

-1-

**SYSTEM AND METHOD FOR CLUSTERED TUNNELING OF
REQUESTS IN APPLICATION SERVERS AND
TRANSACTION-BASED SYSTEMS**

5

Copyright Notice

10

A portion of the disclosure of this patent document contains material which is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

Claim of Priority:

U.S. Provisional Application entitled SYSTEM AND METHOD
15 FOR CLUSTERED TUNNELING OF REQUESTS IN APPLICATION
SERVERS AND TRANSACTION-BASED SYSTEMS, Application No.
60/572,644, filed May 19, 2004, by Andrew Piper (Atty. Docket No.: BEAS-
01564US0), and incorporated herein by reference.

U.S. Patent Application entitled SYSTEM AND METHOD FOR
20 CLUSTERED TUNNELING OF REQUESTS IN APPLICATION SERVERS
AND TRANSACTION-BASED SYSTEMS, Application No. _____,
filed May 16, 2005, by Andrew Piper (Atty. Docket No. BEAS-01564US1),
and incorporated herein by reference.

25 **Field of the Invention:**

The invention is related generally to application servers, messaging systems, and transaction-based systems, and specifically to a system and method for clustered tunneling of requests in those systems.

-2-

Background:

In a typical clustered enterprise level application server environment, clients may access the cluster using the Internet Interoperability Protocol (IIOP). **Figure 1** shows an example of such a clustered system. The cluster **100** comprises a number of servers or cluster members C1 **102**, C2 **104**, C3 **106**, each of which can be identified by an object request (IOR) specifying the servers name or address, and a port identifier. The client **108** makes IIOP requests **110** upon the cluster using tunneled IIOP. The problem is that this only works when the client communicates directly with the cluster and no proxy is involved. If, as is shown in **Figure 1**, a proxy **114** is involved then the situation becomes more complicated:

Proxy's that are hardware load balancers are unable to route requests to specific servers. They can only pick servers at random and then use some form of "stickiness" in the http header information. Client-side clustering relies on the ability to route to specific servers;

Proxy's generally perform Network Address Translation (NAT), requiring that cluster member (IOR's) advertise the address of the proxy rather than the actual external addresses of the cluster members. So whereas in the example shown in **Figure 1**, each of the cluster members C1 through C3 have a distinct address and port, the proxy can only advertise a single address and port to the client;

Unsigned applets can only open connections to the servers that can be accessed.

In order to provide efficient load-balancing and failover throughout the enterprise, there exists a need for a mechanism by which tunneled IIOP clients can work in clustered environment.

Summary:

In accordance with an embodiment of the present invention, a system and method for clustered tunneling of requests in application servers and other systems is provided. The system includes a client, a cluster having
5 a plurality of members, and a proxy. The cluster does not advertise the internal addresses of its members since clients might attempt to directly connect to that member. Instead, the client is configured to include a tunneling layer within which the proxy address is set. The advertised address of the cluster member is then set in a header at the client. During runtime,
10 the client connects to the proxy as before, but then uses its internal information to connect or communicate with the appropriate cluster member. In accordance with an embodiment the client sets a cookie which is then sent to the proxy. The proxy re-writes the cookie with the appropriate cluster member information and uses this information to tunnel the request to the
15 appropriate server.

Brief Description of the Figures:

Figure 1 shows an example of a clustered system that does not provide clustered tunneling of requests.

20 **Figure 2** shows an example of a clustered system that provides clustered tunneling of requests in accordance with an embodiment of the invention.

Figure 3 shows a flowchart of a process for handling clustered tunneling of requests in accordance with an embodiment of the invention.

25

Detailed Description:

In accordance with an embodiment of the present invention, a system and method for clustered tunneling of requests in application servers and other systems is provided. The system includes a client, a cluster having a plurality of members, and a proxy. The cluster does not advertise the internal addresses of its members since clients might attempt to directly connect to that member. Instead, the client is configured to include a tunneling layer within which the proxy address is set. The advertised address of the cluster member is then set in a header at the client. During runtime, the client connects to the proxy as before, but then uses its internal information to connect or communicate with the appropriate cluster member. In accordance with an embodiment the client sets a cookie which is then sent to the proxy. The proxy re-writes the cookie with the appropriate cluster member information and uses this information to tunnel the request to the appropriate server.

A principal problem with proxy's that are hardware load balancers is that they are unable to route requests to specific servers. They can only pick servers at random and then use some form of "stickiness" in the http header information. Client-side clustering relies on the ability to route to specific servers. In addition, proxy's generally perform Network Address Translation (NAT), requiring that cluster member (IOR's) advertise the address of the proxy rather than the actual external addresses of the cluster members. Also, unsigned applets can only open connections to the servers that can be accessed. There are three proposed solutions to this problem:

Using the proxy to perform client directed load-balancing and failover;

Performing server-side load-balancing and failover; and

-5-

Performing client-side load-balancing, with failover via HTTP 400.

Proxy load-balancing requires that actual addressing information is carried out-of-band and acted on by the proxy. Most load-balancers support URL-directed load balancing, so it is possible for a
5 tunneled client to embed addressing information in the URL, and for the load balancer to route the request based on this embedded information. This has the additional property that plugins would not be necessary in the application server. Instead, each application server is individually configured in the load
10 balancer or web server that receives the request.

With both server-side and client-side load balancing, selection of the target server is triggered when a request is sent to or received at the cluster. The proxy is not assumed to maintain connections to specific hosts, only that it can make repeated requests to a server it as already been
15 connected to. An initial request may be randomly routed to any cluster member, but will carry addressing information for the server to which the request is intended. The request is then communicated to the correct server. When routed the request is infected with appropriate session information that causes the proxy to continue sending requests to that server.
20 The addressing information can be carried in a number of ways, including:

Rewrite the URLs to encode addressing information. In this implementation, tunneled requests are rewritten based on the target server address and port information. For example, if the proxy is located at proxyhost:80, and the cluster member is located at cluster1:7001, then a
25 tunneled request may be written as:

`http://proxyhost:80/wls_internal/cluster1/7001/iiop`

-6-

or a variation thereof, depending on the particular implementation as to how the requests should be prepared.

An alternative means is to add addressing information in a custom HTTP header.

5 Having provided the address information, load balancing and failover may then be performed either by server-side load balancing, or client-side load balancing.

10 With server-side load balancing, each member of the cluster is configured to proxy requests based on the cluster information in the URL or header. Thus, even if a request fails over and a load balancer picks a new host, that request would still end up on the appropriate server. When the response is sent back to the client, it contains an appropriate cookie for the target server. If the server has already been accessed by the load balancer then subsequent requests will be routed correctly.

15 With client-side load balancing, the scenario is similar, but instead of routing the request, the server raises a HTTP 400 message. If the load balancer knew about that server, it would route accordingly, if not then the client would keep retrying until it reached another server. In this instance the servers do not need to proxy at all, they simply have to be able to set
20 appropriate cookie information for the client to use.

25 **Figure 2** shows an example of a clustered system that provides clustered tunneling of requests in accordance with an embodiment of the invention. As shown in **Figure 2**, the cluster **120** includes a number of cluster members or servers **122**, **124**, **126** as before. A client **128** communicates with the cluster via a proxy **130**. Each of the cluster members publishes an IOR including the specific addressing information for that member such as the name and port. However, the proxy cannot advertise the internal addresses of the server because the client would then attempt to connect to that server. Instead, in a tunneling layer of the client, only the

-7-

address of the proxy is set so that all requests then go to that proxy. The advertised address and port of the cluster member is set in a special header. The client sends **136** a cookie **134** to the proxy, wherein the cookie specifies the target IOR. When the client then connects **138** to the proxy using the proxy address, the proxy uses the information in the cookie to locate the correct server. In one embodiment the proxy rewrites **140** the address based on the IOR of the designated cluster member, as specified in the cookie.

During a failover situation the client gets the response that the destination server has failed. The proxy may attempt to forward the request to another server. The tunneling layer at the client will detect that the server has changed, and a failover code will then connect the client to another server, again via the proxy.

Figure 3 shows a flowchart of a process for handling clustered tunneling of requests in accordance with an embodiment of the invention. As shown in **Figure 3**, in step **150**, the client gets header information for the specific cluster member or server it is communicating with. In step **152**, the client writes a cookie to the proxy with this information. In step **154**, the proxy receives requests from the client destined for the cluster. In step **156** the proxy uses the information in the cookie to rewrite the requests with the appropriate IOR. In step **158**, the request is routed to the appropriate cluster member.

The present invention may be conveniently implemented using a conventional general purpose or a specialized digital computer or microprocessor programmed according to the teachings of the present disclosure. Appropriate software coding can readily be prepared by skilled programmers based on the teachings of the present disclosure, as will be apparent to those skilled in the software art.

In some embodiments, the present invention includes a computer program product which is a storage medium (media) having

-8-

instructions stored thereon/in which can be used to program a computer to perform any of the processes of the present invention. The storage medium can include, but is not limited to, any type of disk including floppy disks, optical discs, DVD, CD-ROMs, microdrive, and magneto-optical disks, ROMs, RAMs, EPROMs, EEPROMs, DRAMs, VRAMs, flash memory devices, magnetic or optical cards, nanosystems (including molecular memory ICs), or any type of media or device suitable for storing instructions and/or data.

The foregoing description of the present invention has been provided for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention to the precise forms disclosed. The embodiments were chosen and described in order to best explain the principles of the invention and its practical application, thereby enabling others skilled in the art to understand the invention for various embodiments and with various modifications that are suited to the particular use contemplated. It is intended that the scope of the invention be defined by the following claims and their equivalence.

Claims:

What is claimed is:

1. A system for clustered tunneling of requests with a proxy, comprising:
5 a cluster, including a plurality of cluster members;
a proxy for receiving requests from a client and routing requests to members in said cluster; and
wherein said proxy receives a cookie from a client, said cookie specifying information about the destination cluster member to be used with
10 said client, and then routes subsequent requests from said client according to said information.
2. The system of claim 1 wherein said cluster members advertise distinct IOR's identifying each cluster member.
15
3. The system of claim 2 wherein a client receives said IOR information about the cluster member and uses this information to create the cookie.
4. A method for clustered tunneling of requests with a proxy, comprising
20 the steps of:
providing a cluster including a plurality of cluster members;
receiving from a client at a proxy, a cookie specifying information about the destination cluster member to be used with said client; and
receiving subsequent requests at the proxy from the client and routing
25 those requests to the member in said cluster specified by said information.
5. The method of claim 4 wherein said cluster members advertise distinct IOR's identifying each cluster member.

-10-

6. The method of claim 5 wherein a client receives said IOR information about the cluster member and uses this information to create the cookie.

7. A proxy for clustered tunneling of requests between a client and a cluster, comprising:

a proxy for receiving requests from a client and routing requests to members in a cluster; and

wherein said proxy receives a cookie from a client, said cookie specifying information about the destination cluster member to be used with said client, and then routes subsequent requests from said client according to said information.

8. The proxy of claim 7 wherein said cluster members advertise distinct IOR's identifying each cluster member.

9. The proxy of claim 8 wherein the proxy uses the cookie to map client requests to said IOR information.

10. A client for use in clustered tunneling of requests between the client and a cluster, comprising:

a tunneling layer which receives IOR information about a cluster member and uses this information to create a cookie which is then sent to a proxy; and

wherein the tunneling layer communicates requests to the cluster member via the proxy.