

Internal distribution code:

- (A) [-] Publication in OJ
(B) [-] To Chairmen and Members
(C) [-] To Chairmen
(D) [X] No distribution

**Datasheet for the decision
of 13 November 2014**

Case Number: T 1338/10 - 3.5.06

Application Number: 04714154.4

Publication Number: 1599793

IPC: G06F9/00, G06F9/46, G06F15/16

Language of the proceedings: EN

Title of invention:
SYSTEM AND METHOD FOR SERVER LOAD BALANCING AND SERVER
AFFINITY

Applicant:
Oracle International Corporation

Headword:
Server affinity/ORACLE

Relevant legal provisions:
EPC 1973 Art. 56

Keyword:
Inventive step - (yes)

Decisions cited:

Catchword:



**Beschwerdekammern
Boards of Appeal
Chambres de recours**

European Patent Office
D-80298 MUNICH
GERMANY
Tel. +49 (0) 89 2399-0
Fax +49 (0) 89 2399-4465

Case Number: T 1338/10 - 3.5.06

D E C I S I O N
of Technical Board of Appeal 3.5.06
of 13 November 2014

Appellant: Oracle International Corporation
(Applicant) M/S 50P7
500 Oracle Parkway
Redwood Shores, CA 94065 (US)

Representative: Davies, Simon Robert
D Young & Co LLP
120 Holborn
London EC1N 2DY (GB)

Decision under appeal: **Decision of the Examining Division of the European Patent Office posted on 15 January 2010 refusing European patent application No. 04714154.4 pursuant to Article 97(2) EPC.**

Composition of the Board:

Chairman W. Sekretaruk
Members: S. Krischer
G. Zucka

Summary of Facts and Submissions

- I. The appeal is directed against the decision of the examining division, posted on 15 January 2010, to refuse the application 04714154 for lack of inventive step over document:

D1 WO 01/13228 A, 22 February 2001.
- II. A notice of appeal was received on 10 March 2010. The fee was received on 12 March 2010. A statement of the grounds of appeal was received on 20 May 2010. A claim set was filed with the grounds. Oral proceedings were requested.
- III. In its summons to oral proceedings, the board gave reasons for its preliminary opinion that independent method claim 10 lacked an inventive step over D1.
- IV. In a letter dated 10 October 2014, the appellant filed claim sets of three auxiliary requests.
- V. Oral proceedings were held on 13 November 2014 during which the appellant filed a claim set of an amended main request. At the end of the oral proceedings, the board announced its decision.
- VI. The appellant requests that the decision be set aside and a patent be granted on the basis of claims 1-18 of the main request filed during oral proceedings on 13 November 2014, or claims 1-4 of the first auxiliary request or claims 1-18 of the second or third auxiliary request filed on 10 October 2014.
The further text is: description pages 1, 19 filed on 13 November 2014 during oral proceedings; pages 2-5,

7-18 as originally filed; pages 6, 6a filed on 10 June 2009; drawing sheets 1-3 as originally filed.

VII. Claim 1 of the main request reads as follows:

"1. A system for server load balancing that includes server affinity, comprising: a cluster that includes a plurality of server instances providing services, and wherein each of the services provides a plurality of method calls;

 a load balancing and affinity processor that assigns server instances from said cluster to service client requests in the form of method calls from external clients;

 a client-side stub on an external client obtained for a service, wherein the client-side stub attempts to choose a server instance to which the external client is already connected, and the client-side stub continues to use the same server instance and the same connection for method calls of that service;

 wherein if the server instance becomes unavailable, the stub fails over to a server instance to which the client is already connected and which provides said service; and

 wherein the cluster is adapted to use a load balancing algorithm that includes server affinity to govern connections between external clients and server instances and wherein the load balancing algorithm is overridden by a user-configured load balancing algorithm for the service maintained in the client-side stub."

VIII. Independent method claim 10 of the main request reads as follows (additions or modifications with respect to

claim 10 of the refused main request are marked in *italics*; deletions are ~~struck through~~):

"10. A method for server load balancing that includes server affinity, comprising the steps of:

 providing a plurality of server instances as a cluster providing services, and wherein each of the services provides a plurality of method calls;

 assigning servers ~~instances~~ from said cluster to service client requests in the form of method calls from *external* clients;

 wherein said step of assigning includes using a client-side stub *on an external client* obtained for a service, wherein the client-side stub attempts to choose a server instance to which *the external client* is already connected, and *the client-side stub* continues to use the same server instance and same connection for method calls for that service;

 wherein if the server instance becomes unavailable, the stubs fails over ~~if possible~~ to a server instance to which the *external* client is already connected *and which provides said service*; and

 wherein the cluster uses a load balancing algorithm that includes server affinity to govern connections between external clients and server instances and wherein the load balancing algorithm is overridden by a *user-configured* load balancing algorithm for the service maintained in the client-side stub."

IX. In view of the board's decision, the claim text of the auxiliary requests is irrelevant.

Reasons for the Decision

1. *Overview of the invention*

The application relates to a method of distributing method calls from external client computers via client-side stubs to server instances (i.e. server programs running on a server computer). The server instances may be RMI (remote method invocation) objects like JMS (Java message service) or EJB (enterprise Java bean) interfaces (original description paragraph [29], second sentence). For method calls from clients to services *not* configured for server affinity, a conventional load balancing algorithm (LBA) like round-robin is used for both internal and external connections ([22], paragraph 2). For method calls from external clients to services configured for server affinity, *no* LBA is used, but server affinity which means that the client-side stub attempts to choose a server instance to which it is already connected ([21], second and third sentence; [22], paragraph 5). Furthermore, if a server instance becomes unavailable, then the client-side stub fails over to another already connected server instance ([21], last sentence; [32]). And an LBA which an administrator configures for a service overrides the default LBA for the cluster ([23], fourth sentence).

2. *Overview of the decision*

The board holds that the claims of the main request are inventive and allowable.

3. *Remark*

Like the appealed decision (2.1), the board mainly focuses on independent method *claim 10* instead of system claim 1. However, all the reasoning applies accordingly to claim 1.

4. *Original disclosure of the main request*

4.1 The examining division did not raise any objections under Article 123(2) EPC in its decision and the board concurs that there was no reason to do so with respect to the claims as refused.

4.2 Besides some trivial clarifications, there are two main amendments in independent method claim 10 of the current main request when compared with claim 10 of the refused main request:

a) everywhere in the claim, it has been specified that the clients are *external*: see [21], second sentence;

b) the wording "if possible" was deleted in the failover step: the board is of the opinion that the wording "if possible" in [21], fifth sentence means "if the external client is connected to a server which provides the service" (see the example in [32]).

4.3 Thus the board finds that amended claim 10 of the current main request satisfies the requirements of Article 123(2) EPC.

4.4 System claim 1 has been amended accordingly and is therefore also in compliance with Article 123(2) EPC.

5. *Inventiveness of claim 10 of the main request*

5.1 The board considers D1 to be the closest prior art, as it was also considered to be in the appealed decision. This was not contested by the appellant.

5.2 The grounds of appeal (5.) have designated by the letters a)-h) the features of claim 10 which are contested to be disclosed in D1. With respect to claim 10 of the (current) main request, the board has the following opinions on the disputed features (using the same letters a)-h) as in the grounds):

- a) "providing a plurality of server instances as a cluster providing services, and wherein each of the services provides a plurality of method calls":

According to the grounds, D1 does not teach a cluster of server *instances*, but a cluster of server *computers*. Whereas server instances are software components (i.e. running programs), server computers are separate physical machines, e.g. application servers (D1, page 5, lines 30-31; figures 2A-2C).

The board is not convinced by this argument: Since every functioning server computer has at least one server program (i.e. server instance) running on it, D1 discloses a cluster of server instances by disclosing a cluster of application servers.

- b) "assigning server instances from said cluster to service client requests in the form of method calls from external clients": see d).

- c) "wherein said step of assigning includes using a client-side stub on an external client obtained for a service":

The board agrees with the grounds that neither D1 nor general knowledge disclose that IIOP stubs are obtained *for a service*. Even if so, D1 (in particular page 15, line 27 to page 16, line 26) does not disclose that IIOP stubs perform the sticky load balancing.

In its summons to oral proceedings, the board was of the preliminary opinion that D1 disclosed a client-side component (web server plug-in) which performs the load balancing (D1, page 12, line 1 to page 13, line 12). The situation was said the same as in [33] and figure 3 of the application: a server had the role of a client to a cluster of application servers. See D1, page 12, line 5: "... the 'client' of an application server is a web server." and lines 35-36: "... a web server client 300 with a web server plug-in 302 comprising a load balancer component 304 which distributes requests across an application server cluster". See also the application, [33], lines 4-5: "1. A JSP on MS4 obtains a stub for Service B. MS4 in this instance is acting as a client to MS1, MS2 and MS3." Note that MS probably means "Managed Server" (see line 8 of [33]) and that Java server pages (JSP) are usually executed on a web server.

However, during oral proceedings the appellant explained that the application and D1 differentiate between *external* and *internal* clients: External (or "normal" or "classical") clients are computers at the user's side, i.e. computers which are no servers (e.g. the client in figures 1, 2

and in paragraphs [31], [32] of the application; or client computer 250 in figure 4 of D1), whereas internal clients are themselves servers (e.g. managed server MS4 in figure 3 and in paragraph [33] of the application; or web server client 250 in figure 4 of D1). The claim only related to external clients as shown in figure 2 and not to internal clients as in figure 3. The appellant then filed amended claims for the main request which explicitly specify "external clients" as requesting services and having client-side stubs performing the assignments of server instances and the failover handling.

The board agrees that this differentiation is important. Therefore, one cannot combine the embodiments of an *internal* web server plug-in performing the load balancing (D1, page 12, line 1 to page 13, line 12) with the passage in D1 about "sticky" load balancing by an *external* client (page 15, line 27 to page 16, line 26). Since the latter passage in D1 (about sticky load balancing) relates to *server-side* load balancing (lines 35-38: "depending on the outcome of the load balancing decisions described above"; the preceding section is about *server-side* load balancing), it does not disclose a (client-side) stub on an *external* client *performing load balancing*.

Furthermore, there is no disclosure about a stub *obtained for a service* in the above mentioned passage of D1. There might be something similar to a stub on page 16 (lines 14-15: "... the client computer(s) may instead be operable to maintain information regarding sticky requests so that requests are sent directly to the correct

application server."), but *not for a specific service and not doing load balancing when no connection to a server instance exists.*

Accordingly, D1 neither discloses a stub on an *external client performing load balancing*, nor a stub *obtained for a service.*

Furthermore, in the summons the board expressed its preliminary opinion that D1 disclosed that the *sticky* load balancing is performed at the client side (see again D1, page 16, lines 14-15: "... the client computer(s) may instead be operable to maintain information regarding sticky requests so that *requests are sent directly to the correct application server.*", emphasis added). It was said that it followed from the text in italics that a client-side component (stub) performs the sticky load balancing.

However, the above cited passage does not relate to the *initial* request for an assignment of a server instance for a service (see e) below), but to "continuing to use the same server instance" (see f) below).

- d) The board agrees with the grounds that in D1 the embodiment of figure 2C (with a client computer as the *external* client to the application server cluster) and that of figures 2A and 2B (with a web server as the *internal* client to the application server cluster) are different and cannot be cited together in a problem-solution approach.

Therefore, the board will not cite features from the embodiments of figures 2A and 2B in its analysis.

- e) "wherein the client-side stub attempts to choose a server instance to which the external client is already connected":

The board recognises in feature e) the same difference with respect to D1 which the decision (2.1.3) recognised in the failover situation, namely *to choose, if possible, a server instance to which it is already connected*, but here this is done in the context of servicing an *initial* request for a service from a client. Thus, the board holds that D1 does not disclose to choose a server instance to which the client is already connected in order to serve an initial request for a service from a client. In contrast hereto, in D1 this initial assignment to a server instance is done by (server-side) *load balancing* (page 15, lines 35-38).

- f) "and the client-side stub continues to use the same server instance and same connection for method calls for that service":

According to the grounds, the client computer in the passage cited by the decision (D1, page 16, lines 14-15) appears to correspond to the client computer 100 in figure 2A, and not to the web server 104.

As said above, the board agrees with the appellant that web server 104 can neither be taken as the (external) client computer of the cited passage in D1 about sticky load balancing, nor as the external client in claim 10. Furthermore, the board remarks that the client computer in the cited passage does not relate to the client computer 100 of figure 2A, but to client computer 114 of figure 2C as explained in d).

g) The feature of replica-awareness is not present in current claim 10.

h) "wherein if the server instance becomes unavailable, the stub fails over to a server instance to which the external client is already connected and which provides said service":

The board agrees with the grounds that the polling threads have nothing to do with IIOP. But as before, the board does not use IIOP in its analysis.

The board considers the passage about polling threads (page 18, lines 32-33) and its surrounding section named "Request failover" (page 17, line 15 to page 18, line 25) to disclose a client-side stub which fails over to a server instance of another server. However, according to page 17, line 16 the passage relates to an *internal* client ("such as a web server"), and not to an external client. Furthermore, there is no disclosure that a server instance to which the client is already connected is used for failover.

5.3 It follows that the only embodiments in D1 which can be taken for comparing with claim 10 are the passage about sticky load balancing by an external client (page 15, line 27 to page 16, line 26) and the one about failover (page 17, line 15 to page 18, line 25).

5.4 Therefore, current claim 10 *differs* from D1 in that for the *initial* request of an external client for a service and during failover, the client-side stub chooses, if possible, a server instance *to which it is already connected*, whereas D1 chooses a server determined by a (server-side) load balancing for the initial request

(page 15, lines 35-38). For failover no procedure how to choose a server instance is disclosed in D1.

Furthermore as explained above under c), D1 neither discloses a stub on an external client *performing load balancing*, nor a stub *obtained for a service*.

As to the similarities between the claim and D1, in both of them *further* requests from the same client for the same service stick to the chosen server instance.

5.5 The objective technical problem as formulated in the decision (2.1.4), i.e. how to minimise the number of connections, does not seem to be appropriate for the current claim, since it implies already the solution in it and it does not cover all the differences.

5.6 The board formulates the technical problem as how to reduce the time to serve an initial request from a client for a service and to react in a failure situation.

5.7 The solution is to reuse an existing connection of a client to a server instance for any initial request for a service from a client, or for requests when the previously used server instance failed. To achieve this in D1, the method disclosed on page 15, line 27 to page 16, line 26 would have to be completely rebuilt: replace the *server-side load balancing* by a *client-side "connection-first"* strategy (called "server affinity" in the application) for the initial request; do this by a stub *obtained for that service*; for failover also select this connection-first strategy.

The concept of a component similar to a stub disclosed in another embodiment in D1 (namely the web server plug-in 242 in figure 4) is not directly applicable for solving the problem, since this is

situated in a very different context of a two-level architecture (figure 2A) with a web server 104 placed between the (external) client computer 100 and the application servers 108A and 108B. The claim explicitly specifies a one-level architecture with a direct connection between the external client and the server instances (i.e. the application servers). Furthermore, also this web server plug-in does not disclose a connection-first strategy, but mere load balancing. The only hint to a connection-first strategy in D1 would be the continued usage of the server instance previously selected by the server-side load balancing in the sticky load balancing example. However, as argued by the appellant during oral proceedings, the reason for that continued usage seems to be to avoid the migration of the data structure "ShopCart" from one server instance to another, and not to minimise open connections (i.e. sockets) or to avoid the time to open a connection.

- 5.8 Therefore, claim 10 of the main request is inventive in the sense of Article 56 EPC 1973.

6. Corresponding system claim 1 of the main request is consequently also inventive.

Order

For these reasons it is decided that:

- 1) The decision under appeal is set aside.

- 2) The case is remitted to the examining division, with the order to grant a European patent on the basis of claims 1-18 of the main request filed on 13 November 2014 during oral proceedings; description pages 1, 19 filed on 13 November 2014 during oral proceedings; pages 2-5, 7-18 as originally filed; pages 6, 6a filed on 10 June 2009; drawing sheets 1-3 as originally filed.

The Registrar:

The Chairman:



B. Atienza Vivancos

W. Sekretaruk

Decision electronically authenticated