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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
21 February 2002 (21.02.2002)

PCT

(10) International Publication Number
WO 02/15526 A1

(51) International Patent Classification[7]: H04L 29/06, 29/14

(21) International Application Number: PCT/US00/22216

(22) International Filing Date: 11 August 2000 (11.08.2000)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant *(for all designated States except US)*: **NETSCAPE COMMUNICATIONS CORPORATION** [US/US]; 501 E. Middlefield Road, Mountain View, CA 94043 (US).

(71) Applicants and
(72) Inventors *(for all designated States except US)*: **MOUNT, George** [US/US]; 342 Central Avenue, Mountain View, CA 94043 (US). **SRIVATSA, Ramachandra** [IN/US]; 1055 Escalon Avenue, #712, Sunnyvale, CA 94086 (US).

(74) Agents: **GLENN, Michael** et al.; Glenn Patent Group, 3475 Edison Way, Suite L., Menlo Park, CA 94025 (US).

(81) Designated States *(national)*: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, EE, 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, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA.

(84) Designated States *(regional)*: ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
— *with international search report*
— *with amended claims*

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.

WO 02/15526 A1

(54) Title: SERVER-SIDE SESSION MANAGEMENT

(57) Abstract: The invention relates to the backup of a user session on a network. When the user at a client, such as an ISP, makes a request on the network, this request is routed by a load balancer to a primary host server on the network. The selection of the server is done by the load balancer in such a manner as to distribute the load on the system evenly and prevent overloading of one particular server. Once the user request has been routed, user specific information such as a session identification code or a password is provided to the primary host server and the session is set up. A single session must comprise of at least one application process on the selected primary host server. On commencement of the session, the load balancer designates at least one other server on the network to act as primary backup server for the user session set up on the primary host server. The primary backup server stores session inforamtion from the primary host server and in case of a server breakdown, the backup server takes over as the primary server. The load balancer then configures another server to act as backup server for this new primary server. According to th invention, the backup function of the servers is equitably distributed across servers so as to ensure that no one server is excessively burdened with the backup functions.

Server-Side Session Management

BACKGROUND OF THE INVENTION

TECHNICAL FIELD

The invention relates to server side session management. A session is defined as the information that the user provides to the host server using any browser for the purpose of carrying out a sequence of related online transactions. More particularly, the invention relates to maintaining easily retrievable backups of a user session in a scalable and performant way on the network so as to continue managing a session even during server failures at web-hosting sites.

DESCRIPTION OF THE PRIOR ART

Internet usage has exploded over the years and is becoming more and more popular by the day. There has also been a phenomenal increase in inline shopping; *i.e.* e-commerce, with statistics showing that in 1999 there were more online shoppers than all the shoppers in the prior three years put together.

An on-line shopping transaction, which is one of the most significant applications of session management, comprises a number of steps. The purchaser/online shopper begins by visiting multiple sites for research. Then the shopper chooses a particular web site and then within the web site parses through various web pages (links) to narrow down on the particular product. The process ends with the completion of the sales transaction through the input of relevant information required from the purchaser for the purpose.

In this context, session management assumes great significance. In carrying out an on-line transaction, such as a purchase, it is important that the information submitted by the user during the course of an on-line session be stored and be easily retrievable in case of server break down. Thus, a need exists for backing up information and data on a user session, so that the

session information does not get lost in case of a server break down, thus not requiring the user to restart the session all over again.

Although duration of a typical session lasts only from a few minutes to hours, the growing number of Internet users has dramatically increased the number of sessions taking place per second. Therefore, a need exists for robust session management methods on the server side to handle the ever-increasing number of sessions and make the overall online experience a pleasant one for the user. Good session management is important because performance is critical.

Good session management involves *inter alia* restoring the session to the state it was in prior to the failure of the server, thereby not requiring the user to restart the session all over again in case of a server breakdown. There are two main types of session management. One is client side session management and the other is server side session management.

Client Side

1. Session management with cookies
2. Session management using invariable time forms

Server side

1. Session management methods on the server side.

Session management using cookies involves leaving a cookie (block of data) on the PC of the user. This cookie contains information such as the session history, personal information about the user like user name and shopping cart, and any other pertinent information. In the event of disruption of the session due to any reason, such as server failure, the information is stored only in the cookie and the session continues without failure.

Session management using hidden variable within forms involves storing all the information input by the user and transmitting that information also as the user moves from one page to the next. The next form contains the same information input by the user in the first form.

Server side session management methods, broadly speaking, involve storing data about a user session on the network server with which the user is in contact, such a server performing the task of constantly updating and

maintaining recent session information. The process begins with the input of data by the user to the server. The server keeps track of the information and maintains cached copy of session data retrievable for each continuing request. At the same time, such a method proves to be less taxing on the network bandwidth, because the session data does not need to be passed on every request.

Server side session management systems need to be robust, performant and scalable, so as to better address the needs of users and be a reliable system of backup. The backup strategy needs to be efficient so as to come into play immediately on the failure of the primary server so that information about the session gets transmitted to the backup server, which immediately assumes the functions hitherto performed by the primary server that just failed. The speed at which these transactions occur is important so that the user does not experience any discontinuity or disruption, thus ensuring the overall satisfaction of the session for the end user.

There are several methods of server side session management. According to a traditional method of server side session backup, session information from all the different servers on a network are backed up in the database directly.

Netscape Enterprise Server 4.0 (NES), manufactured by Netscape Communications Corporation of Mountain View, California has a server side session management mechanism for session backup on a single machine called the MMapSessionManager. In this implementation each server writes session information to a common file, maintained by MMapSessionManager. Under this implementation, if one of the servers fails, other servers can access the information of the failed server from the Session Manager and thus restore the session seamlessly. Each server is a peer and can handle a request from any user equally.

The main disadvantages to this kind of a system of session management are in the areas of file storage, a single point of failure and scalability. Performance of a session is critical, such performance being dependent upon the session information files being stored locally so that retrieval may be faster. Although, storing of session backups locally makes retrieval faster and performance efficient, storing of all the files locally on one machine, *i.e* the MmapSessionManager means that there is a single point of failure. As a result the system is ill equipped to overcome an eventuality where the

hardware holding the Session information fails. In such a situation, the entire session information, which was backed up by the Sessions Manager is lost forever with no way to retrieve it whatsoever.

Even in cases where files are stored in a remote file server because efficient or speedy performance is not a critical issue, the system still has a single point of failure as discussed earlier, meaning thereby that if the storage server hardware fails, the session data thus lost becomes irretrievable.

Netscape Application Server (NAS) has a different implementation of safe server-side session management. ANAS is an application server and therefore has numerous functionalities such as database access across servers. Server-side session management is an important feature of NAS.

Under NAS, when a user conducts a session with a particular server, all session information is backed up synchronously with both a Master instance as well as a Slave instance. The server conducting the session is given a token to update the session information. If any other server needs to update the session, then the token must be transferred from the owner to the other server through the Master instance.

When the server conducting the session fails, the user is routed to another server, which attempts to attain the token through the Master instance. The Master instance detects that the server which owned the token has failed and grants a new token to the server now conducting the user's session.

Safe server-side session management on NAS is not performant. During updates to the session, the data must be written to the Master instance which in turn updates the Slave, all of which must be done before responding to the user.

Scalability also suffers greatly, because there are only two servers responsible for session backup. The load cannot be distributed among more servers as the number of servers on the network grows.

Hence, it is evident that a need exists for a mechanism of session management that includes failover servers to take over the load of the failed primary server without compromising speed and performance. Such a system should be scalable so as to meet the growing demands of the e-

commerce community. Further, a need exists for a system of session management, which is performant

SUMMARY OF THE INVENTION

The current invention relates to a scalable and performant backup method for server-side sessions. When a user first visits a web site, he is routed by a load balancer to a primary server which will take the responsibility to service his requests. The server sets up a session for that user and returns an identifier to the user's browser. The session manager sets up a backup server for the user's session based on the identifier. All information written to the primary server's session is sent to the backup as well. In case the primary server fails, the load balancer follows the same algorithm as the session manager to assign the backup server as the new primary server. The new primary server then designates new backup server(s) which take on the responsibility for backing up the session. According to the present invention, the backup function of the servers is equitably distributed across servers so as to ensure that no one server is excessively burdented with the backup functions.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a schematic representation of a server side session management system using a load balancer depicting the takeover mechanism; and

Fig. 2 is a flow diagram that illustrates the failover algorithm for session management on a network.

DETAILED DESCRIPTION OF THE INVENTION

The invention uses a new and unique method of server-side session management, wherein each web server is equally given the task of backing up sessions from each other web server.

Figure 1 is a schematic diagram, which provides a general overview of how a session is set up and handled by the server-side management system, using a load balancer.

According to a preferred embodiment of this invention, on-line users Joe (16), Kit (17), and Bob (18) login to the site from their respective computers onto a network with four servers A-D. Joe (16), Kit (17), and Bob (18) are routed by the load balancer to a particular server. The load balancer will continue to send all requests from Joe (16) to the same server A (11) until that server fails or the session ends. Similarly, the load balancer will continue to send all requests from Kit (17) to the same server B (12) until that server fails or the session ends. Finally, the load balancer chose server C (13) to service Bob's (18) requests.

When Joe (16) first visits the web site, server A (11) creates a session for him with the identifier 1234. According to the session management algorithm, the backup machine for session 1234 is designated as server D (14). All data that Joe (16) sends to server A that needs to be stored in the session is copied to server D (14).

When Kit (17) first visits the web site, server B (12) creates a session for him with the identifier 5678. According to the session management algorithm, the backup machine for session 5678 is designated as server C (13). All data that Kit (17) sends to server B (12) that needs to be stored in the session is copied to server C (13).

When Bob (18) first visits the web site, server C (13) creates a session for him with the identifier 9012. According to the session management algorithm, the backup machine for session 9012 is designated as server A (11). All data that Bob (18) sends to server A that needs to be stored in the session is copied to server A (11).

According to the preferred embodiment of this invention, session data is backed up on only a limited set of servers, as determined by the failover algorithm. The load balancer routes all traffic for a single session to a single host. This host has the most up-to-date information about the session because it stores the information locally. When session data is written, it is first written to the memory of the primary server and then backed up to at least one other server, maybe more depending upon the configuration of the session manager. In case of failure of the primary server, the load balancer follows the same algorithm as the session manager to choose the backup server as the new primary server.

At any given time, the session information is stored on the primary server and at least one backup server. For any given primary server, the backup server is

chosen on a per-session basis. This mechanism ensures that the backup load is distributed evenly across all servers.

FAILOVER ALGORITHM FOR SESSION MANAGEMENT

Fig. 2 is a flow diagram that illustrates the failover algorithm for session management. According to the preferred embodiment of this invention, when the user makes a request (100), if it is the first request, then the user is routed to a server via the load balancer and that server is designated to continue serving that user's request and is called the primary server (102). This assignment of a user to a server is called "stickiness."

The primary server assigned to the user then creates a session and assigns backup server(s) for that session. The session data is updated with the request data from the user (103) and the data is copied to the backup server(s) (104).

During subsequent requests from the user to the system (100), if the primary server has not failed (106), then the request is routed to the primary server and the session data is updated and backed up normally (105). If the primary server has failed (106), the load balancer designates the backup server as the primary (107). The primary server then designates backup servers for the retrieved session. The request information is then stored in the session and backed up on the backup servers (109).

BACKUP AND FAILOVER LOAD DISTRIBUTION

According to the preferred embodiment of this invention, on failure of the primary server, all sessions handled by that server should be evenly distributed amongst the remaining servers. Further, the backup algorithm must also spread the load evenly among the other servers.

In a two server scenario, server A is designated as the primary server for half of the sessions. Server B is designated as the backup for all of Server A's sessions. Server A is designated as backup for all of Server B's sessions. Thus, server A must handle half of the load as primary server, as well as backing up half of the total sessions. When server A fails, all sessions are routed to server B, which must handle the full load as primary server.

In a three server scenario, server A, B, and C, each are designated as the primary server for one-third of the total sessions. Servers A, B, and C, each must also act as the backup for each other, giving each of them, the responsibility to back up one-third of the total user sessions. For example, server A must handle half of the backups for server B and half of the backups for server C, each of which have one third of the total session load. Thus, each server handles an equal amount of load as primary server as backup server.

Similarly, with 100 servers, each server handles 1/100 (1%) of the load as primary server, and each also handles 1/99 of the backups for 99 other servers. The total backup load for any server is then 1% of the total user sessions.

An object and advantage of this invention is that this solution is very scalable. An increase in the number of servers on the network does not cause a proportional increase in the backup overload on the individual servers on the network because any one server on the network is only responsible for backing up a fraction of the total number of sessions from the load of every other server.

On server failure, backups of the sessions are distributed among the remaining servers. No one server is loaded more than any other, each taking the load of an equal percent of the sessions from the failed server.

An object and advantage of this invention is that there is no single point of failure on this scheme of session management. In case any one of the servers fails on a network fails, the backup machine assumes the functions of the primary server and another server takes over as the backup server. So at any given point of time there are at least two servers (or more) that have the most recent data regarding the session.

Server performance under this solution is very high. The sessions are stored locally on each primary server and the backup server(s). Also, for a particular session, data is only backed up to a limited number of servers, all from the primary unlike in the case of NAS. On failure, the session does not need to be retrieved from any backup since the backup server is automatically chosen as the new primary server.

Thus, the present invention reduces the disadvantages affecting the other session management systems such as NAS and NES and proves to be a

more effective tool for session management and is a highly performant, scalable and reliable solution.

Accordingly, although the invention has been described in detail with reference to a particular preferred embodiment, persons possessing ordinary skill in the art to which this invention pertains will appreciate that various modifications and enhancements may be made without departing from the spirit and scope of the claims that follow.

CLAIMS

1. A method for managing a user session on a network comprising the steps of:

making a request by a user at a client on the network;
routing said request to a primary host server by a load balancer;
providing user specific information to said primary host server;
setting up a session between said user and said primary host server;
performing at least one application process during any particular session on a selected primary host server;
configuring at least one other server on said network to act as primary backup server for a user session on said primary host server;
storing user session information of said primary host server on at least one other server configured as a primary backup server by the load balancer; and
designating said primary backup server to act as primary host server to take over on failure of said primary host server for that user session.

2. A method according to Claim 1, wherein said load balancer routes all application processes for a single user session to a single primary host server, said load balancer further comprising:

node for receiving a request from a user at a client; and
node for transmitting said request to a server on the network.

3. A method according to Claim 1, wherein user specific information is provided by the session manager in the form of a session identification code for use by a particular user during a session.

4. A method according to Claim 1, wherein said primary host server stores most recent information about a single user session.

5. A method according to Claim 1, wherein data from a particular user session on the primary host server is backed up to at least one other server designated as the primary backup server on the network.

6. A method according to Claim 1, wherein said primary backup server is configured to take over as the primary host server in case of failure of primary host server and continue the user session.

7. A method according to Claim 1, wherein said session manager may designate more servers as secondary and tertiary backup servers.

8. An apparatus for managing a user session on a network comprising :

a terminal for making of a request by a user at a client on said network;
a load balancer to transfer said request to a primary host server;
user specific information in the form of session identification code to set up a session between the user and a primary host server, said session comprising at least one application process; and
a primary backup server for said user session on said primary host server configured by said session manager to take over in case of failure of primary host server.

9. An apparatus according to Claim 8, wherein said load balancer routes all application processes for a single user session to a single primary host server, said load balancer further comprising:

a node for receiving a request from a user at a client; and
a node for transmitting said request to a server on the network.

10. An apparatus according to Claim 8, wherein user specific information is provided by the session manager in the form of a session identification code for use by a particular user during a session.

11. An apparatus according to Claim 8, wherein said primary host server stores the most recent information about a single user session.

12. An apparatus according to Claim 8, wherein data from a particular user session on the primary host server is backed up to at least one other server designated as the primary backup server on the network.

13. An apparatus according to Claim 8, wherein said primary backup server is configured to take over as the primary host server in case of failure of primary host server and continue the user session.

14. An apparatus according to Claim 8, wherein said session manager may designate more servers as secondary and tertiary backup servers.

AMENDED CLAIMS

[received by the International Bureau on 15 August 2001 (15.08.01);
original claims 1-14 replaced by new claims 1-8 (2 pages)]

1. A method for managing user session load in a network of servers, the method comprising the steps of:

assigning a plurality of the servers as primary host servers;

assigning to each one of the primary host servers an equal portion of the user session load;

assigning each one of the primary host servers as a backup server for the rest of the primary host servers; and

when one of the primary host servers fails, distributing the corresponding portion of the user session load equally on the rest of the primary host servers.

2. A method for managing user session load in a network of servers, the method comprising the steps of:

assigning a plurality of the servers as primary host servers;

assigning to each one of the primary host servers a portion of the user session load;

assigning each one of the primary host servers as a backup server for a plurality of the rest of the primary host servers; and

when one of the primary host servers fails, distributing the corresponding portion of the user session load on the plurality of the rest of the primary host servers.

3. The method of Claim 2, wherein the step of assigning to each one of the primary host servers includes assigning to each one of the primary host servers an equal portion of the user session load.

4. The method of Claim 2, wherein the step of assigning each one of the primary host servers as a backup server includes assigning each one of the primary host servers as a backup server for the rest of the primary host servers.

5. The method of Claim 5, wherein the distributing step includes distributing the corresponding portion of the user session load equally on the rest of the primary host servers.

6. An apparatus for managing user session load in a network of servers, the apparatus comprising:

means for assigning a plurality of the servers as primary host servers;

means for assigning to each one of the primary host servers an equal portion of the user session load;

means for assigning each one of the primary host servers as a backup server

for the rest of the primary host servers; and

means for distributing the corresponding portion of the user session load equally on the rest of the primary host servers when one of the primary host servers fails.

7. A computer readable medium embodying a method for managing user session load in a network of servers, the method comprising the steps of:

assigning a plurality of the servers as primary host servers;

assigning to each one of the primary host servers an equal portion of the user session load;

assigning each one of the primary host servers as a backup server for the rest of the primary host servers; and

when one of the primary host servers fails, distributing the corresponding portion of the user session load equally on the rest of the primary host servers.

8. An apparatus for managing user session load in a network comprising:

a database for maintaining user session information; and

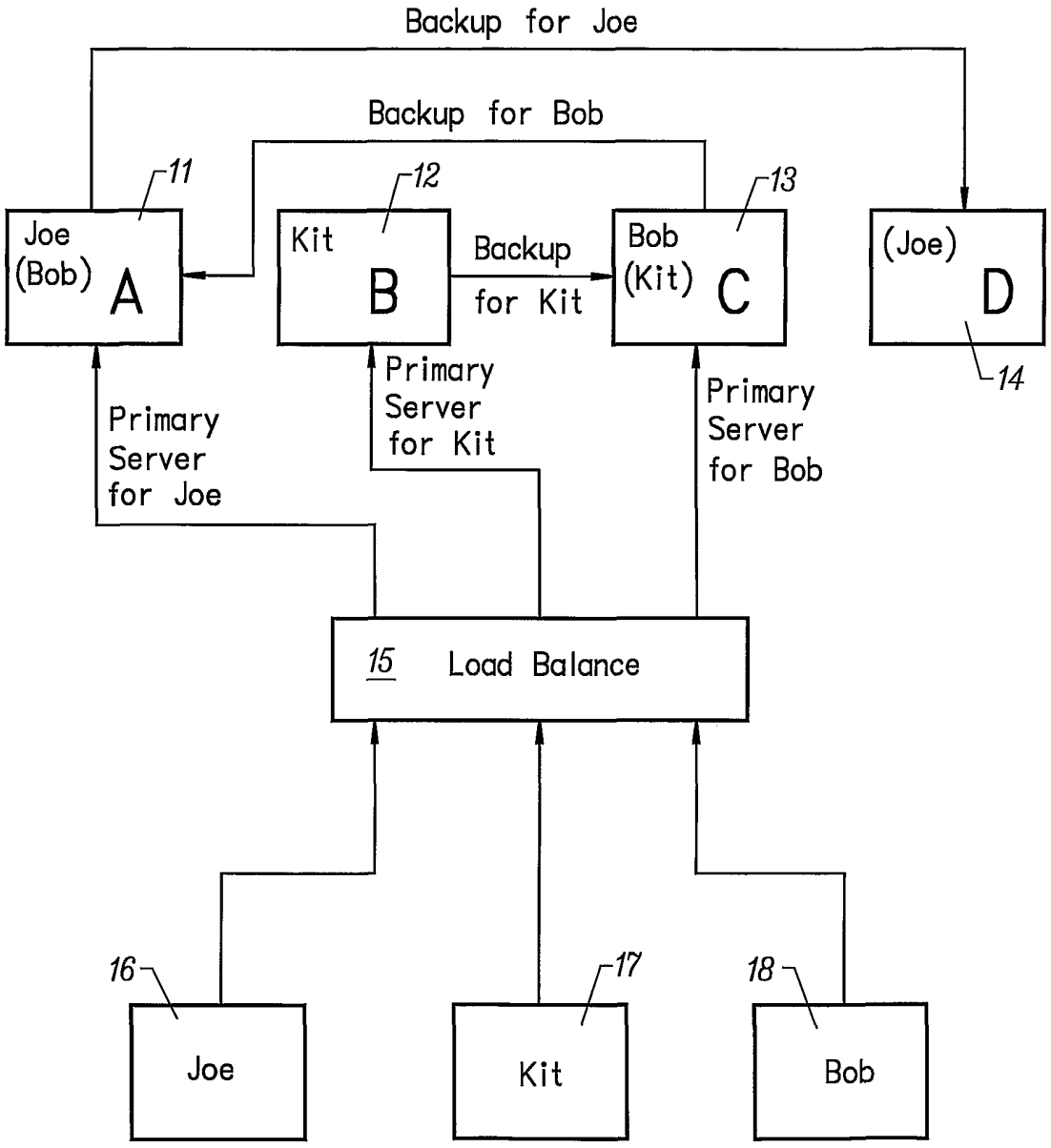
a processor configured for:

assigning a plurality of the servers as primary host servers;

assigning to each one of the primary host servers an equal portion of the user session load;

assigning each one of the primary host servers as a backup server for the rest of the primary host servers; and

when one of the primary host servers fails, distributing the corresponding portion of the user session load equally on the rest of the primary host servers.



(X) indicates backup for user X.

FIG. 1

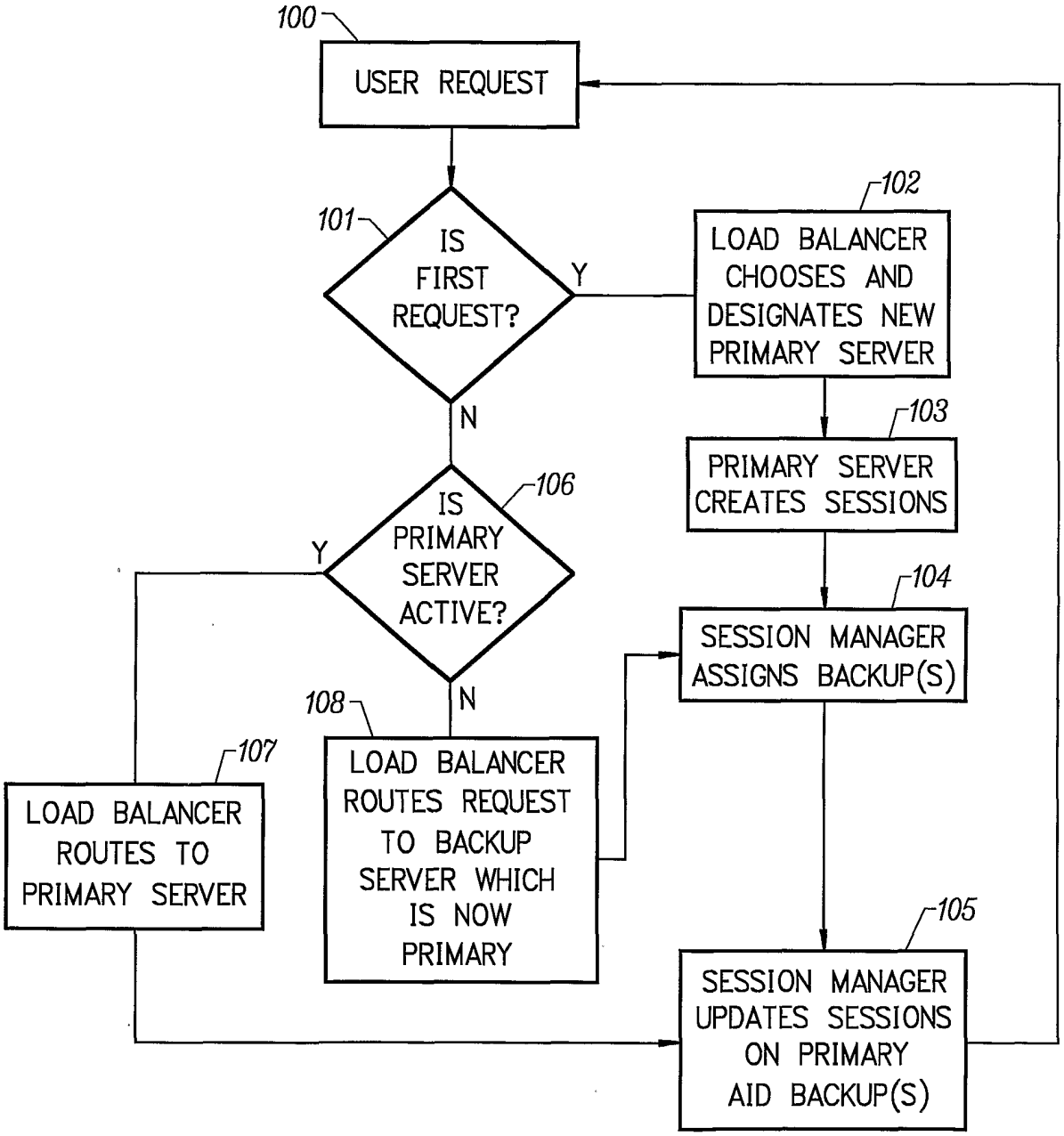

100
USER REQUEST
101
IS FIRST REQUEST?
Y
N
102
LOAD BALANCER CHOOSES AND DESIGNATES NEW PRIMARY SERVER
103
PRIMARY SERVER CREATES SESSIONS
104
SESSION MANAGER ASSIGNS BACKUP(S)
105
SESSION MANAGER UPDATES SESSIONS ON PRIMARY AID BACKUP(S)
106
IS PRIMARY SERVER ACTIVE?
Y
N
107
LOAD BALANCER ROUTES TO PRIMARY SERVER
108
LOAD BALANCER ROUTES REQUEST TO BACKUP SERVER WHICH IS NOW PRIMARY

FIG. 2

INTERNATIONAL SEARCH REPORT

In tional Application No
PCT/US 00/22216

A. CLASSIFICATION OF SUBJECT MATTER
IPC 7 H04L29/06 H04L29/14

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
IPC 7 H04L 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, IBM-TDB, INSPEC

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 5 951 694 A (CHOQUIER PHILIPPE ET AL) 14 September 1999 (1999-09-14) abstract column 8, paragraph 2 claims 9,10; figure 1	1,3-8, 10-14
Y	US 5 155 678 A (FUNAHASI TAKAYUKI ET AL) 13 October 1992 (1992-10-13) abstract column 1, paragraph 2 column 1, line 66 -column 2, line 22 column 3, paragraph 2	1,3-8, 10-14
A	US 5 796 934 A (BHANOT PRADEEP ET AL) 18 August 1998 (1998-08-18) abstract column 5, paragraph 3	1-14

-/--

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

[X] Patent family members are listed in annex.

° Special categories of cited documents :

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
11 July 2001	19/07/2001

Name and mailing address of the ISA	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL – 2280 HV Rijswijk Tel. (+31–70) 340–2040, Tx. 31 651 epo nl, Fax: (+31–70) 340–3016	Blanco Cardona, P

Form PCT/ISA/210 (second sheet) (July 1992)

INTERNATIONAL SEARCH REPORT

Ir ational Application No
PCT/US 00/22216

C.(Continuation) DOCUMENTS CONSIDERED TO BE RELEVANT

Category °	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 5 652 908 A (GLOWNY DAVID ANDREW ET AL) 29 July 1997 (1997-07-29) abstract column 5, paragraphs 1,3	1-14
A	US 6 076 108 A (HUDDLESTON ERIK L ET AL) 13 June 2000 (2000-06-13) abstract column 1, paragraph 3 - paragraph 6 column 7, line 59 - line 61	1-14

INTERNATIONAL SEARCH REPORT

Information on patent family members

International Application No
PCT/US 00/22216

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5951694 A	14-09-1999	US 5774668 A	30-06-1998
US 5155678 A	13-10-1992	DE 3685870 A	06-08-1992
		DE 3685870 T	04-02-1993
		EP 0221274 A	13-05-1987
		JP 62105247 A	15-05-1987
US 5796934 A	18-08-1998	NONE	
US 5652908 A	29-07-1997	EP 0537903 A	21-04-1993
		JP 2566711 B	25-12-1996
		JP 5257916 A	08-10-1993
		US 5537642 A	16-07-1996
US 6076108 A	13-06-2000	US 6085220 A	04-07-2000