidate I/O requests. For example, assume that the logical characteristic that is to be used as the basis for a priority-based policy is the targeted-database characteristic. In a storage system that stores data for databases DB1, DB2, DB3, DB4 and DB5, the possible values of the targeted-database characteristic are DB1, DB2, DB3, DB4 and DB5. Consequently, a priority-based policy that uses the targeted-database characteristic may specify that I/O requests that target DB1, DB2, DB3, DB4 and DB5 have first, second, third, fourth and fifth priority, respectively.

[0057] When a priority-based policy is used, no I/O requests that have a lower-priority logical characteristic value are processed as long as there are pending I/O requests that are associated with a higher-priority logical characteristic value. Thus, as long as any I/O requests that target DB1 are pending, I/O requests that target DB2, DB3, DB4 and DB5 will not be selected by storage server 102. On the other hand, I/O requests that target DB5 will not be selected by storage server 102 until there are no pending I/O requests that target any of DB1, DB2, DB3, and DB4. Priority-based policies ensure that I/O requests associated with lower-priority logical characteristics never adversely impact the performance of I/O requests associated with higher-priority logical characteristics.

HYBRID POLICIES

[0058] Hybrid policies are policies that incorporate both priority levels and ratios. Specifically, multiple logical characteristics may be assigned to each priority level. Within each priority level, ratios are assigned to each of the logical characteristics. For example, I/O requests that target DB1, DB2 and DB3 may all be assigned to the first priority level. Within the first priority level, I/O requests that target DB1, DB2 and DB3 may be assigned the ratios 50, 40 and 10, respectively. I/O requests that target DB4 and DB5 may be assigned to the second priority level. Within the second priority level, I/O requests that target DB4 and DB5 may be assigned the ratios 70 and 30, respectively.

[0059] According to this example, storage server 102 will allocate load between I/O requests that target DB1, DB2 and DB3 according to their respective ratios. As long as any of the pending I/O requests target DB1, DB2 or DB3, storage server 102 will not process any I/O requests that target DB4 or DB5. If no pending I/O requests target DB1, DB2 or DB3, then the load will be allocated between I/O requests that target DB4 and DB5 based on the ratios 70 and 30, respectively.

DATABASE-SELECTION POLICIES

[0060] As illustrated in the examples given above, the database to which an I/O request is targeted is one logical characteristic that may be used as the basis for a policy. Policies for selecting between I/O requests based on the targeted-database characteristic are referred to herein as "database-selection policies". Examples of database-selection policies include:

- 70% of storage performance capacity should be allocated to data warehouse finance, and 30% of storage performance capacity should be allocated to data warehouse sales
- Production databases should have priority over test and development databases

[0061] Database-selection policies may be simple or complex. Table 1 illustrates a three-level hybrid database-selection policy:

TABLE 1

Database	Level 1	Level 2	Level 3
Sales Production Data Warehouse	80%		
Finance Production Data Warehouse	20%		
Customer Service Standby Database		100%	
Sales Test Database			50%
Sales Development Database			50%

[0062] Referring to Table 1, it illustrates database-selection policies for a system that includes five databases. At the first priority level, the Sales Production Data Warehouse is assigned 80% of the load, and the Finance Production Data Warehouse is assigned 20% of the load. As long as either of these two level 1 databases have pending I/Os, the storage server will not issue I/Os for any of the other three databases. Only when no I/O requests are pending for the level 1 databases will I/O requests that target the Customer Service Standby Database be added to an output I/O queue by the storage server. Similarly, only when no I/O requests are pending for level 1 or level 2 databases will the load be divided 50/50 between I/O requests that target the Sales Test Database and the Sales Development Database.

CONSUMER-GROUP-SELECTION POLICIES

[0063] A database often has many types of workloads. These workloads may differ in their performance requirements and the amount of I/O they issue. "Consumer groups" provide a way to group sessions that comprise a particular workload. For example, if a database stores data for four different applications, then four consumer groups can be created, one for sessions for each application. Similarly, if a data warehouse has three types of workloads, such as critical queries, normal queries, and ETL (extraction, transformation, and loading), then a consumer group can be created for each type of workload.

[0064] Thus, the "consumer group" of an I/O request is a logical characteristic of the I/O request that is derived from the values of one or more other logical characteristics of the I/O request. In one embodiment, the consumer group to which an I/O request belongs is based

on a "logical-characteristic-value-to-consumer-group" mapping that maps logical characteristic values to consumer groups. For example, the consumer group of an I/O request may be derived from one or more of the following logical characteristic values: (1) a service identifier, (2) a user identifier, (3) a program name, (4) a purpose identifier, and (5) statistics about the operation that submitted the I/O request.

[0065] Referring to FIG. 2, it is a block diagram illustrating a mapping between logical characteristic values and consumer groups, according to an embodiment of the invention. In the mapping illustrated in FIG. 2, three consumer groups have been defined: Priority DSS, DSS, and Maintenance. According to the illustrated mapping, the Priority DSS consumer group includes all I/O requests where either the service is "PRIORITY" or the username is "LARRY". The DSS consumer group includes all I/O requests where either the username is "DEV" or the query has been running for more than one hour. The Maintenance consumer group includes all I/O requests where either the client program name is "ETL" or the function of the I/O request is "BACKUP".

[0066] When consumer group membership is defined by logical-characteristic-value-to-consumer-group mappings, a single I/O request may have logical characteristic values that map to several different consumer groups. For example, an I/O request may have a user identifier of "LARRY", a client program identifier of "DEV" and a purpose identifier of "BACKUP". According to one embodiment, logical-characteristic-value-to-consumer-group mappings indicate how to resolve situations in which the logical characteristic values of an I/O request map to multiple consumer groups. For example, a logical-characteristic-value-to-consumer-group mapping may indicate that username is the most significant logical characteristic value, followed by service, program statistics, function, and finally client program. Under such a selection policy, an I/O request that has a user identifier of "LARRY", a client program identifier of "DEV" and a purpose identifier of "BACKUP" would be mapped to the Priority DSS consumer group.

[0067] A logical-characteristic-value-to-consumer-group mapping may also have a "default" consumer group. In one embodiment, if none of the logical characteristic values of an I/O request maps to any other consumer group, then the I/O request is treated as belonging to the default consumer group.

[0068] Similar to database-selection policies, consumer-group policies may be ratio-based, priority-based or hybrid. A hybrid consumer-group policy is illustrated in Table 2:

TABLE 2

Consumer Group	Level 1	Level 2
Priority DSS	80%	
Maintenance	20%	
DSS		100%

[0069] According to the hybrid consumer-group policy illustrated in Table 2, I/O requests that map to the Priority DSS consumer group are allocated 80% of the load, while I/O requests that map to the Maintenance consumer group are allocated 20% of the load. I/O requests that map to the DSS consumer group are only issued by the storage server when there are no pending I/O requests that belong to either of the other two consumer groups.

[0070] In one embodiment, I/O requestors use logical-characteristic-value-to-consumer-group mappings to determine the consumer-groups to which their I/O requests belong, and include a consumer-group identifier in the request metadata that they send with each I/O request. In embodiments where consumer-group identifiers are included in the request metadata, the logical characteristics from which the consumer-group is derived need not also be included in the request metadata.

[0071] In an alternative embodiment, the consumer-group of an I/O request is not directly indicated in the request metadata received by the storage system from the I/O requestors. Instead, the logical-characteristic-value-to-consumer-group mappings are provided to the storage system, and the storage server within the storage system determines the consumer group of each I/O request based on (a) the logical characteristic values reflected in the request metadata, and (b) the logical-characteristic-value-to-consumer-group mappings.

CATEGORY-SELECTION POLICIES

[0072] According to one embodiment, one logical characteristic of an I/O request is referred to herein as the "category" of the I/O request. Typically, the category of an I/O request indicates the type of workload associated with the I/O request. For example, a category-selection policy may define three categories: critical, somewhat-critical, and not-critical.

[0073] Category-selection policies indicate policies for selecting among I/O request candidates based on the category to which the I/O request candidates belong. Similar to database-selection policies and consumer-group based policies, category-selection policies may be ratio-based, priority-based, or hybrid. For example, a priority-based category-selection policy may specify that critical I/O requests have the highest priority, somewhat-critical I/O requests have medium priority, and not-critical I/O requests have low priority.

THE POLICY-TO-PHASE ASSIGNMENT

[0074] As mentioned above, the storage server may apply the policies specified in a selection policy in phases. The choice of which policies to assign to each phase is itself a policy decision that may be indicated in the selection policy. FIG. 3 is a block diagram illustrating a selection policy 300 in which policies have been assigned to three phases: phase 1, phase 2 and phase 3. In particular, a priority-based category-selection policy has been assigned to phase 1, a hybrid database-selection policy has been assigned to phase 2, and a hybrid consumer-group-selection policy has been assigned to phase 3.

[0075] Within the phase 2 hybrid database-selection policy, the first priority level includes DB1, DB2 and DB3, and the second priority level includes DB4 and DB5. Within the first priority level, DB1, DB2 and DB3 are associated with allocation ratios 50%, 40% and 10% respectively. Within the second priority level, DB4 and DB5 are both associated with allocation ratios of 50%.

[0076] Similarly, within the phase 3 hybrid consumer-group-selection policy, the first priority level includes consumer group 1 and consumer group 2, and the second priority level includes consumer group 3. Within the first priority level, consumer groups 1 and 2 are associated with allocation rations 80% and 20%, respectively. Within the second priority level, consumer group 3 has an allocation ratio of 100%.

[0077] The pending I/O requests that continue to be candidates after the selection policies in a phase have been applied are referred to herein as the "survivors" of the phase. For example, according to the selection policy 300 of FIG. 3, if there are any pending I/O requests that belong to category 1, then the survivors of phase 1 will only include the pending I/O requests belong to category 1.

[0078] On the other hand, if there are no pending I/O requests that belong to category 1, but there is at least one pending I/O request that belongs to category 2, then the survivors of

phase 1 will only include the pending I/O requests that belong to category 2. Finally, if there are no pending I/O requests that belong to categories 1 or 2, then the survivors of phase 1 will only include the pending I/O requests belong to category 3.

[0079] The policies of a subsequent phase are only applied to the survivors of the previous phase. Thus, the hybrid database-selection policies assigned to phase 2 in selection policy 300 are only applied to the survivors of phase 1. Similarly, the hybrid consumer-group-selection policy assigned to phase 3 is only applied to the survivors of phase 2. Phases continue to be applied until all of the survivors belong to the same waiting queue. The I/O request at the head of that waiting queue is then selected for execution.

[0080] Because the phases are applied in sequence, the policies applied in early phases have a greater impact on the ultimate I/O request selection than the policies applied in later phases. Consequently, the policies that deal with more significant logical characteristics are generally assigned to earlier phases than policies that deal with less significant logical characteristics.

[0081] For example, in situations where the users of various databases trust each other not to monopolize the load of the storage system, the targeted-database characteristic becomes a less significant logical characteristic. Consequently, the selection policy may not have any database-selection policies, or may assign the database-selection policies to a later phase in the I/O request selection process.

[0082] On the other hand, in situations where the users of the various databases might try to monopolize the load, the targeted-database characteristic may be the most significant logical characteristic. Under these circumstances, database-selection policies may be assigned to the first phase performed by the storage server to ensure that an agreed-upon bandwidth allocation between the databases is maintained.

PER-SURVIVOR-GROUP POLICIES

[0083] In the examples given above, the policies that are applied by the storage server in subsequent phases are independent of which pending I/O requests survived the previous phases. However, according to one embodiment, the policy that applies in a subsequent phase may hinge on the group of I/O requests that survived the previous phase. Policies that are dependent on which I/O requests survived previous phases are referred to herein as per-survivor-group policies.

[0084] For example, phase 2 of FIG. 3 has five possible outcomes. Specifically, because phase 2 applies a database-selection policy, the survivors of phase 2 of FIG. 3 will be I/O requests that target only one of the five databases. In one embodiment, the hybrid consumer-group-selection policy that is illustrated in phase 3 of FIG. 3 may only apply to I/O requests that target DB1. Phase 3 may have completely different policies if the I/O requests that survive phase 2 are I/O requests that target one of the other databases.

[0085] Referring to FIG. 4, it is a block diagram illustrating a selection policy 400 similar to selection policy 300, in which there is a distinct phase 3 policy for each of the five possible survivor groups of phase 2. The phase 3 policy for I/O requests that target one database may be completely different than the phase 3 policies for I/O requests that target another database. For example, the phase 3 policy that applies to I/O requests that target DB1 may be based on three consumer groups that are determined based on a particular logical-characteristic-value-to-consumer-group mapping, while the phase 3 policy that applies to I/O requests that target DB2 may be based on five consumer groups that are determined based on a different logical-characteristic-value-to-consumer-group mapping.

[0086] Because the policies that apply at a particular phase may depend on which I/O requests survive the previous phase, selection policies can be arbitrarily sophisticated. For example, FIG. 5 is a block diagram that illustrates the relationships between the possible survivor groups of each phase of a selection policy 500, and policies that are applied by the storage server in subsequent phases.

[0087] Referring to FIG. 5, phase 1 applies a category-based selection policy. The possible survivor groups produced by phase 1 include (a) I/O requests that belong to category 1, (b) I/O requests that belong to category 2, and (c) I/O requests that belong to category 1. In the case that phase 1 produces I/O requests that belong to category 1, the storage server will apply database selection policy 1 during phase 2. In the case that phase 1 produces I/O requests that belong to category 2, the storage server will apply database selection policy 2 during phase 2. In the case that phase 1 produces I/O requests that belong to category 3, the storage server will apply consumer group selection policy 1 during phase 2.

[0088] The possible survivor groups of database selection policy 1 include (a) category 1 I/O requests that target DB1, and (b) category 1 I/O requests that target DB2. In the case that category 1 I/O requests that target DB1 survive phase 2, the storage server will apply consumer group selection policy 2 during phase 3. In the case that category 1 I/O requests

that target DB2 survive phase 2, the storage server will apply consumer group selection policy 3 during phase 3.

[0089] The possible survivor groups of database selection policy 2 include (a) category 2 I/O requests that target DB1, and (b) category 2 I/O requests that target DB2. In the case that category 2 I/O requests that target DB1 survive phase 2, the storage server will apply consumer group selection policy 4 during phase 3. In the case that category 2 I/O requests that target DB2 survive phase 2, the storage server will apply consumer group selection policy 5 during phase 3.

[0090] The possible survivor groups of consumer group selection policy 1 include (a) category 3 I/O requests that map to consumer group 1 (CG1), (b) category 3 I/O requests that map to consumer group 2 (CG2), and (c) category 3 I/O requests that map to category group 3 (CG3). In the case that category 3 I/O requests that map to CG1 survive phase 2, the storage server will apply database selection policy 3 during phase 3. In the case that category 3 I/O requests that map to CG2 survive phase 2, the storage server will apply database selection policy 4 during phase 3. In the case that category 3 I/O requests that map to CG3 survive phase 2, the storage server will apply database selection policy 5 during phase 3.

[0091] FIG. 5 is merely one example of how the policies that are applied at subsequent phases may differ based on which I/O requests survive the prior phases. The policy relationships established in a selection policy may be arbitrarily complex. For example, given a certain set of outcomes, the storage server may only apply three phases of policies, while other outcomes may require application of five phases of policies.

DATABASE-SPECIFIC CONSUMER GROUPS

[0092] As illustrated in FIG. 5, each database may have its own consumer-group-selection policy. Further, the consumer-group selection policy for one database may be based on an entirely different attribute-to-consumer group mapping than the logical-characteristic-value-to-consumer-group mappings employed by other databases. For example, in FIG. 5, consumer-group selection policy 2 may be the hybrid policy illustrated in Table 2, while consumer-group selection policy 3 is the ratio-based policy: CG1: 90%, CG2: 8%, CG3: 1%, CG4: 1%.

[0093] In this example, the consumer group selection policies differ both with respect to the number of consumer groups, and with respect to the structure of the policies. Different

consumer group policies may also differ with respect to how the consumer groups are defined. For example, the attribute-to-consumer group mapping used to determine whether an I/O request belongs to Priority DSS, Maintenance, or DSS, may be completely different than the attribute-to-consumer group mapping used to determine whether an I/O request belongs to CG1, CG2, CG3, or CG4.

[0094] According to one embodiment, once consumer groups have been created, policies are created to specify how sessions are mapped to consumer groups. Sessions can be mapped to consumer groups based, for example, on session attributes. The session attributes may include, for example, a user name, the service that the session used to connect to the database, client machine, client program name, client user name, and so on. If a user is creating consumer groups for each application and each application has a dedicated service, then the user can create mapping policies based on service names. If a user wants to dedicate a consumer group to a particular set of users, then the user creates mapping policies based on their user names. As mentioned above, sessions that are not explicitly assigned to a consumer group may be placed in a default consumer group.

CREATING SELECTION POLICIES

[0095] As mentioned above, a selection policy may include policies that specify how I/O resources are allocated among consumer groups. In one embodiment, a selection policy contains a resource allocation directive for each consumer group, which consists of a percentage and a level. In one embodiment, a user may specify up to eight levels. Consumer groups at level 2 get resources that were not allocated at level 1 or were not consumed by a consumer group at level 1. Similarly, consumer groups at level 3 are allocated resources only when some allocation remains from levels 1 and 2. The same policies apply to levels 4 through 8. Multiple levels not only provide a way of prioritizing, but they provide a way of explicitly specifying how all primary and leftover resources are to be used. A user can construct selection policies that allocate resources across consumer groups using percentages, priorities, or a combination of the two.

ENABLING AND CHANGING SELECTION POLICIES

[0096] According to one embodiment, when a user sets a database selection policy on a database, a description of the selection policy is automatically sent to each storage system

used to store data for the database. In some environments, multiple servers share access to the same database. In such an environment, all database servers in the cluster that share access to a particular database are set to the same selection policy. When a new storage system is added to the database, or an existing storage system is restarted, the current selection policy of the database is automatically sent to the storage system. The selection policy is used to manage resources on both the database server and cells.

[0097] According to one embodiment, selection policies can be changed dynamically. Various types of events may trigger the selection policy change. For example, a selection policy may change at based on the time of day, to allow different selection policies to be in effect during the night than are in effect during the day. As another example, a selection policy change may occur during weekends, or at a particular time of year, such as the end of the fiscal year.

WAITING QUEUES

[0098] As mentioned above, deferred I/O requests are placed in waiting queues within storage system 100. These waiting queues may be implemented in a variety of ways. In one embodiment, the deferred I/O requests are queued based on their "selection group", where each selection group has a separate queue. In this context, a selection group is a group of I/O requests that are treated the same by the policies that are in effect. For example, assume that the only policy in effect is the consumer group policy illustrated in Table 2. This policy establishes three consumer groups: Priority DSS, Maintenance, and DSS. If a selection policy includes only these policies, then the storage system would only have three waiting queues, one for each of the three consumer groups. All deferred I/O requests that have characteristic values that map to Priority DSS will be stored in the Priority DSS queue, regardless of other characteristics (such as the database they target). Similarly, all deferred I/O requests that map to the Maintenance and DSS consumer groups would be place into the waiting queues corresponding to those consumer groups, regardless of the other characteristic values they may have.

[0099] In more complex selection policies, there may be significantly more selection groups than consumer groups. For example, in selection policy 300 illustrated in FIG. 3, the category, target database and consumer groups all have an effect on how I/O requests are treated. Therefore, each (category, target database, consumer group) combination represents

a distinct selection group that may have its own waiting queue. Specifically, (category 1, DB1, consumer group 1) would have one waiting queue, and (category 1, DB1, consumer group 2) would have another waiting queue. Because selection policy 300 establishes three categories, five databases and three consumer groups, the total number of selection groups (and therefore waiting queues) would be 3x5x3 = 45.

PHASE-BASE SELECTION OF A SELECTION GROUP

[0100] As mentioned above, when the storage server is selecting a deferred I/O request to process, the storage server applies the policies of the selection policy in phases. Each phase reduces the number of selection groups that survive. When only one selection group survives, the storage server selects the I/O request at the head of the waiting queue that corresponds to that selection group. For example, assume that a storage system implementing selection policy 300 becomes unsaturated and must select a deferred I/O request to process. Further assume that the only non-empty waiting queues correspond to selection groups: (category 2, DB1, consumer group 1), (category 2, DB2, consumer group 2), (category 2, DB2, consumer group 3), (category 3, DB2, consumer group 1), and (category 2, DB4, consumer group 3).

[0101] Under these circumstances, only those selection groups that are associated with category 2 would survive phase 1, because category 2 has higher priority than category 3, and there are no non-empty selection groups associated with category 1. Thus, after phase 1, the surviving selection groups would be: (category 2, DB1, consumer group 1), (category 2, DB2, consumer group 2), (category 2, DB2, consumer group 3), and (category 2, DB4, consumer group 3).

[0102] During phase 2, the storage server would determine which of DB1 or DB2 is furthest behind in achieving its I/O quota. DB4 is not considered, because the policies indicate that DB4 is only considered if no deferred I/Os target DB1, DB2 or DB3. Assuming that DB2 is the furthest behind in achieving its I/O quota, the selection groups that survive phase 2 would be: (category 2, DB2, consumer group 2), (category 2, DB2, consumer group 3).

[0103] During phase 3, the storage server would select consumer group 2, because the policy indicates that consumer group 3 is only considered if no pending I/O requests correspond to consumer groups 1 and 2. Consequently, after phase 3, the only remaining

selection group is (category 2, DB2, consumer group 2). Therefore, the storage server would select the I/O request at the head of the waiting queue that corresponds to the selection group (category 2, DB2, consumer group 2).

[0104] In the example given above, all three phases had to be applied before a single non-empty selection group remained. However, depending on which selection groups are non-empty, it may be that fewer than all of the phases need to be applied. For example, consider the situation where the non-empty selection groups include: (category 1, DB5, consumer group 3), (category 2, DB1, consumer group 1), (category 2, DB2, consumer group 2), (category 2, DB2, consumer group 3), (category 3, DB2, consumer group 1), and (category 2, DB4, consumer group 3). In this situation, after phase 1 of selection policy 300 the only surviving selection group would be (category 1, DB5, consumer group 3), because this is the only non-empty selection group with pending I/O requests associated with category 1. Consequently, after phase 1, the storage server would simply select the I/O request at the head of the queue associated with (category 1, DB5, consumer group 3).

LOGICAL WAITING QUEUES

[0105] The queues into which the storage server places pending I/O requests need not be actual distinct data structures. Specifically, storage server may simply track (1) the logical characteristics of each pending I/O request, and (2) the time the I/O request was received by the storage system. Pending I/O requests that have the same logical characteristics belong to the same selection group, and are therefore treated as belonging to the same logical queue, even though no separate queue structure is used for the selection group.

[0106] In an embodiment that uses logical waiting queues rather than separate queue structures, the time the I/O requests were received may be used to indicate the order of each logical queue. Thus, for each selection group, the pending I/O request with the earliest arrival time being treated as residing at the head of the logical queue of the selection group. Alternatively, all pending I/O requests may be stored in a single waiting queue. For each selection group, the pending I/O request that is closest to the head of the single waiting queue is treated as being at the head of the logical waiting queue for the selection group.

HARDWARE OVERVIEW

[0107] According to one embodiment, the techniques described herein are implemented by one or more special-purpose computing devices. The special-purpose computing devices may be hard-wired to perform the techniques, or may include digital electronic devices such as one or more application-specific integrated circuits (ASICs) or field programmable gate arrays (FPGAs) that are persistently programmed to perform the techniques, or may include one or more general purpose hardware processors programmed to perform the techniques pursuant to program instructions in firmware, memory, other storage, or a combination. Such special-purpose computing devices may also combine custom hard-wired logic, ASICs, or FPGAs with custom programming to accomplish the techniques. The special-purpose computing devices may be desktop computer systems, portable computer systems, handheld devices, networking devices or any other device that incorporates hard-wired and/or program logic to implement the techniques.

[0108] For example, FIG. 6 is a block diagram that illustrates a computer system 600 upon which an embodiment of the invention may be implemented. Computer system 600 includes a bus 602 or other communication mechanism for communicating information, and a hardware processor 604 coupled with bus 602 for processing information. Hardware processor 604 may be, for example, a general purpose microprocessor.

[0109] Computer system 600 also includes a main memory 606, such as a random access memory (RAM) or other dynamic storage device, coupled to bus 602 for storing information and instructions to be executed by processor 604. Main memory 606 also may be used for storing temporary variables or other intermediate information during execution of instructions to be executed by processor 604. Such instructions, when stored in storage media accessible to processor 604, render computer system 600 into a special-purpose machine that is customized to perform the operations specified in the instructions.

[0110] Computer system 600 further includes a read only memory (ROM) 608 or other static storage device coupled to bus 602 for storing static information and instructions for processor 604. A storage device 610, such as a magnetic disk or optical disk, is provided and coupled to bus 602 for storing information and instructions.

[0111] Computer system 600 may be coupled via bus 602 to a display 612, such as a cathode ray tube (CRT), for displaying information to a computer user. An input device 614, including alphanumeric and other keys, is coupled to bus 602 for communicating information

and command selections to processor 604. Another type of user input device is cursor control 616, such as a mouse, a trackball, or cursor direction keys for communicating direction information and command selections to processor 604 and for controlling cursor movement on display 612. 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.

[0112] Computer system 600 may implement the techniques described herein using customized hard-wired logic, one or more ASICs or FPGAs, firmware and/or program logic which in combination with the computer system causes or programs computer system 600 to be a special-purpose machine. According to one embodiment, the techniques herein are performed by computer system 600 in response to processor 604 executing one or more sequences of one or more instructions contained in main memory 606. Such instructions may be read into main memory 606 from another storage medium, such as storage device 610. Execution of the sequences of instructions contained in main memory 606 causes processor 604 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.

[0113] The term "storage media" as used herein refers to any media that store data and/or instructions that cause a machine to operation in a specific fashion. Such storage media may comprise non-volatile media and/or volatile media. Non-volatile media includes, for example, optical or magnetic disks, such as storage device 610. Volatile media includes dynamic memory, such as main memory 606. Common forms of storage media include, for example, a floppy disk, a flexible disk, hard disk, solid state drive, magnetic tape, or any other magnetic data storage medium, a CD-ROM, any other optical data storage medium, any physical medium with patterns of holes, a RAM, a PROM, and EPROM, a FLASH-EPROM, NVRAM, any other memory chip or cartridge.

[0114] Storage media is distinct from but may be used in conjunction with transmission media. Transmission media participates in transferring information between storage media. For example, transmission media includes coaxial cables, copper wire and fiber optics, including the wires that comprise bus 602. Transmission media can also take the form of acoustic or light waves, such as those generated during radio-wave and infra-red data communications.

[0115] Various forms of media may be involved in carrying one or more sequences of one or more instructions to processor 604 for execution. For example, the instructions may initially be carried on a magnetic disk or solid state drive 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 600 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 602. Bus 602 carries the data to main memory 606, from which processor 604 retrieves and executes the instructions. The instructions received by main memory 606 may optionally be stored on storage device 610 either before or after execution by processor 604.

[0116] Computer system 600 also includes a communication interface 618 coupled to bus 602. Communication interface 618 provides a two-way data communication coupling to a network link 620 that is connected to a local network 622. For example, communication interface 618 may be an integrated services digital network (ISDN) card, cable modem, satellite modem, or a modem to provide a data communication connection to a corresponding type of telephone line. As another example, communication interface 618 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 618 sends and receives electrical, electromagnetic or optical signals that carry digital data streams representing various types of information.

[0117] Network link 620 typically provides data communication through one or more networks to other data devices. For example, network link 620 may provide a connection through local network 622 to a host computer 624 or to data equipment operated by an Internet Service Provider (ISP) 626. ISP 626 in turn provides data communication services through the world wide packet data communication network now commonly referred to as the "Internet" 628. Local network 622 and Internet 628 both use electrical, electromagnetic or optical signals that carry digital data streams. The signals through the various networks and the signals on network link 620 and through communication interface 618, which carry the digital data to and from computer system 600, are example forms of transmission media.

[0118] Computer system 600 can send messages and receive data, including program code, through the network(s), network link 620 and communication interface 618. In the

Internet example, a server 630 might transmit a requested code for an application program through Internet 628, ISP 626, local network 622 and communication interface 618.

[0119] The received code may be executed by processor 604 as it is received, and/or stored in storage device 610, or other non-volatile storage for later execution.

[0120] 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 comprising:

while a storage system is in a saturated state, queuing, within the storage system, a plurality of I/O requests received by the storage system; and

when the storage system is ready to process a pending I/O request that has been queued in the storage system, a storage server within the storage system selecting a particular I/O request to process based, at least in part, on one or more logical characteristics associated with particular I/O requests;

processing the particular I/O request in response to the storage server selecting the particular I/O request;

wherein the method is performed by one or more special-purpose computing devices.

2. The method of Claim 1 further comprising:

the storage system receiving one or more selection policies; and

the storage server using policies indicated in each of the one or more selection policies to select which of the plurality of I/O requests to select for processing.

3. The method of Claim 2 wherein storage server applies the policies in a plurality of phases.

4. The method of Claim 3 wherein:

the storage system stores data for a plurality of databases; and

a particular phase that precedes all other phases of the plurality of phases includes policies for selecting between I/O requests based on the databases targeted by the I/O requests.

5. The method of Claim 3 wherein:

the storage system stores data for a plurality of databases; and

a particular phase that follows at least one other phase of the plurality of phases includes policies for selecting between I/O requests based on the databases targeted by the I/O requests.

4. The method of Claim 3 wherein:
I/O requests that have similar logical characteristics belong to a selection group;
the plurality of I/O requests correspond to a plurality of selection groups;
the policies applied by the storage server in a later phase of the plurality of phases are based, at least in part on which selection groups survive one or more earlier phases.

5. The method of Claim 2 wherein the policies include one or more ratio-based policies.

6. The method of Claim 5 wherein the policies further include one or more priority-based policies.

7. The method of Claim 2 wherein the policies include one or more hybrid policies.

8. The method of Claim 3 wherein at least one of the plurality of phases is associated with a hybrid policy.

9. The method of Claim 1 wherein:
a plurality of consumer groups are specified in a logical-characteristic-value-to-consumer-group mapping; and
the one or more logical characteristics include the consumer group to which the particular I/O request belongs.

10. The method of Claim 9 further comprising:
the storage system receiving the logical-characteristic-value-to-consumer-group mapping; and
the storage server determining the consumer group to which the particular I/O request belongs based on request metadata of the particular I/O request and the logical-characteristic-value-to-consumer-group mapping.

11. The method of Claim 9 further comprising:
the storage system receiving request metadata for said particular I/O request;
wherein the request metadata includes a consumer group identifier; and

the storage server storage server selecting the particular I/O request based, at least in part, on the value of the consumer group identifier.

12. The method of Claim 1 further comprising receiving at the storage server, from an I/O requestor that is submitting an I/O request, values for the one or more logical characteristics associated with the I/O request.

13. The method of Claim 1 further comprising:

receiving at the storage system, from a first I/O requestor that is submitting a first I/O request, first values for the one or more logical characteristics associated with the first I/O request; and

receiving at the storage system, from a second I/O requestor that is submitting a second I/O request, second values for the one or more logical characteristics associated with the second I/O request; and

wherein the storage server determines which of the first I/O request and the second I/O request to process first based, at least in part, on the first values and the second values.

14. The method of Claim 13 wherein the first I/O requestor is a first database server and the second I/O requestor is a second database server.

15. The method of Claim 1 wherein the one or more logical characteristics include which database, of a plurality of databases stored within the storage system, is targeted by the I/O requests.

16. The method of Claim 1 wherein the one or more logical characteristics include which users are responsible for requesting the operations that produced the I/O requests.

17. The method of Claim 1 wherein the one or more logical characteristics include what type of workload is represented by the I/O requests.

18. The method of Claim 2 further comprising:

the storage system determining an load setting based, at least in part, on information from the one or more selection policies; and

the storage system determining when to process I/O requests based on the load setting.

19. The method of Claim 1 further comprising:

the storage system determining logical characteristics associated with I/O requests received by the storage system based on request metadata that accompanies the I/O requests;

the storage system determining an load setting based, at least in part, on the logical characteristics associated with the I/O requests received by the storage system; and

the storage system determining when to process I/O requests based on the load setting.

20. The method of Claim 19 wherein the steps of determining logical characteristics and determining an load setting are repeated periodically to dynamically vary the load based on how the storage system is being used.

21. A method comprising:

submitting to a storage system, by an I/O requestor, an I/O request; and

submitting to the storage system, by the I/O requestor, values for one or more logical characteristics of the I/O request to enable a storage server within the storage system to prioritize I/O requests based on the one or more logical characteristics;

wherein the method is performed by one or more special-purpose computing devices.

22. The method of Claim 21 wherein the I/O requestor is a database server, and the method further comprises the database server sending to the storage system a selection policy that indicates policies for the storage server to apply when selecting which I/O request to process.

23. The method of Claim 20 wherein the storage system stores data for a plurality of databases, and the policies include database-selection policies.

24. A method comprising:

a storage system automatically selecting a scheduling policy based on workloads that are associated with I/O requests that are received by the storage system; and

the storage system determining when to issue pending I/O requests based on the scheduling policy.

25. The method of Claim 24 further comprising:

the storage system receiving metadata that indicates one or more logical characteristics of the I/O requests that are received by the storage system; and

the storage system determining workloads that are associated with I/O requests based on the metadata.

26. The method of Claim 24 further comprising:

the storage system receiving one or more selection policies for selecting which pending I/O request to issue; and

the storage system determining workloads that are associated with I/O requests based on the one or more selection policies.

27. The method of Claim 24 further comprising changing the scheduling policy from a first scheduling policy to a second scheduling policy in response to an event, wherein the first and second scheduling policies specify different target loads.

28. The method of Claim 27 wherein the event is a detected change in the relative frequency at which I/O requests are associated with particular workloads.

29. A computer-readable storage medium comprising instructions which, when executed by one or more processors, cause:

while a storage system is in a saturated state, queuing, within the storage system, a plurality of I/O requests received by the storage system; and

when the storage system is ready to process a pending I/O request that has been queued in the storage system, a storage server within the storage system selecting a particular I/O request to process based, at least in part, on one or more logical characteristics associated with particular I/O requests;

processing the particular I/O request in response to the storage server selecting the particular I/O request;

wherein the computer-readable storage medium is performed by one or more special-purpose computing devices.

30. The computer-readable storage medium of Claim 29 further comprising instructions for:

the storage system receiving one or more selection policies; and

the storage server using policies indicated in each of the one or more selection policies to select which of the plurality of I/O requests to select for processing.

31. The computer-readable storage medium of Claim 30 wherein storage server applies the policies in a plurality of phases.

32. The computer-readable storage medium of Claim 31 wherein:

the storage system stores data for a plurality of databases; and

a particular phase that precedes all other phases of the plurality of phases includes policies for selecting between I/O requests based on the databases targeted by the I/O requests.

33. The computer-readable storage medium of Claim 31 wherein:

the storage system stores data for a plurality of databases; and

a particular phase that follows at least one other phase of the plurality of phases includes policies for selecting between I/O requests based on the databases targeted by the I/O requests.

32. The computer-readable storage medium of Claim 31 wherein:

I/O requests that have similar logical characteristics belong to a selection group;

the plurality of I/O requests correspond to a plurality of selection groups;

the policies applied by the storage server in a later phase of the plurality of phases are based, at least in part on which selection groups survive one or more earlier phases.

33. The computer-readable storage medium of Claim 30 wherein the policies include one or more ratio-based policies.

34. The computer-readable storage medium of Claim 33 wherein the policies further include one or more priority-based policies.

35. The computer-readable storage medium of Claim 30 wherein the policies include one or more hybrid policies.

36. The computer-readable storage medium of Claim 31 wherein at least one of the plurality of phases is associated with a hybrid policy.

37. The computer-readable storage medium of Claim 29 wherein:

a plurality of consumer groups are specified in a logical-characteristic-value-to-consumer-group mapping; and

the one or more logical characteristics include the consumer group to which the particular I/O request belongs.

38. The computer-readable storage medium of Claim 37 further comprising instructions for:

the storage system receiving the logical-characteristic-value-to-consumer-group mapping; and

the storage server determining the consumer group to which the particular I/O request belongs based on request metadata of the particular I/O request and the logical-characteristic-value-to-consumer-group mapping.

39. The computer-readable storage medium of Claim 37 further comprising instructions for:

the storage system receiving request metadata for said particular I/O request;

wherein the request metadata includes a consumer group identifier; and

the storage server storage server selecting the particular I/O request based, at least in part, on the value of the consumer group identifier.

40. The computer-readable storage medium of Claim 29 further comprising instructions for receiving at the storage server, from an I/O requestor that is submitting an I/O request, values for the one or more logical characteristics associated with the I/O request.

41. The computer-readable storage medium of Claim 29 further comprising instructions for:

receiving at the storage system, from a first I/O requestor that is submitting a first I/O request, first values for the one or more logical characteristics associated with the first I/O request; and

receiving at the storage system, from a second I/O requestor that is submitting a second I/O request, second values for the one or more logical characteristics associated with the second I/O request; and

wherein the storage server determines which of the first I/O request and the second I/O request to process first based, at least in part, on the first values and the second values.

42. The computer-readable storage medium of Claim 41 wherein the first I/O requestor is a first database server and the second I/O requestor is a second database server.

43. The computer-readable storage medium of Claim 29 wherein the one or more logical characteristics include which database, of a plurality of databases stored within the storage system, is targeted by the I/O requests.

44. The computer-readable storage medium of Claim 29 wherein the one or more logical characteristics include which users are responsible for requesting the operations that produced the I/O requests.

45. The computer-readable storage medium of Claim 29 wherein the one or more logical characteristics include what type of workload is represented by the I/O requests.

46. The computer-readable storage medium of Claim 30 further comprising instructions for:

the storage system determining an load setting based, at least in part, on information from the one or more selection policies; and

the storage system determining when to process I/O requests based on the load setting.

47. The computer-readable storage medium of Claim 29 further comprising instructions for:

the storage system determining logical characteristics associated with I/O requests received by the storage system based on request metadata that accompanies the I/O requests;

the storage system determining an load setting based, at least in part, on the logical characteristics associated with the I/O requests received by the storage system; and

the storage system determining when to process I/O requests based on the load setting.

48. The computer-readable storage medium of Claim 47 wherein the steps of determining logical characteristics and determining an load setting are repeated periodically to dynamically vary the load based on how the storage system is being used.

49. A computer-readable storage medium comprising instructions which, when executed by one or more processors, cause:

submitting to a storage system, by an I/O requestor, an I/O request; and

submitting to the storage system, by the I/O requestor, values for one or more logical characteristics of the I/O request to enable a storage server within the storage system to prioritize I/O requests based on the one or more logical characteristics;

wherein the computer-readable storage medium is performed by one or more special-purpose computing devices.

50. The computer-readable storage medium of Claim 49 wherein the I/O requestor is a database server, and the computer-readable storage medium further comprises the database server sending to the storage system a selection policy that indicates policies for the storage server to apply when selecting which I/O request to process.

51. The computer-readable storage medium of Claim 48 wherein the storage system stores data for a plurality of databases, and the policies include database-selection policies.

52. A computer-readable storage medium comprising instructions which, when executed by one or more processors, cause:

a storage system automatically selecting a scheduling policy based on workloads that are associated with I/O requests that are received by the storage system; and

the storage system determining when to issue pending I/O requests based on the scheduling policy.

53. The computer-readable storage medium of Claim 52 further comprising instructions for:

the storage system receiving metadata that indicates one or more logical characteristics of the I/O requests that are received by the storage system; and

the storage system determining workloads that are associated with I/O requests based on the metadata.

54. The computer-readable storage medium of Claim 52 further comprising instructions for:

the storage system receiving one or more selection policies for selecting which pending I/O request to issue; and

the storage system determining workloads that are associated with I/O requests based on the one or more selection policies.

55. The computer-readable storage medium of Claim 52 further comprising instructions for changing the scheduling policy from a first scheduling policy to a second scheduling policy in response to an event, wherein the first and second scheduling policies specify different target loads.

56. The computer-readable storage medium of Claim 55 wherein the event is a detected change in the relative frequency at which I/O requests are associated with particular workloads.

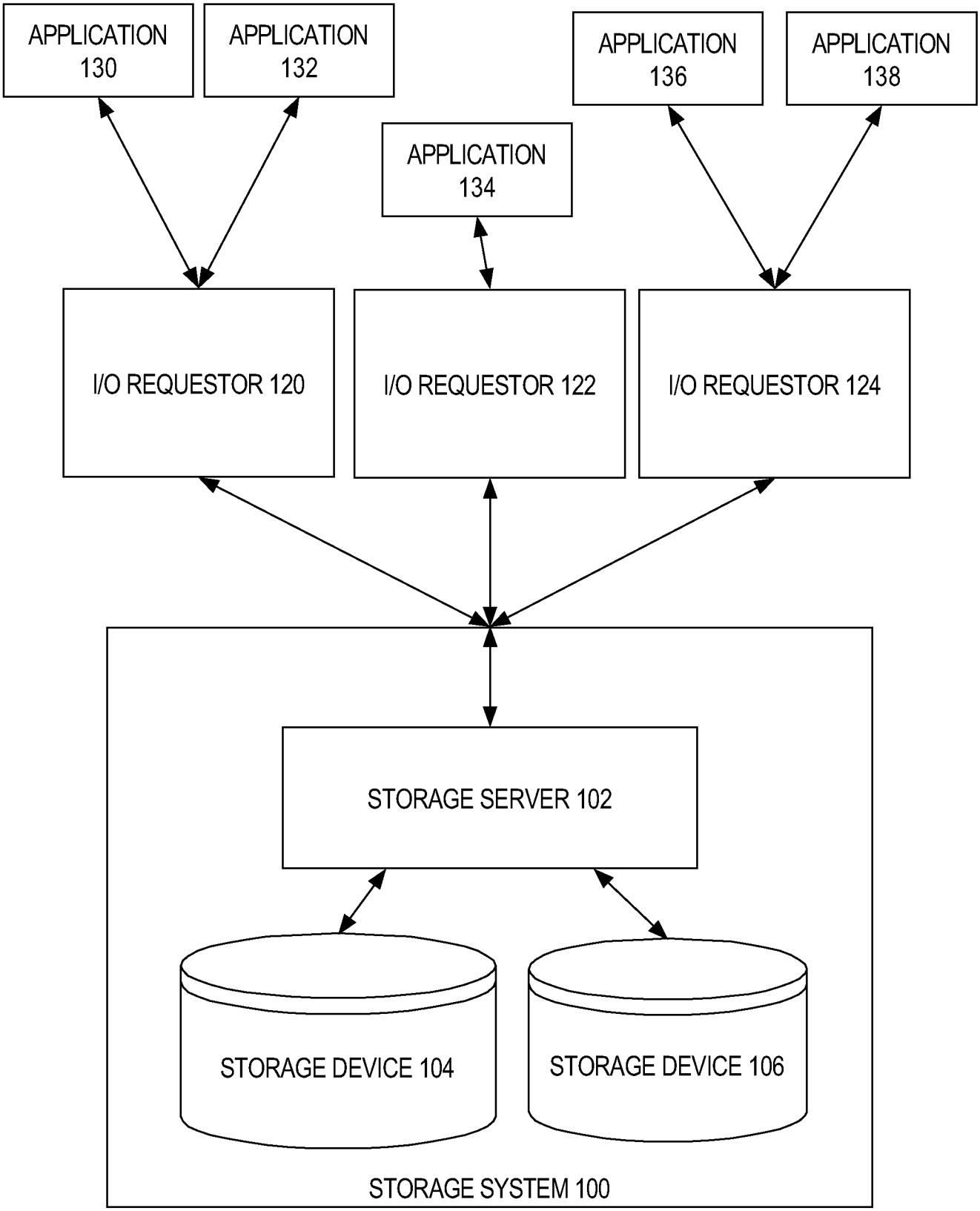
APPLICATION 130
APPLICATION 132
APPLICATION 134
APPLICATION 136
APPLICATION 138
I/O REQUESTOR 120
I/O REQUESTOR 122
I/O REQUESTOR 124
STORAGE SERVER 102
STORAGE DEVICE 104
STORAGE DEVICE 106
STORAGE SYSTEM 100

FIG. 1

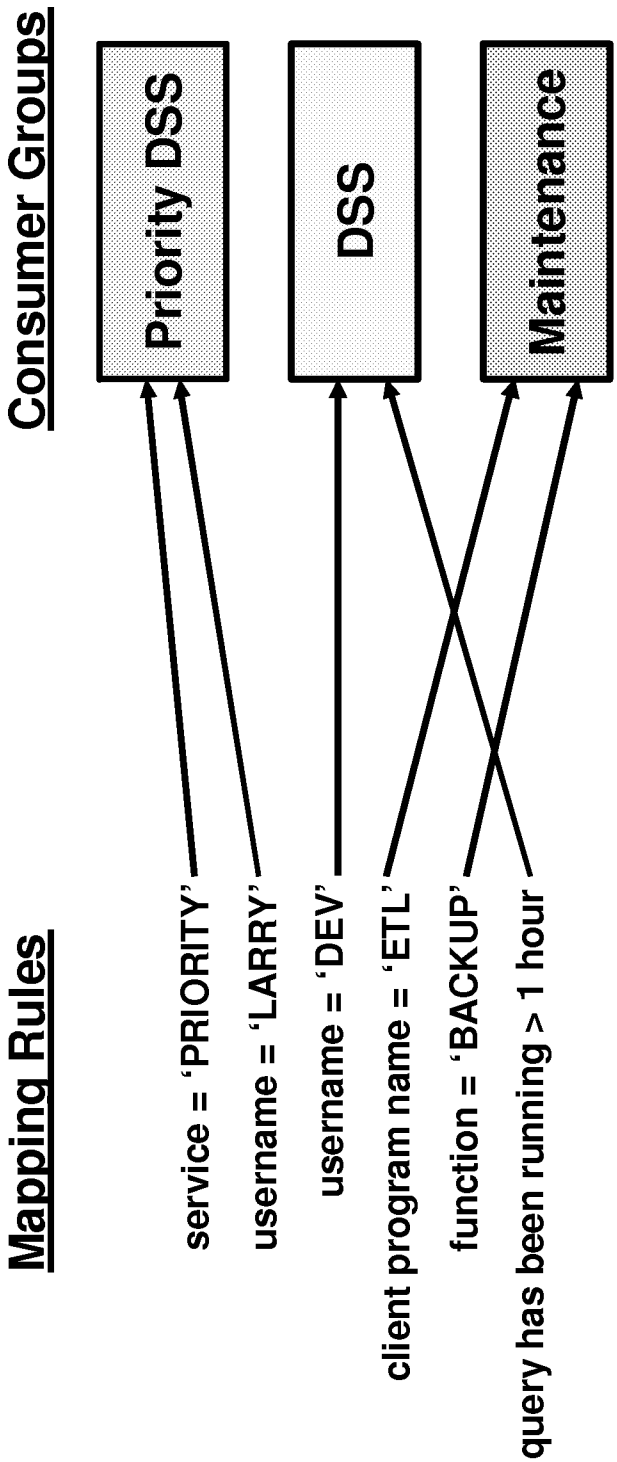
Consumer Groups
Priority DSS
DSS
Maintenance
Mapping Rules
service = 'PRIORITY'
username = 'LARRY'
username = 'DEV'
client program name = 'ETL'
function = 'BACKUP'
query has been running > 1 hour

FIG. 2

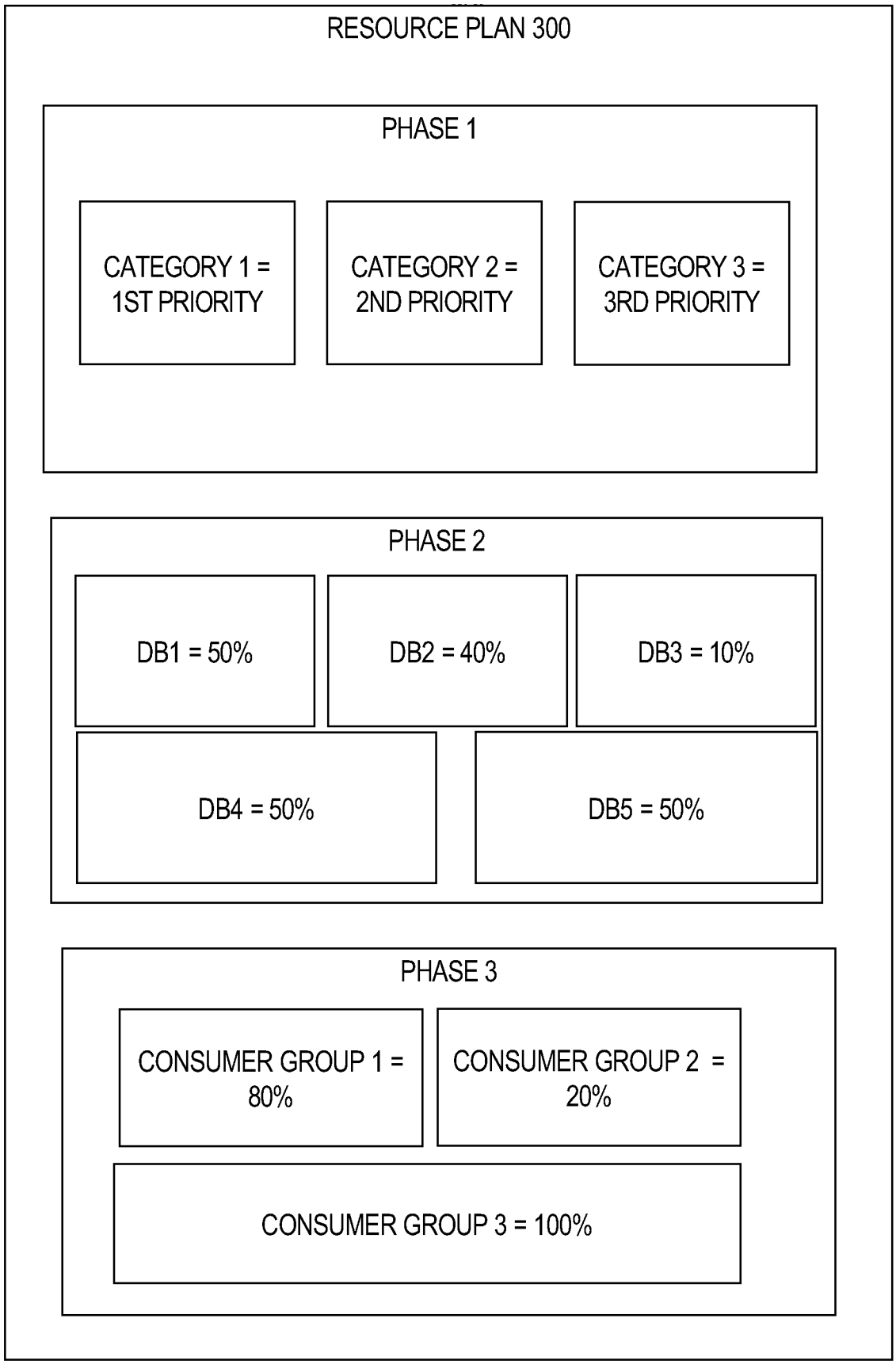
RESOURCE PLAN 300
PHASE 1
CATEGORY 1 = 1ST PRIORITY
CATEGORY 2 = 2ND PRIORITY
CATEGORY 3 = 3RD PRIORITY
PHASE 2
DB1 = 50%
DB2 = 40%
DB3 = 10%
DB4 = 50%
DB5 = 50%
PHASE 3
CONSUMER GROUP 1 = 80%
CONSUMER GROUP 2 = 20%
CONSUMER GROUP 3 = 100%

FIG. 3

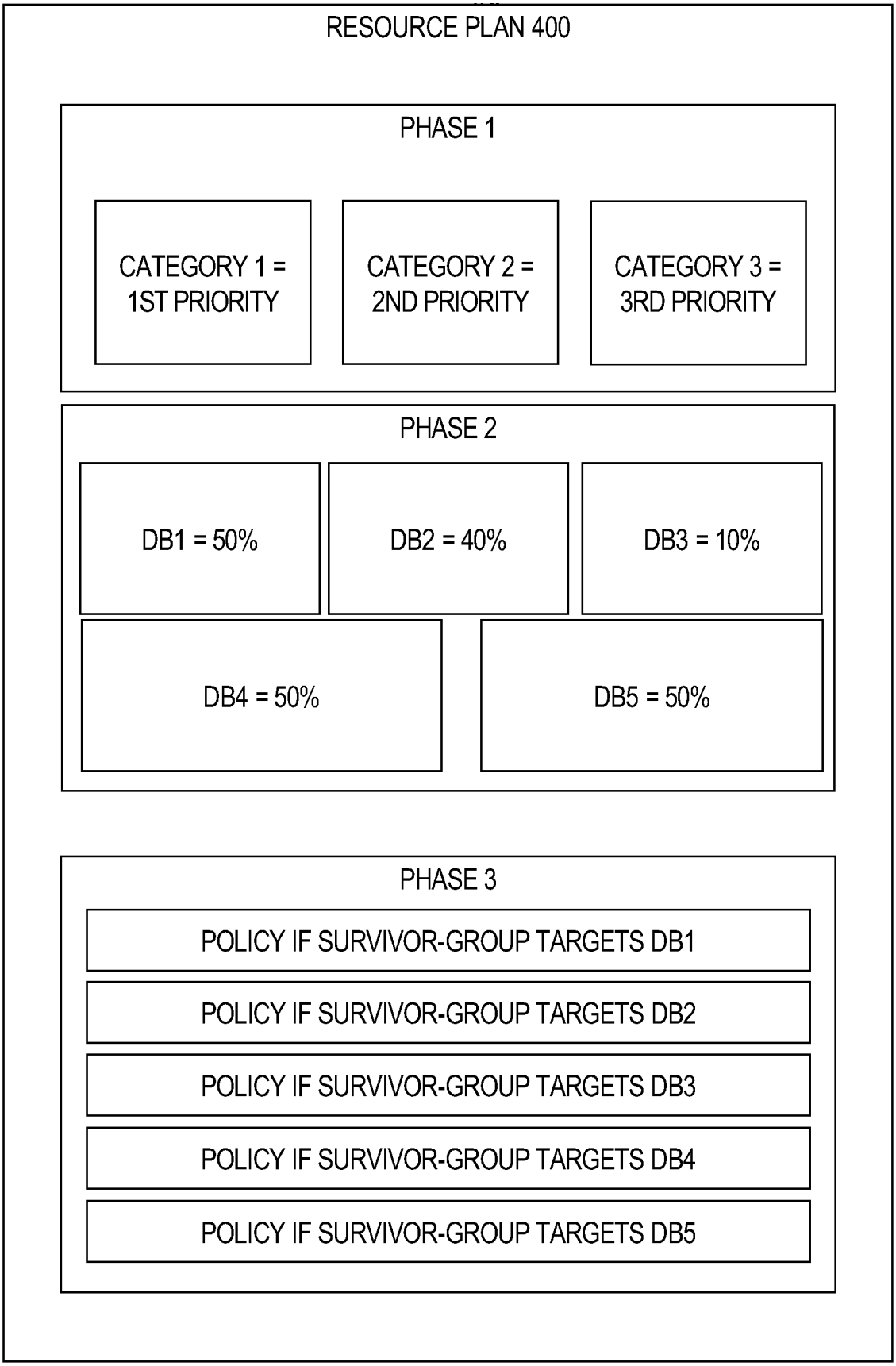
RESOURCE PLAN 400
PHASE 1
CATEGORY 1 = 1ST PRIORITY
CATEGORY 2 = 2ND PRIORITY
CATEGORY 3 = 3RD PRIORITY
PHASE 2
DB1 = 50%
DB2 = 40%
DB3 = 10%
DB4 = 50%
DB5 = 50%
PHASE 3
POLICY IF SURVIVOR-GROUP TARGETS DB1
POLICY IF SURVIVOR-GROUP TARGETS DB2
POLICY IF SURVIVOR-GROUP TARGETS DB3
POLICY IF SURVIVOR-GROUP TARGETS DB4
POLICY IF SURVIVOR-GROUP TARGETS DB5

FIG. 4

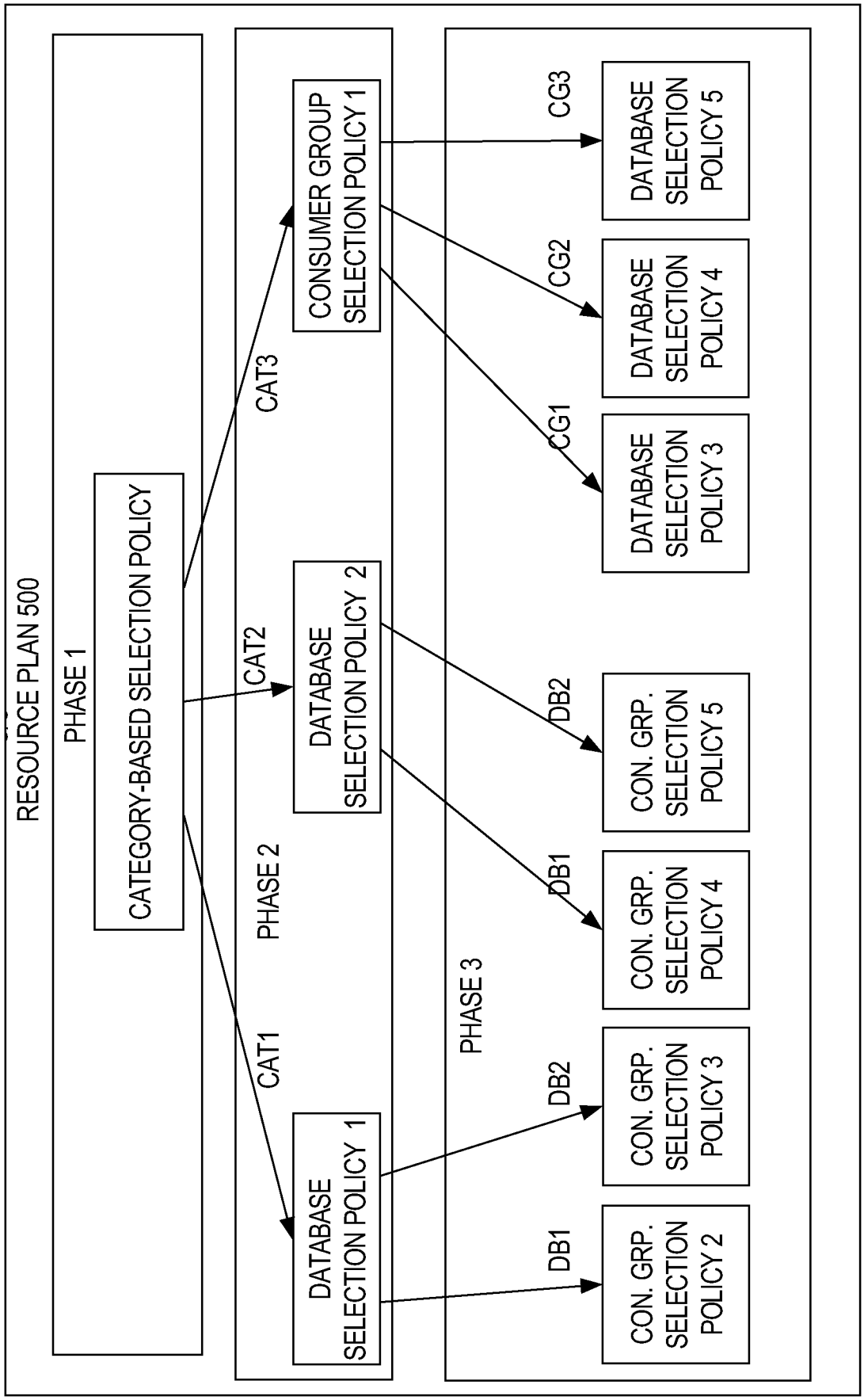
RESOURCE PLAN 500
PHASE 1
CATEGORY-BASED SELECTION POLICY
PHASE 2
CAT1
CAT2
CAT3
DATABASE SELECTION POLICY 1
DATABASE SELECTION POLICY 2
CONSUMER GROUP SELECTION POLICY 1
PHASE 3
DB1
DB2
DB1
DB2
CON. GRP. SELECTION POLICY 2
CON. GRP. SELECTION POLICY 3
CON. GRP. SELECTION POLICY 4
CON. GRP. SELECTION POLICY 5
CG1
CG2
CG3
DATABASE SELECTION POLICY 3
DATABASE SELECTION POLICY 4
DATABASE SELECTION POLICY 5

FIG. 5

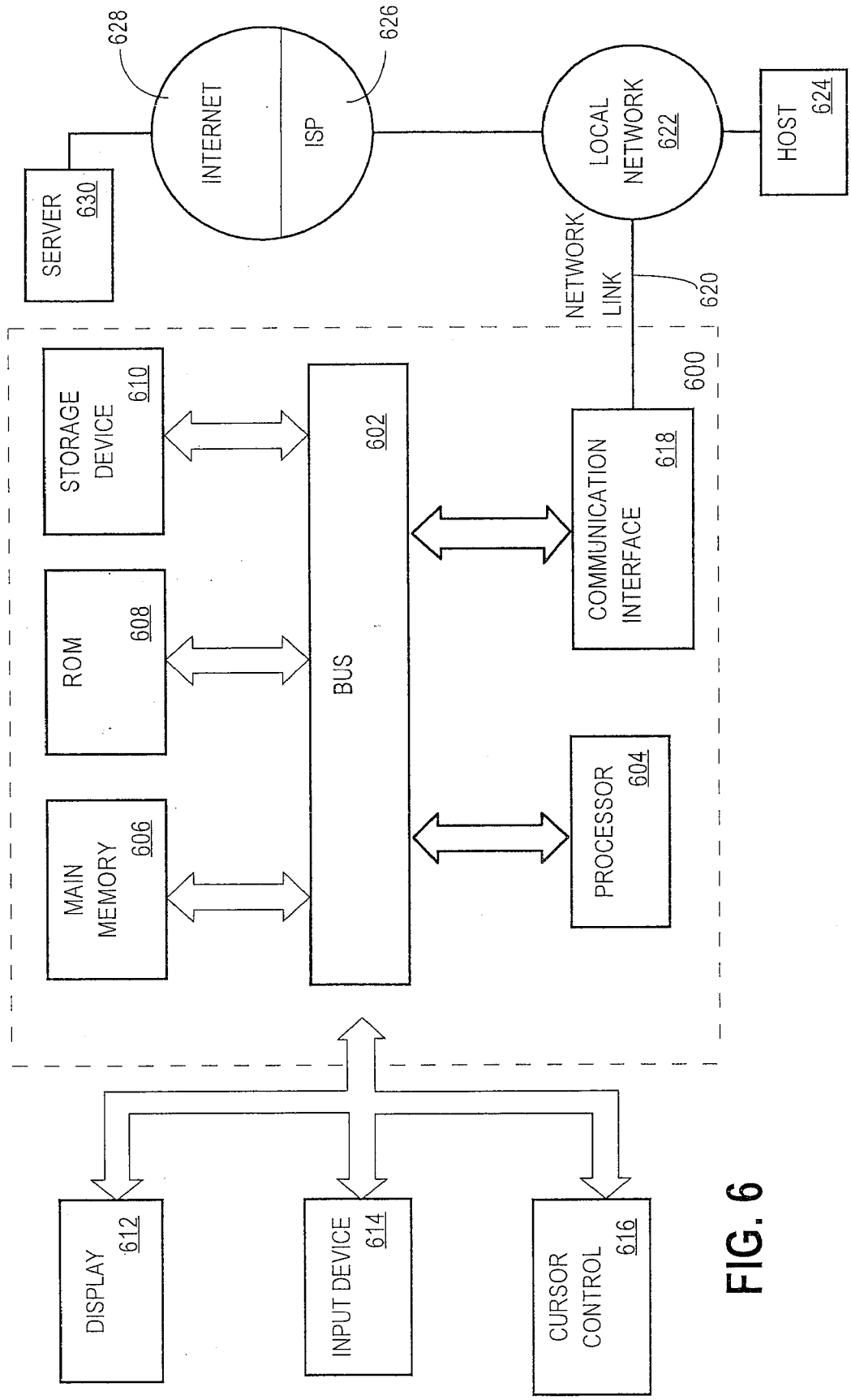
628
626
INTERNET
ISP
SERVER 630
LOCAL NETWORK 622
HOST 624
NETWORK LINK 620
600
STORAGE DEVICE 610
602
BUS
COMMUNICATION INTERFACE 618
ROM 608
MAIN MEMORY 606
PROCESSOR 604
DISPLAY 612
INPUT DEVICE 614
CURSOR CONTROL 616

FIG. 6

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2009/057590

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F17/30

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, INSPEC, COMPENDEX

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 6 044 367 A (WOLFF JAMES J [US]) 28 March 2000 (2000-03-28) abstract column 2, line 32 - line 35 column 6, line 5 - line 30 column 6, line 39 - line 41 column 9, line 7 - line 12 column 9, line 42 - line 53 column 10, line 15 - line 18 column 11, line 1 - line 24 column 15, line 61 - column 17, line 38 figures 1B,3A-C ----- -/--	1-56

[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
5 February 2010	11/02/2010

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, Fax: (+31-70) 340-3016	Wohner, Wolfgang

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

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2009/057590

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 7 237 027 B1 (RACCAH DAVID [US] ET AL) 26 June 2007 (2007-06-26) the whole document abstract column 4, line 30 - line 54 column 7, line 23 - column 10, line 52 figures 4,5	1-56
A	US 2008/177803 A1 (FINEBERG SAM [US] ET AL) 24 July 2008 (2008-07-24) the whole document	1-56
A	US 2005/120025 A1 (RODRIGUEZ ANDRES [US] ET AL RODRIGUEZ ANDRES [US] ET AL) 2 June 2005 (2005-06-02) the whole document	1-56

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2009/057590

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 6044367 A	28-03-2000	US 6185601 B1	06-02-2001
US 7237027 B1	26-06-2007	NONE	
US 2008177803 A1	24-07-2008	NONE	
US 2005120025 A1	02-06-2005	US 2007094316 A1	26-04-2007