Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2522467 C 2010/12/21

(11)(21) 2 522 467

(12) BREVET CANADIEN
CANADIAN PATENT

(13) C

(86) Date de dépôt PCT/PCT Filing Date: 2003/10/27
(87) Date publication PCT/PCT Publication Date: 2004/06/10
(45) Date de délivrance/Issue Date: 2010/12/21
(85) Entrée phase nationale/National Entry: 2005/04/19
(86) N° demande PCT/PCT Application No.: GB 2003/004650
(87) N° publication PCT/PCT Publication No.: 2004/049157
(30) Priorité/Priority: 2002/11/27 (US10/306,301)

(51) Cl.Int./Int.Cl. *G06F 9/50* (2006.01),
G06F 1/32 (2006.01)

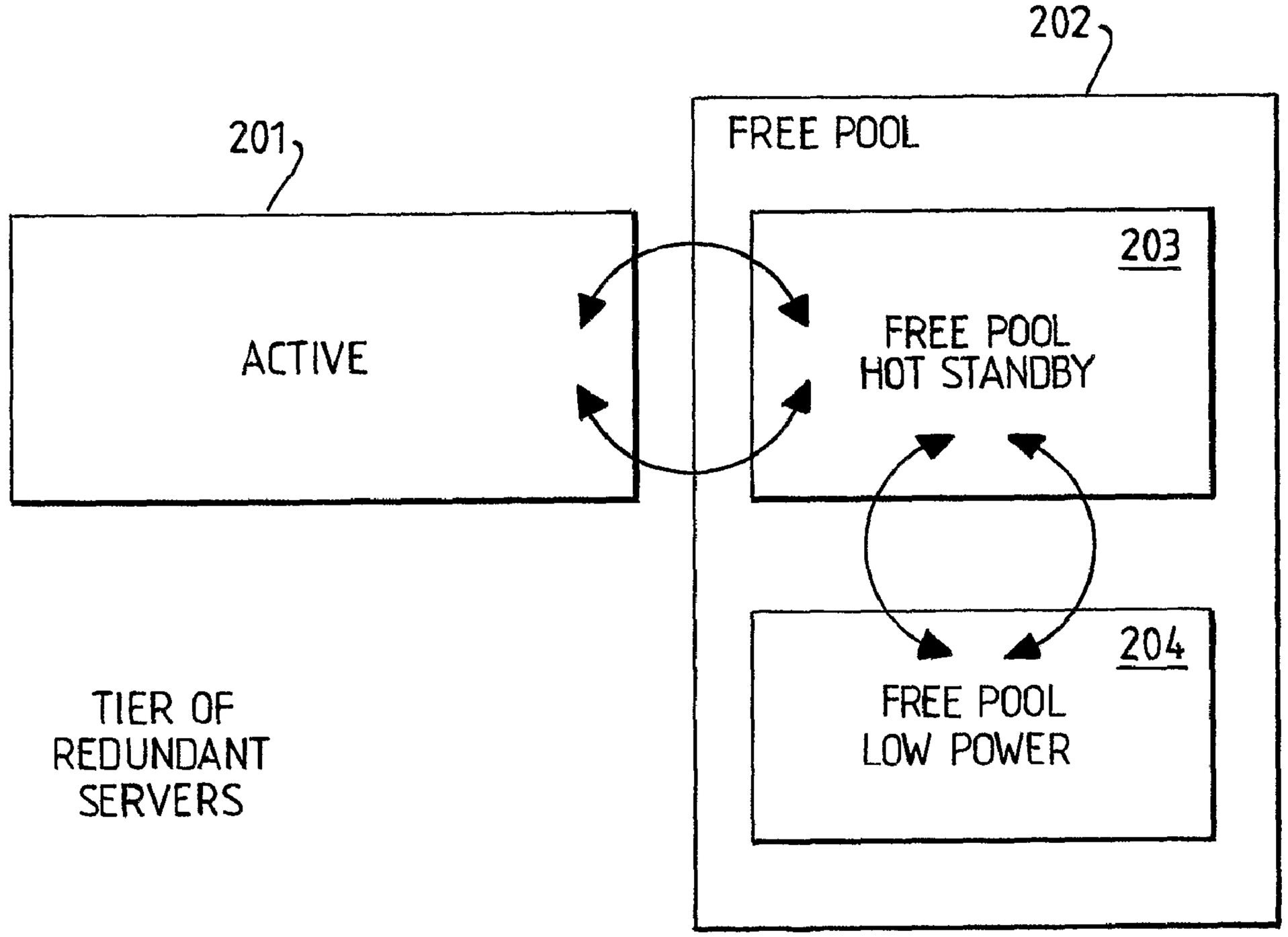
(72) Inventeurs/Inventors:
BIRAN, OFER, IL;
HARPER, RICHARD EDWIN, US;
KRISHNAKUMAR, SRIRAMA MANDYAM, US;
MACKENZIE, BRUCE KENNETH, US;
PRUETT, GREGORY BRIAN, US;
YASSOUR, BEN-AMI, IL

(73) Propriétaire/Owner:
INTERNATIONAL BUSINESS MACHINES CORPORATION, US

(74) Agent: WANG, PETER

(54) Titre : POLITIQUES DE GESTION DE PUISSANCE AUTOMATISEE BASEES SUR DES CARACTERISTIQUES DE REDONDANCE PROPRES A UNE APPLICATION

(54) Title: AUTOMATED POWER CONTROL POLICIES BASED ON APPLICATION-SPECIFIC REDUNDANCY CHARACTERISTICS



(57) Abrégé/Abstract:
Power and redundancy management policies are applied individually to the tiers of redundant servers of an application service such that power is reduced while maintaining a high level of system availability. Servers which are determined to be relatively

CA 2522467 C 2010/12/21

(11)(21) 2 522 467

(13) C

(57) Abrégé(suite)/Abstract(continued):
inactive are moved to a free pool. Certain servers of the free pool are maintained in a hot standby state, while others are powered-off or set to operate in a low power mode. During times of high load, the servers in the hot standby state can be provisioned quickly into the application service.

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
10 June 2004 (10.06.2004)

PCT

(10) International Publication Number
WO 2004/049157 A3

(51) International Patent Classification[7]: G06F 9/50, 1/32

(21) International Application Number:
PCT/GB2003/004650

(22) International Filing Date: 27 October 2003 (27.10.2003)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
10/306,301 27 November 2002 (27.11.2002) US

(71) Applicant: INTERNATIONAL BUSINESS MACHINES CORPORATION [US/US]; New Orchard Road, Armonk, NY 10504 (US).

(71) Applicant *(for MG only)*: IBM UNITED KINGDOM LIMITED [GB/GB]; P.O. Box 41, North Harbour, Portsmouth, Hampshire PO6 3AU (GB).

(72) Inventors: **BIRAN, Ofer**; Derech Hayam 151A, 34748 Haifa (IL). **HARPER, Richard, Edwin**; 105 Winston Ridge Drive, Chapel Hill, NC 27516 (US). **KRISHNAKUMAR, Srirama, Mandyam**; 2 Montana Place, White Plains, NY 10607 (US). **MACKENZIE, Bruce, Kenneth**; 3410 Tarlton Lane, Austin, TX 78746 (US). **PRUETT, Gregory, Brian**; 6113 Carlyle Drive, Raleigh, NC 27614 (US). **YASSOUR, Ben-Ami**; Deisraeli 20, 34334 Haifa (IL).

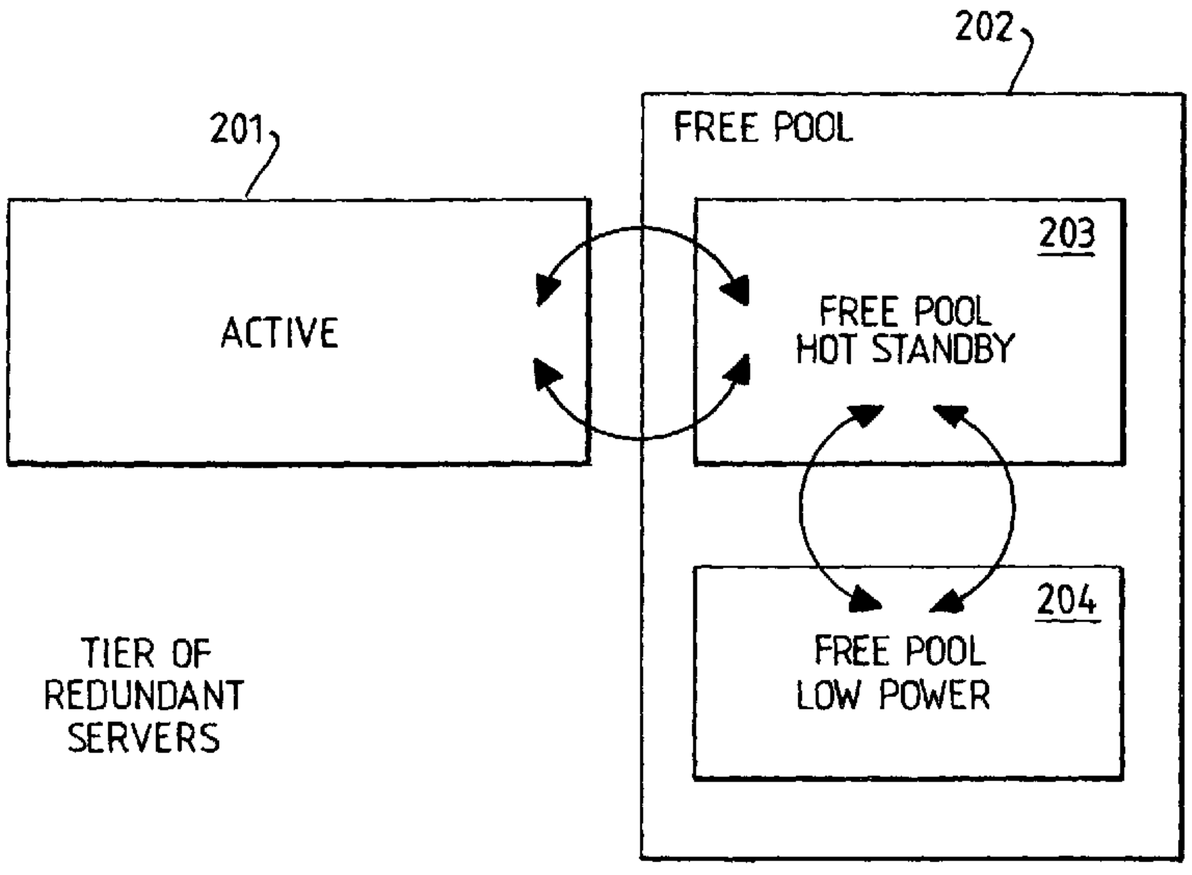
(74) Agent: **LITHERLAND, David, Peter**; IBM United Kingdom Limited, Intellectual Property Law, Hursley Park, Winchester, Hampshire SO21 2JN (GB).

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

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IT, LU, MC, NL, PT, RO,

[Continued on next page]

(54) Title: AUTOMATED POWER CONTROL POLICIES FOR DISTRIBUTED SERVER POOLS BASED ON APPLICATION-SPECIFIC COMPUTATIONAL REQUIREMENTS



(57) Abstract: Power and redundancy management policies are applied individually to the tiers of redundant servers of an application service such that power is reduced while maintaining a high level of system availability. Servers which are determined to be relatively inactive are moved to a free pool. Certain servers of the free pool are maintained in a hot standby state, while others are powered- off or set to operate in a low power mode. During times of high load, the servers in the hot standby state can be provisioned quickly into the application service.

WO 2004/049157 A3

WO 2004/049157 A3

SE, SI, SK, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:

— *with international search report*

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

(88) Date of publication of the international search report:
6 May 2005

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

AUTOMATED POWER CONTROL POLICIES BASED ON APPLICATION-SPECIFIC REDUNDANCY CHARACTERISTICS

BACKGROUND OF THE INVENTION

This invention pertains to clustered computing systems and other clustered information handling systems.

In order to meet the demands of heavily loaded Internet application services, providers of application services have turned to redundancy in order to increase the response time of the application service. Redundancy typically refers to the addition of one or more functionally identical servers to the application service. When a single physical server acts logically as a plurality of servers, the plurality of servers are generally referred to as virtual servers. When adding servers in a redundant server environment, the servers added can be physical or virtual.

The network's deployed in heavily loaded Internet application services typically contain a finite number of network nodes. At each node resides a server or a number of servers. Either the server or the number of servers can be virtual or physical servers or any combination of both.

Service Providers typically have a large number of nodes that may be allocated among multiple customers to provide application services at various points in time. Because the offered workload may vary over time, there will be times when the numbers of nodes in the facility exceed the number of nodes required to provide the service. This presents a problem to the service provider because the excess servers consume power and other resources. The servers which are inactive and remain powered-on not only consume more power but are also susceptible to derated reliability for those reliability components which correlate to total power-on time.

Managing Energy and Server Resources in Hosting Centers by J Chase, D Anderson, P Thakar, Amin Vahdat and R Doyle discloses the design and implementation of an architecture for resource management in a hosting center operating system. The solution disclosed provisions server resources for co-hosted services in a way that automatically adapts to offered load, improves the energy efficiency of server clusters by dynamically resizing the active server set, and responds to power supply disruptions or thermal events by degrading service in accordance with negotiated Service Level Agreements (SLAs).

Load Balancing and Unbalancing for Power and Performance in Cluster-Based Systems by E Pinheiro, R Bianchini, E Carrera and T Heath addresses power conservation for clusters of workstations or PCs. An algorithm is disclosed that makes load balancing and unbalancing decisions by considering both the total load imposed on a cluster and the power and performance implications of turning nodes off.

Energy-Efficient Server Clusters by E Elnozahy, M Kistler and R Rajomony is a paper which evaluates five policies for cluster-wide power management in server farms. The policies employ various combinations of dynamic voltage scaling and node vary-on/vary-off (VOVO) to reduce the aggregate power consumption of a server cluster during periods of reduced workload.

Oceano - SLA Based Management of a Computing Utility by K Appleby, S Fakhouri, L Fon, G Goldszmidt, M Kalantar, S Krishnakumar, D Pazel, J Pershing and B Rochwerger discloses a prototype of a highly available, scalable, and manageable infrastructure for an e-business computing utility. It enables multiple customers to be hosted on a collection of sequentially shared resources.

US Patent Application Publication 2002/0004912 discloses a network architecture, computer system and/or server, circuit, device, apparatus, method, computer program and control mechanism for managing power consumption and workload in computer system and data and information servers. Further provides power and energy consumption and workload management and control systems and architectures for high-density and modular multi-server computer systems that maintain performance while conserving energy.

To maximize cost savings it would seem to be beneficial to power off as many servers as possible. Powering off the servers, however, exposes the service provider to the risk of not being able to provision servers quickly enough during times of peak load or failure of servers that provide the application service.

SUMMARY OF THE INVENTION

Accordingly there is provided a computer program comprising program code means adapted to carry out the following method steps when executed on a computer: (a) determining the workload of each server of a cluster of

servers which provide an application service; (b) determining the topology of the application service wherein said topology determination includes a correspondence between a first component of the application service and the server on which the first component is executed; and (c) setting the power state of at least one server of the cluster of servers based upon said workload determination and said topology determination.

The invention provides a program product comprising: a computer usable medium having computer readable program code embodied therein, the computer readable program code in said program product being effective in executing the steps of: (a) determining the workload of each server of a cluster of servers which provide an application service; (b) determining the topology of the application service wherein said topology determination includes a correspondence between a first component of the application service and the server on which the first component is executed; and (c) setting the power state of one server of the cluster of servers based upon said workload determination and said topology determination.

Preferably there is provided a way of reducing the power consumed by a clusters of redundant servers.

The present invention preferably provides a solution to the provider of an Internet application service seeking to power down inactive servers while at the same time eschewing the risk of not being able to provision servers quickly enough during times of peak load or server failure. In one embodiment the workload of each server of a cluster of servers which provide an application service is made. Then, the topology of the application service is determined. This topology information includes a correspondence between the components of the application service and the servers on which the components are run. And based on this workload and topology information, the power state of one or more servers is then changed.

In the preferred embodiment, the concepts of the present invention are provided in the form of a computer program product. The product is distributed on a computer readable medium such as a floppy disk or CD-ROM and installed into pre-existing (or provided) computer systems for the benefit of the customer.

In a different embodiment, the concepts of the present invention are provided in the form of an apparatus and can include the servers themselves.

In a still different embodiment, the concepts of the present invention are provided in the form of an apparatus and a program product or method and in any combination and can include the servers being managed.

In one embodiment the setting (c) is to a low power state when said workload determination determines a workload which is below a predetermined threshold.

In one embodiment, the setting (c) is further based upon the extent to which server resources are fully utilized.

Preferably the low power state is a state selected from the group consisting of: standby, sleep, hibernate, and off states.

Preferably the topology determination includes a first count of the total number of active servers and wherein said setting (c) is further a function of a ratio of powered-on inactive servers to the first count and includes a provision for setting a minimum number of powered on inactive servers.

Preferably the setting (c) is to a low power state when said workload determination determines a workload indicative of the elapsed time since the one server was last reset.

Preferably the setting (c) is to a powered-on state when said workload determination determines a workload which is above a predetermined threshold.

According to a preferred embodiment, there is provided a computer program comprising program code means adapted to perform the following method steps when executed upon a computer: determining the workload of each server of a first tier of scalable redundant servers which provide an application service; determining the topology of the application service wherein said topology determination includes a first count of the total number of active first-tier servers and a correspondence between a first component of the application service and the first-tier server on which the first component is executed; transitioning a first-tier server between

the application service and a free pool of first-tier servers based upon said workload determination; and controlling the power applied to each of the servers of the free pool of first-tier servers according to a first predetermined power management policy and based upon said topology determination.

According to a preferred embodiment the step of determining the workload of each server of each of a cluster of servers comprises determining the workload of each server of a first tier of scalable redundant servers which provide an application service, wherein the topology determination includes a first count of the total number of active first-tier servers and the correspondence is between the first component of the application service and the first-tier server on which the first component is executed, the computer program means further adapted to perform the following step when executed on a computer: (e) transitioning a first-tier server between the application service and a free pool of first-tier servers based upon said workload determination, and wherein the setting step (c) comprises: (f) controlling the power applied to each of the servers of the free pool of first-tier servers based on the topology determination and according to a first predetermined power management policy which is based upon the workload determination.

Preferably the program means is further adapted to perform the following method steps when executed on a computer: (g) determining the workload of each server of a second tier of scalable redundant servers which provide the application service wherein responses from the second tier of scalable redundant servers depend on responses from the first tier of scalable redundant servers; (h) transitioning a second-tier server between the application service and a free pool of second-tier servers; and wherein said topology determination further includes a second count of the total number of active second-tier servers, and (i) controlling the power applied to each of the servers of the free pool of second-tier servers according to a second predetermined power management policy and based upon said topology determination; wherein the second power management policy is a policy selected from the group consisting of the first power management policy and a policy independent of the first power management policy.

According to one embodiment the transitioning is from the application service to the free pool of first-tier servers when said workload determination determines a workload which is below a predetermined threshold.

Preferably the transitioning is further based upon the extent to which server resources of the transitioning first-tier server are fully utilized.

According to one embodiment the transitioning is from the free pool of first-tier servers to the application service when said workload determination determines a workload which is above a predetermined threshold.

Preferably the first predetermined power management policy applied to the servers of the free pool of first-tier servers maintains a first number of servers in a powered-on inactive state while the remaining servers are set to a low power state.

Preferably the first predetermined power management policy applied to the servers of the free pool of first-tier servers is a function of a ratio of powered-on inactive servers to the first count and includes a provision for setting a minimum number of powered-on inactive servers.

Preferably the low power state is a state selected from the group consisting of: standby, sleep, hibernate, and off states.

According to another aspect, the invention provides a method comprising: (a) determining the workload of each server of a cluster of servers which provide an application service; (b) determining the topology of the application service wherein said topology determination includes a correspondence between a first component of the application service and the server on which the first component is executed; and (c) setting the power state of at least one server of the cluster of servers based upon said workload determination and said topology determination.

According to a preferred embodiment, there is provided a method comprising: (a) determining the workload of each server of a first tier of scalable redundant servers which provide an application service; (b) determining the topology of the application service wherein said topology determination includes a first count of the total number of active first-tier servers and a correspondence between a first component of the application service and the first-tier server on which the first component is executed; (c) transitioning a first-tier server between the application service and a free pool of first-tier servers based upon said workload determination; and (d) controlling the power applied to each of the servers of the free pool of first-tier servers according to a first

predetermined power management policy and based upon said topology determination.

According to another aspect, there is provided an apparatus comprising: a workload monitor which detects the workload of each server of a cluster of servers which provide an application service; a topology sensor which determines the topology of the application service including a correspondence between a first component of the application service and the server on which the first component is executed; and a power controller which sets the power state of one server of the cluster of servers based upon the workload as determined by said workload monitor and the topology as determined by said topology sensor.

Preferably the apparatus comprises each server of the cluster of servers which provide the application service.

According to a preferred embodiment, there is provided an apparatus comprising: a workload monitor which detects the workload of each server of a first tier of scalable redundant servers which provide an application service; a topology sensor which determines the topology of the application service including a first count of the total number of active first-tier servers and a correspondence between a first component of the application service and the first- tier server on which the first component is executed; a move module which transitions a first-tier server between the application service and a free pool of first-tier servers based upon the workload as determined by said workload monitor; and a power controller which sets the power state of each of the servers of the free pool of first-tier servers according to a first predetermined power management policy and based upon the topology as determined by said topology sensor.

In a preferred embodiment, the apparatus comprises each server of the first tier of scalable redundant servers which provide the application service.

BRIEF DESCRIPTION OF THE DRAWINGS

Preferred embodiments of the present invention will now be described, by way of example, and with reference to the following drawings:

Fig. 1 is a topological diagram of an exemplary application service in which power saving concepts are deployed in accordance with a preferred embodiment of the present invention.

Fig. 2 depicts the provisioning of servers of the application service of Fig. 1 according to a preferred embodiment of the present invention, including the moving of servers between the application service and a free pool. The free pool of servers are kept in a hot standby state and a low power state.

Fig. 3 is a flow diagram of the power and redundancy management policies in accordance with the teachings of a preferred embodiment of the present invention.

Fig. 4 is a block diagram of an alternative embodiment of the invention implemented in the form of an apparatus.

DETAILED DESCRIPTION OF THE ILLUSTRATIVE EMBODIMENTS

Referring now more particularly to the accompanying drawings, Fig. 1 depicts the topology of an exemplary application service and in which the power saving concepts of the current invention are utilized. The application service of Fig. 1 is comprised of web servers 100 and network dispatchers 101 and 102. The network dispatchers 101 and 102 function as load balancing components that distribute web requests to web servers. A single hot-standby network dispatcher 102 is shown in Fig. 1, however, an application service can have any number of network dispatchers with any subset of them functioning in the hot standby state. The function of the network dispatcher is to be explained in further detail hereinafter. The term hot- standby will also be explained as the description of the preferred embodiments progresses. The web servers 100 perform the function of serving the web pages and are functionally identical. The network dispatcher 101 receives requests for web pages from the Internet and forwards the requests to the web servers 100. The web servers 100 process the individual requests and send the individual responses back to the clients on the Internet. The hot standby network dispatcher 102 is used to take over the role of the network dispatcher 101 in the case of a failure on the part of network dispatcher 101. The application service of Fig. 1 is shown with three servers and two network dispatchers. However, in general the application service can contain any number of web servers and any number of network dispatchers. Further, the application service is shown as having two tiers of servers, tier 1 for the web servers and

tier 2 for the network dispatchers. Nevertheless, the application service can contain any number of tiers of redundant servers. Each tier performs a distinct function in the application service. For example, another tier can be added for servers performing a web or database application which feed information to the web servers 100. Indeed, any server function which can benefit from a scalable redundant architecture can also benefit from the concepts of the present invention. These functions, for example, include a proxy cache, FTP, DNS, Gopher, FrontPage ™ and authentication functions. The database application can be any database application including a relational database such as IBM's SQL ™.

The term -- load -- can be defined as the number of requests arriving at the network dispatcher 101 per unit of time. According to the preferred embodiment, in the case where the load on the application service is low, the number of servers used for the application service is reduced by removing certain servers from the application service. Power is then reduced by controlling the amount of power consumed by a subset of the servers removed from the application service. Additionally, a redundancy management policy is implemented for the servers of any given tier such that the application service has enough online powered-on capacity to meet the workload, plus a certain amount of spare online powered-on capacity in the form of one or more additional servers. For the example given in Fig. 1, When the load on the application service is very low the number of web servers 100 used for the application service can be reduced, from say three to two, by powering off one of the web servers for example. Furthermore, rather than powering off the web servers completely, it is also possible to save power by setting the power state to one of the industry standard semi powered-on or ultra low power states such as standby, sleep, hibernate, and other power saving states and modes. Note that the standby state is a state separate and distinct from the hot standby state which is described hereinbelow.

Fig. 2 shows the provisioning and operational states of the servers in a tier of redundant servers performing an identical function in the application service. The servers are moved to and from the active state 201 in the application service and a free pool 202 depending on the workload experienced by the servers. The active state 201 is a fully powered-on state in which a server is engaged in the application service. The servers in the free pool 202 are inactive in the particular application service under consideration. As previously mentioned, a redundancy management policy is implemented which assures enough online powered-on capacity to meet the workload demands of the application

service, plus a certain amount of spare online powered-on capacity in the form of one or more additional servers. These spare online powered-on servers are referred to as hot standby servers. According to the redundancy management policy, a certain number of the servers of the free pool 202 are kept in the hot standby state 203. These hot standby servers, although inactive in the application service, can be provisioned quickly to the active state 201 in case the workload demands on the application service exceed a predetermined threshold. To reduce power, the servers of free pool 202 which are not to be kept in a hot standby state can be set to a low power state 205.

In the preferred embodiment, the topology of the application service is also taken into account when transitioning the servers between the active state 201 and the free pool 202, and between the hot standby 203 and low power 205 states within the free pool. The determination of topology can take many forms and various examples will be given herein. These examples however are not to be taken as limiting the scope of the invention. As a first example, the topology determination can take the form of determining the total number of servers currently active in the application service. In another example, the topology determination can take the form of determining the individual health of each of the servers. In determining health, the topology determination can focus on the amount of remaining memory or the extent to which memory, CPU, I/O, for any other system resource has been exhausted. In determining health, it is well-known that certain operating systems need to be reset (rebooted / restarted) every so often; thus, the topology determination can focus on the total amount of time elapsed since any given server has been reset. System bottlenecks of any sort can also be taken into account.

In still another example, the application service operates in a virtual server environment. In the virtual server environment there is not necessarily a one-to-one correspondence between a virtual server and a physical server. In this environment, the topology determination will consider whether the physical server is performing more than one function or whether the physical server is involved in more than one tier of redundant servers. For example, a physical server may be running hundreds of virtual Apache servers. In this example all Apache servers are operating in the same tier and are redundant. In still another example, the physical servers can be running virtual applications of several tiers such as proxy cache, or network dispatcher tiers. In this virtual server environment, before a server is set to the low power state 205, the topology of all virtual servers within any given physical server is taken

into account and a physical server is not powered down unless all of the virtual functions performed by the physical server are determined to be in the free pool 202 and in the hot standby state 203. Once the appropriate workload and topology determinations have been made, and once it is determined that there are a sufficient number of hot standby servers in the free pool to meet the demands of the application service, then and only then will a physical server be set to a low power state or powered off completely.

A more specific example of topology and workload determination will now be given. In this example, the number of servers to be kept in the hot standby state 203 per each function type are defined by the below parameters. Here, function type refers to the function performed by the tier of redundant servers.

minHotInFreePool - This parameter provides a provision for setting a minimum number of hot standby servers of this function type in the free pool.

addOnlineRatio - This parameter represents the desired ratio between the number of servers of this function type actively participating in the application service to the number of additional servers of this function type that would be kept in the hot standby state 203. For example, if one hundred servers are actively participating in the application service, and the value of addOnlineRatio is 20, then this particular application service would require 100/20=5 hot standby servers in the free pool 202. If there were 15 total servers in the free pool 202, then the remaining 15-5=10 servers can be powered off or otherwise set to a low power mode of operation.

deployedInAllApplicationServices - This parameter denotes the total number of servers of this function type that are currently participating in the service provider's environment.

hotServersInFreePool - This parameter represents the number of servers that are to be in the free pool 202.

Now, the number of servers of a particular function type to be kept in the hot standby state 203 is determined as follows.

hotServersInFreePool =
minHotInFreePool + (deployedInApplicationServices/addOnlineRatio)

In an alternative embodiment, hotServersInFreePool can be computed as follows.

```
hotServersInFreePool =
(minHotInFreePool > (deployedInApplicationServices/addOnlineRatio) ?
minHotInFreePool :
minHotInFreePool + (deployedInApplicationServices/addOnlineRatio)
```

The power management system of this example will attempt to maintain the number of hot servers of a particular type in the free pool 202 to be equal to hotServersInFreePool. The remaining servers in the free pool 202 can be placed into the low-power state 205.

In the preferred embodiment of this invention, and referring now to both Figures 1 and 2, redundancy and power management policies are applied separately for each tier of redundant servers. These tiers are depicted as tier 1 and tier 2 in Fig. 1, and are encircled with dashed lines. The policies are applied separately because each function type is likely to have specific needs. For example, the minimum number of servers which are to be kept in hot standby state 203 are likely to vary across function type. Where the minimum number of web servers 100 can be almost any number, the minimum number of hot standby network dispatchers 102 can be more specific. While the hot standby network dispatcher 102 is not necessary utilized at all times, it cannot be powered off, as it needs to take over the active role in case of failure of the current network dispatcher 101. Power and redundancy management policies can be specified for network dispatchers such that (1) at least two network dispatcher servers must be online at all times, (2) network dispatchers can be powered on to meet additional workload, and (3) network dispatchers can be powered off as the workload decays so long as at least two are online at all times. Alternatively, the power and redundancy management policies applied to various tiers of an application service can be the same.

Fig. 3 is a flow diagram of the power and redundancy management policies in accordance with the teachings of a preferred embodiment. In step 301 the workload of each server of a tier of redundant servers is determined. The workload of any given server is directly related to the load on the application service as a whole, and is further dependent on server specific attributes. Determining the workload of the server is well known in the art and will therefore not be described in any further detail. Once the workloads have been determined processing moves to step

302. In step 302, the topology of the application service is determined as per the above discussion on topology determination. The topology determination includes an accounting of the components of the application service and the servers on which those components execute. And if the application service utilizes virtual servers, a further topological determination is made relative to the virtual servers and corresponding physical servers. Once the workload and topology have been determined according to steps 301 and 302, processing continues at step 304 wherein it is determined whether a transition of any particular server is required.

Transitioning of servers is required when the load placed on application service either exceeds or does not meet predetermined high and low load thresholds. If neither of the thresholds are met, processing continues at step 301. If, on the other hand, either of the thresholds are met, processing continues at step 305 wherein the servers are moved according to the following criteria. In step 305, the servers in the application service are moved out of the application service and into the free pool 202 during times of low load as determined in step 301. Conversely, the servers are moved from the free pool 202 back into the application service during times of high load as determined in step 301.

Processing then continues at step 307 and 308 wherein the power management policies described hereinabove are applied. In step 307 a decision is made relative to the power applied to the servers of the free pool 202 in accordance to the transition or transitions made to the servers in step 305. If it is determined that the current level of power applied to the servers of the free pool 202 is satisfactory, and no change is needed, processing continues at step 301. If on the other hand it is determined that more servers are needed in hot standby mode, or that fewer servers are needed, processing continues at step 308 wherein the power operating levels of one or more servers are set. In step 308, the power management policies discussed hereinabove are applied. As discussed, these policies take into account the workload and the topology as determined in steps 301 and 302. If these power management policies determine that the number of servers in the free pool 202 which are in the hot standby state 203 can be increased based upon a predetermined threshold, free pool servers in the low-power state 205 can be set to the hot standby state 203. If the power management policies determine that the number of servers in the free pool 202 which are in the hot standby state 203 can be decreased based upon a different predetermined threshold, free pool servers in the hot standby state 203 can be set to the

low-power state 205. The power thresholds can be the same or different and can be either dependent or independent of each other. Once the power level or power levels have been set, processing then continues at step 301 wherein the process repeats.

Figure 4 depicts an alternative embodiment of this invention implemented as hardware apparatus 400. Apparatus 400 monitors, senses, moves and controls the power of servers 409 as described hereinabove and as the further described herein below. The servers 409 can be any of the servers or any of the tiers of servers discussed thus far. The apparatus 400 is comprised of workload monitor 401, topology sensor 402, move module 405 and power controller 408 and implements those power and redundancy management policies described herein. Workload monitor 401 determines the workload of each of the servers 409 and analogously performs the functions as described in step 301 with respect to Figure 3. Topology sensor 402 determines the topology of each of the servers 409 and likewise performs the functions as described in step 302 with respect to Figure 3. Move module 405 acts upon the servers 409 analogously to the steps 304 and 305 discussed with respect to Figure 3. Power controller 408 controls the power settings of each of the servers 409 and analogously performs the functions as described in steps 307 and 308 of Figure 3. Furthermore, apparatus 400 can be implemented as a single unit which when coupled to the servers 409 performs the functions as described herein.
Alternatively, apparatus 400 can be implemented as a distributed series of units 401,402,405, and 408. Apparatus 400 can constructed in any of various hardware implementation methods known in the art; such as, gate arrays, microprocessors, microcomputers, custom VLSI modules, embedded network processors, etc.

CLAIMS

1. A computer program product comprising memory having computer readable code embodied therein for execution on a computer, said code comprising:

(a) determining means for determining the workload of each server of a cluster of servers which provide an application service;

(b) determining means for determining the topology of the application service wherein said topology determination includes a correspondence between a first component of the application service and the server on which the first component is executed; and

(c) setting means for setting the power state of at least one server of the cluster of servers based upon said workload determination and said topology determination, wherein said setting means (c) is to a low power state when said workload determination determines a workload which is below a predetermined threshold and wherein said topology determination includes a first count of the total number of active servers and wherein said setting means (c) is further a function of a ratio of powered-on inactive servers to the first count and includes a provision for setting a minimum number of powered on inactive servers.

2. The computer program product of Claim 1 wherein said setting means c) is further based upon the extent to which server resources are fully utilized.

3. The computer program product of Claim 1 wherein said setting means (c) is to a powered-on state when said workload determination determines a workload which is above a predetermined threshold.

4. The computer program product of Claim 1, wherein the determining means for determining the workload of each server of each of a cluster of servers comprises determining the workload of each server of a first tier of scalable redundant servers which provide an application service, wherein the topology determination includes a first count of the total number of active first-tier servers and the correspondence is between the first component of the application service and the first-tier server on which the first component is

executed, the computer program product further adapted to perform the following step when executed on a computer:

(e) transitioning means for transitioning a first-tier server between the application service and a free pool of first-tier servers based upon said workload determination,

and wherein the setting means (c) further comprises:

(f) controlling means for controlling the power applied to each of the servers of the free pool of first-tier servers based on the topology determination and according to a first predetermined power management policy which is based upon the workload determination.

5. The computer program product of Claim 4, further comprising:

(g) determining means for determining the workload of each server of a second tier of scalable redundant servers which provide the application service wherein responses from the second tier of scalable redundant servers depend on responses from the first tier of scalable redundant servers;

(h) transitioning means for transitioning a second-tier server between the application service and a free pool of second-tier servers; and

wherein said topology determination further includes a second count of the total number of active second-tier servers, and

(i) controlling means for controlling the power applied to each of the servers of the free pool of second-tier servers according to a second predetermined power management policy and based upon said topology determination;

wherein the second power management policy is a policy selected from the group consisting of the first power management policy and a policy independent of the first power management policy.

6. The computer program product of Claim 4 wherein said transitioning means transitions from the application service to the free pool of first-tier servers when said workload determination determines a workload which is below a predetermined threshold.

7. The computer program product of claim 6, wherein said transitioning means for transitioning is further based upon the extent to which server resources of the transitioning first-tier server are fully utilised.

8. The computer program product of Claim 4 wherein said transitioning means transitions from the free pool of first-tier servers to the application service when said workload determination determines a workload which is above a predetermined threshold.

9. The computer program product of Claim 4 wherein the first predetermined power management policy applied to the servers of the free pool of first-tier servers maintains a first number of servers in a powered-on inactive state while the remaining servers are set to a low power state.

10. A method comprising:

(a) determining the workload of each server of a cluster of servers which provide an application service;

(b) determining the topology of the application service wherein said topology determination includes a correspondence between a first component of the application service and the server on which the first component is executed; and

(c) setting the power state of at least one server of the cluster of servers based upon said workload determination and said topology determination, wherein said setting (c) is to a low power state when said workload determination determines a workload which is below a predetermined threshold and wherein said topology determination includes a first count of the total number of active servers and wherein said setting (c) is further a function of a ratio of powered-on inactive servers to the first count and includes a provision for setting a minimum number of powered on inactive servers.

11. Apparatus comprising:

a workload monitor which detects the workload of each server of a cluster of servers which provide an application service;

a topology sensor which determines the topology of the application service including a correspondence between a first component of the application service and the server on which the first component is executed; and

a power controller which sets the power state of at least one server of the cluster of servers based upon the workload as determined by said workload monitor and the topology as determined by said topology sensor, wherein the power controller is operable to set the power state of at least one server to a low power state when said workload monitor detects a workload which is below a predetermined threshold and wherein said topology sensor counts the total number of active servers (first count) and wherein said power controller is operable to set the power as a function of a ratio of powered-on inactive servers to the first count and includes a provision for setting a minimum number of powered on inactive servers.

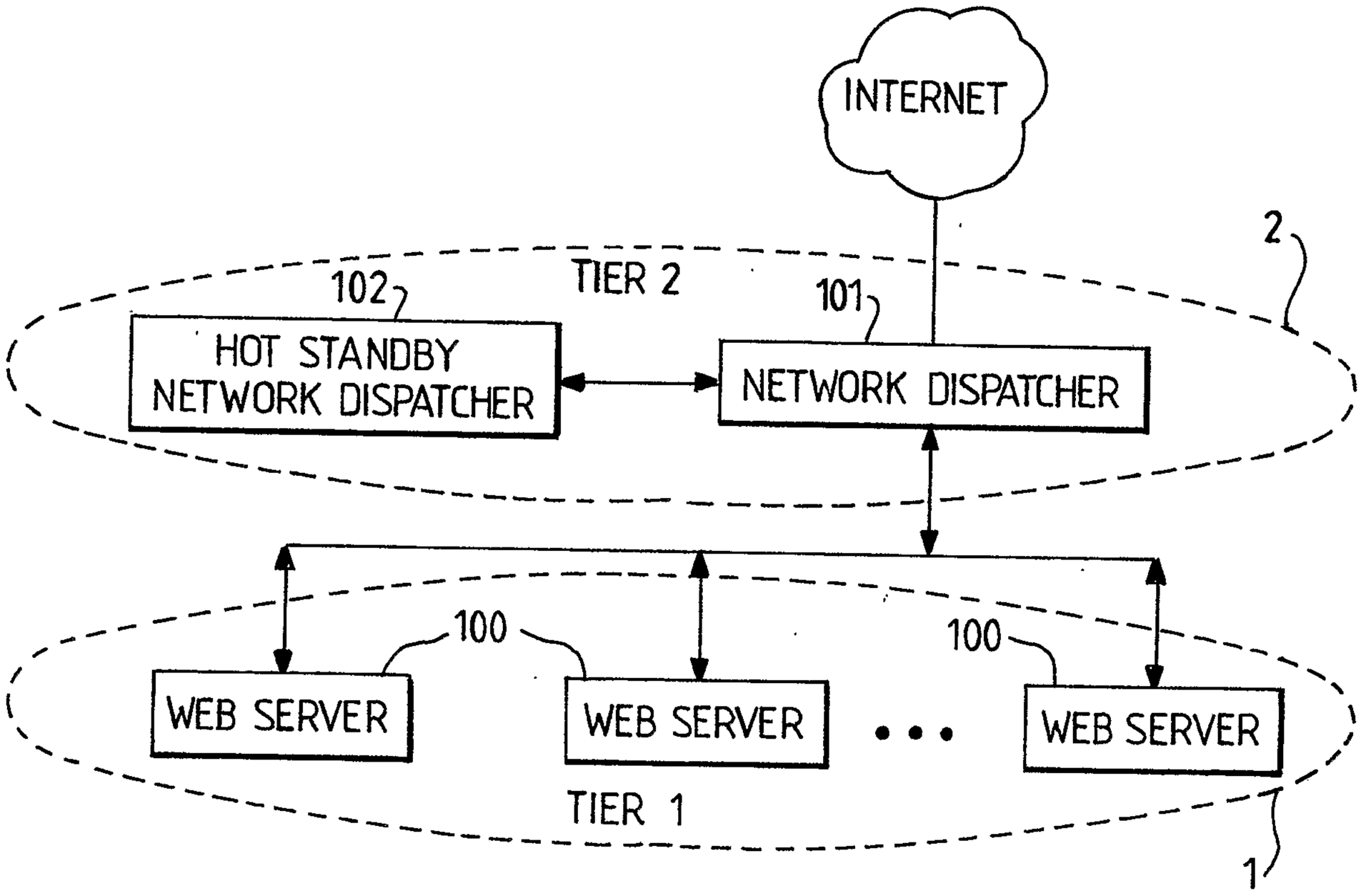

INTERNET
2
TIER 2
102
101
HOT STANDBY NETWORK DISPATCHER
NETWORK DISPATCHER
100
100
WEB SERVER
WEB SERVER
WEB SERVER
TIER 1
1

FIG. 1

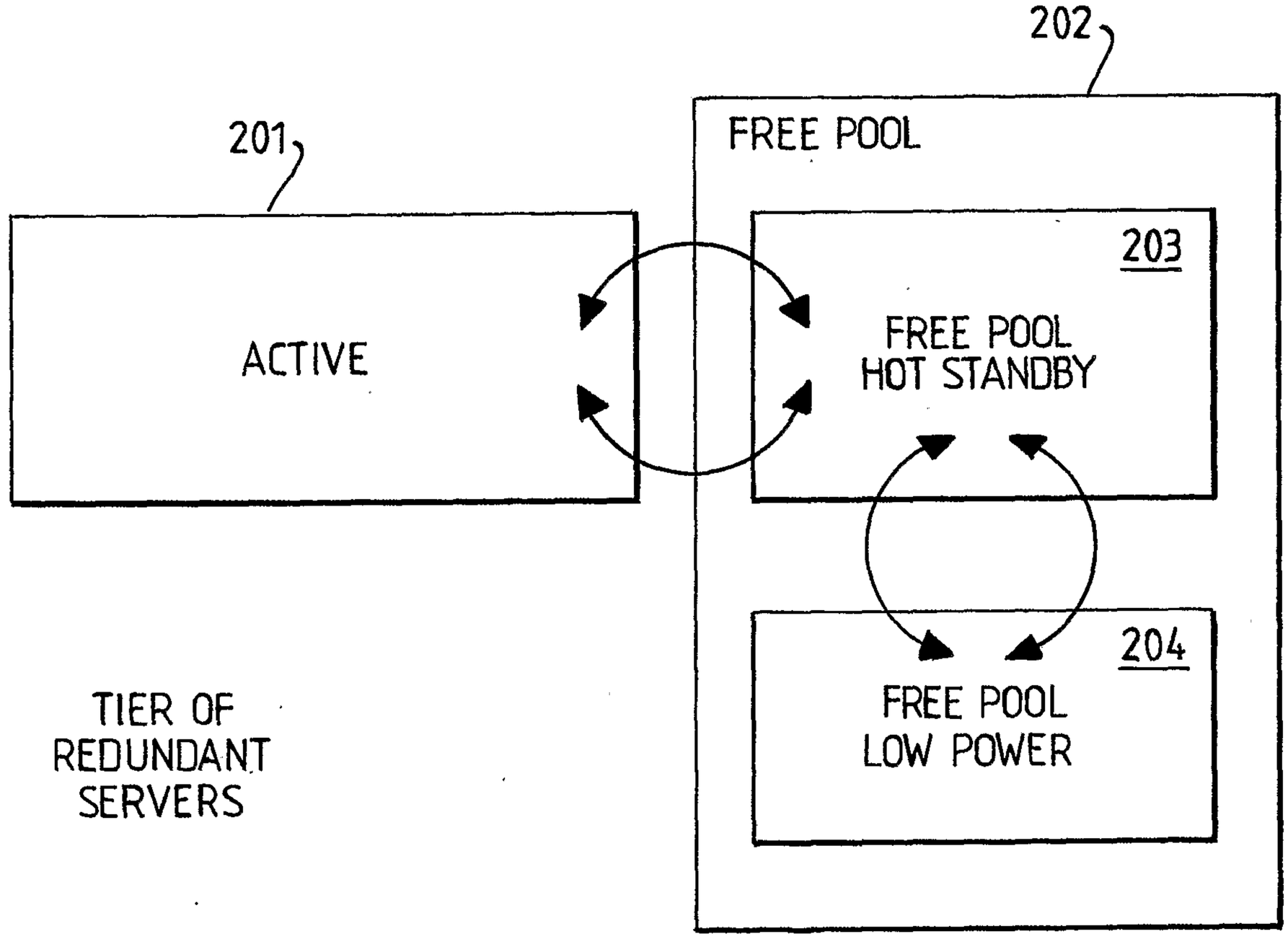

202
FREE POOL
201
ACTIVE
203
FREE POOL
HOT STANDBY
204
FREE POOL
LOW POWER
TIER OF
REDUNDANT
SERVERS

FIG. 2

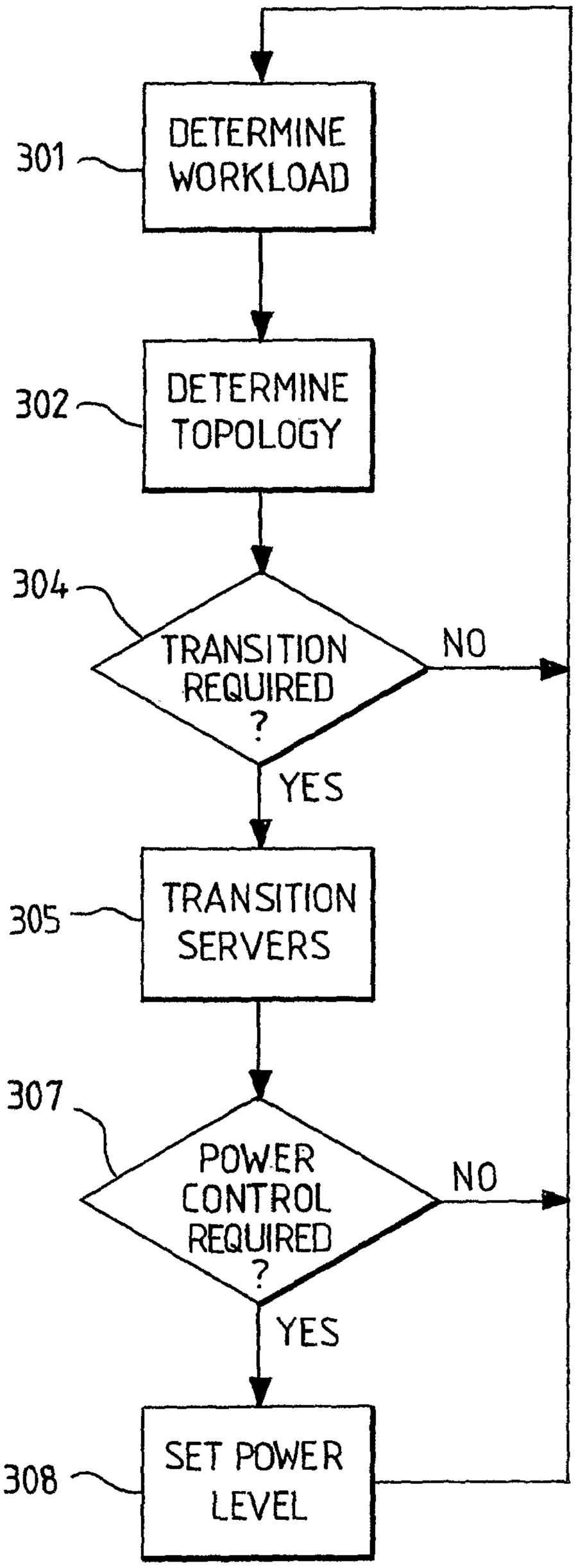

301
DETERMINE WORKLOAD
302
DETERMINE TOPOLOGY
304
TRANSITION REQUIRED ?
NO
YES
305
TRANSITION SERVERS
307
POWER CONTROL REQUIRED ?
NO
YES
308
SET POWER LEVEL

FIG. 3

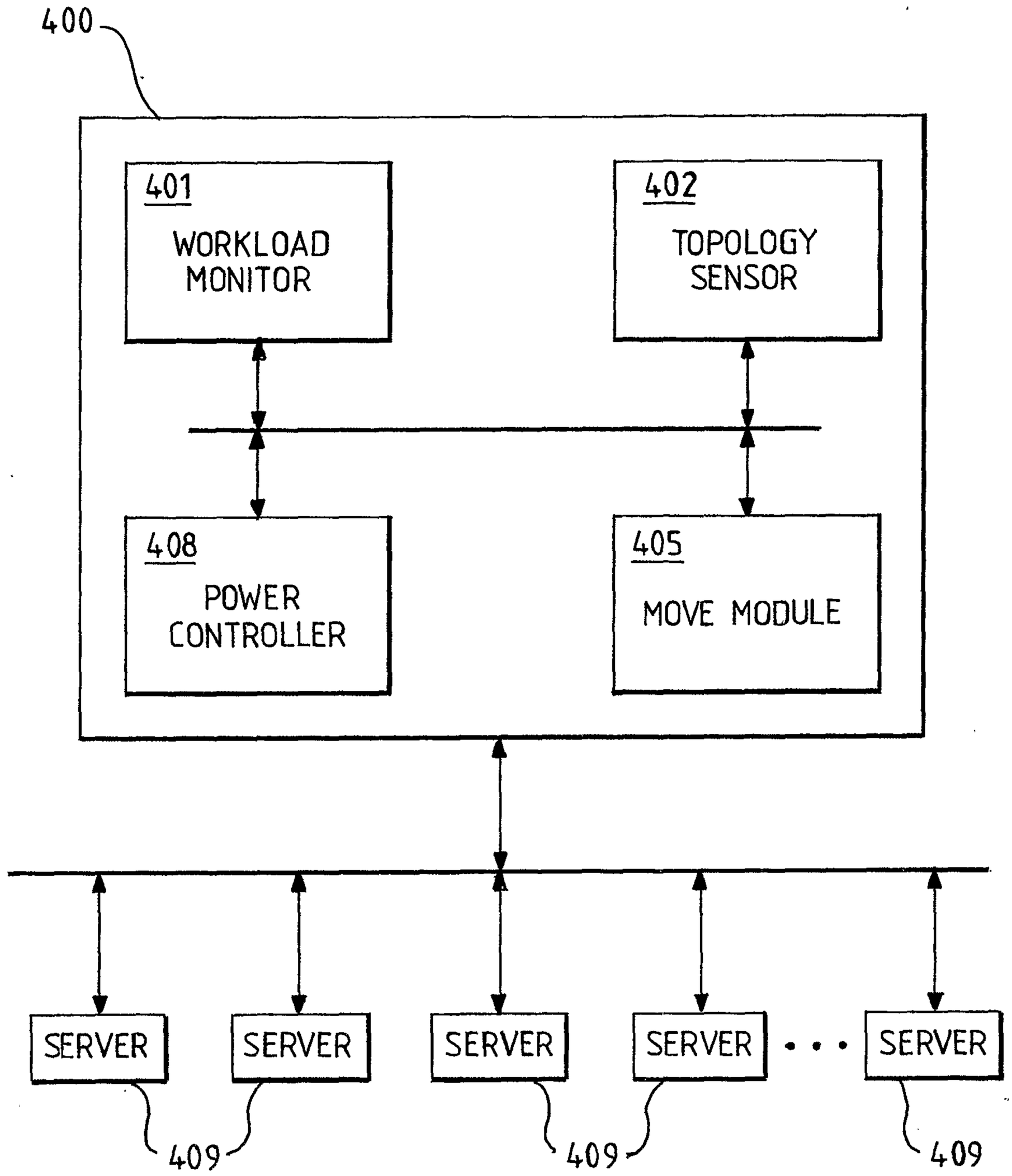
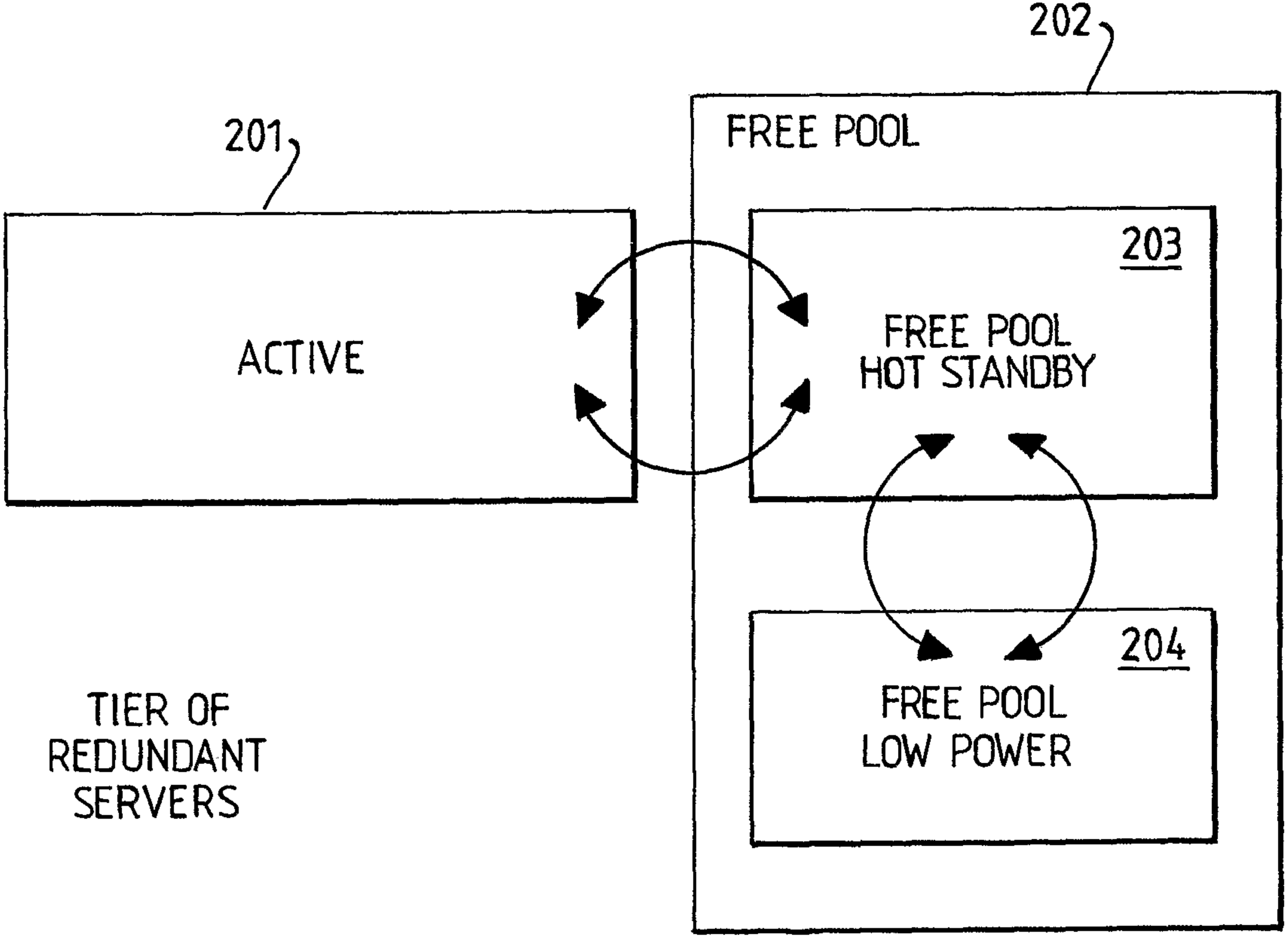

400
401
WORKLOAD MONITOR
402
TOPOLOGY SENSOR
408
POWER CONTROLLER
405
MOVE MODULE
SERVER
SERVER
SERVER
SERVER
SERVER
409
409
409

FIG. 4


202
FREE POOL
201
ACTIVE
203
FREE POOL
HOT STANDBY
204
FREE POOL
LOW POWER
TIER OF
REDUNDANT
SERVERS