

Internal distribution code:

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

**Datasheet for the decision
of 19 October 2012**

Case Number: T 1797/08 - 3.5.06

Application Number: 03760238.0

Publication Number: 1573520

IPC: G06F 9/54

Language of the proceedings: EN

Title of invention:

Method and system for simplifying distributed server management

Applicant:

Bladelogic, Inc.

Headword:

Server management/BLADELOGIC

Relevant legal provisions (EPC 1973):

EPC Art. 83, 56

Keyword:

"Sufficient disclosure - all requests (yes)"

"Inventive step - main and first auxiliary request (no)"

Decisions cited:

-

Catchword:

-



Case Number: T 1797/08 - 3.5.06

DECISION
of the Technical Board of Appeal 3.5.06
of 19 October 2012

Appellant: Bladelogic, Inc.
(Applicant) 100 Crosby Drive
Bedford, MA 01730 (US)

Representative: Scaddan, Gareth Casey
Murgitroyd & Company
Scotland House
165-169 Scotland Street
Glasgow G5 8PL (GB)

Decision under appeal: Decision of the Examining Division of the
European Patent Office posted 18 February 2008
refusing European patent application
No. 03760238.0 pursuant to Article 97(2) EPC.

Composition of the Board:

Chairman: D. H. Rees
Members: S. Krischer
C. Heath

Summary of Facts and Submissions

I. The appeal is directed against the decision of the examining division, posted on 18 February 2008, to refuse the application 03760238. The decision was according to the state of the file, referring to a summons to oral proceedings dated 7 November 2007. The claimed subject-matter was considered to be unclear and further to lack an inventive step over the following documents:

D1 T. Rightnour: "Clusterit version 2.0 (Archive)", pages 1-30, 2001, XP2328731, extract of the tar archive retrieved from:

<http://www.garbled.net/download/clusterit-2.0.tar.gz>

D4 Object Management Group (OMG): "The Common Object Request Broker: Architecture and Specification. Revision 2.0", pages 1-29, July 1995, XP2133788.

II. A notice of appeal was received on 7 April 2008. The fee was paid on the same day. A statement of the grounds of appeal was received on 27 June 2008. Claim sets of a main and two auxiliary requests were filed. Oral proceedings were requested.

III. In its summons to oral proceedings, the board considered that the term "abstract system call", meaning a system call applicable to different operating systems, was clear but well-known. To demonstrate this, the board introduced the following document from its own knowledge:

D5 D. Bovet et al.: "Understanding the Linux Kernel", first edition, O'Reilly (publisher), ISBN 0-596-00002-2, chapter "8. System Calls", pages 217-230, October 2000.

The board however raised a clarity objection concerning a contradiction between the newly introduced phrase "at least one server" and "executing, on one of a plurality of servers" in claim 1 of all requests. In addition, the question was raised whether there was sufficient disclosure in the application with respect to the step of identifying by the virtual server client (30) a target server in claim 1 of all requests. Given these objections, inventive step was not fully analysed, but it was noted that, on the basis of document D5, abstract system calls which are converted into operating system-specific system calls would appear to be a matter of common general knowledge.

- IV. In a letter dated 19 September 2012, the appellant filed claim sets for an amended main request and four auxiliary requests. The former requests were maintained as auxiliary requests 5-7.
- V. Oral proceedings were held on 19 October 2012 during which the appellant filed an amended main request and two auxiliary requests, and withdrew all other requests.
- VI. At the end of the oral proceedings, the board announced its decision.
- VII. The appellant requests to set the decision aside and to grant a patent on the basis of a main request (claims 1-59), a first or a second auxiliary request (both claims 1-57), all filed during oral proceedings.

The further text on file is: description pages 1, 3-38 as filed with letter dated 11 December 2006; pages 2, 2a, 39 as filed with letter dated 20 August 2007; drawing sheets 1-14 as published.

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

"1. A method for receiving an abstract system call from a software application program (25A, 25B, 25C) and executing, on one of a plurality of servers (15A, 15B, 15C, 15D), an operating system-specific system call, the plurality of servers (15A, 15B, 15C, 15D) being accessible to the software application program via a virtual server implemented by a virtual server client (30) and a plurality of virtual server agents (35A, 35B, 35C, 35D) each running on a respective one of the plurality of servers (15A, 15B, 15C, 15D), the method comprising the steps of:

(a) receiving (40), by the virtual server client (30) from a software application program (25A, 25B, 25C), an abstract system call that requests a service from an operating system of the one of the plurality of servers (15A, 15B, 15C, 15D), the abstract system call generated with indifference to the operating system used by the server; and

(b) instantiating (42) in a thread-safe manner the abstract system call by:

identifying, by the virtual server client (30), a target server (15A, 15B, 15C, 15D) to receive the abstract system call and a corresponding virtual server agent (35A, 35B, 35C, 35D) associated with the target server;

transmitting (46) the abstract system call to the identified agent (35A, 35B, 35C, 35D)

for translation into an operating system-specific system call for execution on the target server (15A, 15B, 15C, 15D), the operating system-specific system call being specific to the particular operating system used by the target server; and

receiving (50) execution results from the agent."

IX. Claim 1 of the first auxiliary request differs from claim 1 of the main request in that the end of the preamble reads:

"1. ... each running on a respective one of the plurality of servers (15A, 15B, 15C, 15D), wherein at least two of the plurality of servers (15A, 15B, 15C, 15D) have different operating systems, the method comprising the steps of:

X. Claim 1 of the second auxiliary request reads as follows:

"1. A method for receiving an abstract system call from a software application program (25A, 25B, 25C) and executing, on one of a plurality of servers (15A, 15B, 15C, 15D), an operating system-specific system call, the plurality of servers (15A, 15B, 15C, 15D) being accessible to the software application program via a virtual server implemented by a virtual server client (30) and a plurality of virtual server agents (35A, 35B, 35C, 35D) each running on a respective one of the plurality of servers (15A, 15B, 15C, 15D);

(a) receiving (40), by the virtual server client (30) from the software application program (25A, 25B, 25C), an abstract system call that requests a service

from an operating system of the one of the plurality of servers (15A, 15B, 15C, 15D), the abstract system call generated with indifference to the operating system used by the server; and

(b) instantiating (42) in a thread-safe manner the abstract system call by:

identifying, by the virtual server client (30), a target server (15A, 15B, 15C, 15D) to receive the abstract system call and a corresponding virtual server agent (35A, 35B, 35C, 35D) associated with the target server;

transmitting (46) the abstract system call to the identified agent (35A, 35B, 35C, 35D) for translation into an operating system-specific system call for execution on the target server (15A, 15B, 15C, 15D), the operating system-specific system call being specific to the particular operating system used by the target server;

identifying (55), by the virtual server agent (35A), a user (10) of the software application program;

mapping (56) the identified user to an associated local user of the target server (15A, 15B, 15C, 15D);

impersonating (58) the identified user (10) as the mapped local user on the target server (15A, 15B, 15C, 15D) to provide the identified user (10) with permissions associated with the mapped local user; and

receiving (50) execution results from the agents."

Each of the requests also comprises a corresponding independent claim to a "virtual server".

Reasons for the Decision

1. *Admissibility of the requests*

The main request and the two auxiliary requests were filed after the grounds of appeal and hence according to Article 13(1) of the Rules of Procedure of the Boards of Appeal the board has discretion whether or not to admit them into the proceedings. However, the amendments were made in order to overcome the objection raised in the oral proceedings, that the new requests filed on 19 September 2012 after reception of the summons covered the execution of an abstract system call on *more than one* server. All claim sets filed previously, including the originally filed one and those filed with the grounds, referred to one of a plurality of servers in the preamble. Executing a call on one of a plurality of servers or on a plurality of servers makes a significant difference: Whereas the first might relate to load balancing, the second might relate to managing a server farm. As mentioned in the summons (6.5), original description page 12, lines 13-16 disclose the execution on one of a plurality of servers. The requests filed during oral proceedings reintroduced this feature consistently in claim 1, thereby also remedying the clarity objection raised in section 6.5 of the summons.

As to the other features (identifying, mapping and impersonating a user) added to the second auxiliary request, they are disclosed in original claim 15.

However, this claim further contains encrypting and decrypting of the abstract system call. The appellant argued during oral proceedings that these encrypting/decrypting steps were not necessary for implementing the remaining steps of original claim 15, so that their omission would not represent an intermediate generalisation. The further omitted step of authorizing the abstract system call was said to be included in the impersonating step ("to provide the identified users with permissions associated with the mapped local user"). The board agrees. This selection of claimed features therefore does not add subject-matter to the application as filed.

Therefore, the board admitted the newly filed requests into the procedure.

2. *Sufficient disclosure*

During oral proceedings, the board repeated the objections raised in the summons concerning insufficient disclosure of the "virtual server client" (summons 7.2-7.4). The appellant argued that a virtual server client may be implemented by maintaining a list of servers and its capabilities. A skilled person would know how to program a virtual server client that distributes an abstract system call to a target server at least according to some straightforward strategy. For example, the virtual server client could simply select the first server which has the necessary capabilities. Considering that workload balancing is a standard topic in computer science undergraduate curricula, the board accepts this argument, even though the application itself is silent on how choice of an

unspecified server might take place. This objection is therefore not maintained.

3. *Inventive step*

3.1 Main request

Claim 1 relates to a method of receiving and executing a so-called abstract system call (i.e. a system call which is applicable to different operating systems, see summons 6.3), sent from an application program via a virtual server client to one of a plurality of servers.

In its summons, the examining division chose D1 as closest prior art. This has not been contested by the appellant and the board agrees.

Document D1 discloses the distribution of a shell command to a server automatically selected from a cluster (page 7, last paragraph "Job Scheduled Shell").

The first difference between claim 1 and D1 is that the claim relates to system calls, and not to shell commands as in D1. A shell command usually itself invokes system calls and can therefore be said to be one level above system calls. According to D5 (page 217, 8.1, paragraph 2), a system call is usually invoked via a wrapper routine, e.g. from the libc standard C library in case of Linux operating systems. The board notes that D5 is a document representing common general knowledge, being an extract of a textbook on a topic of general interest in the field.

The second difference between the claim and D1 is the operating system independence of the system calls

("abstract system calls"). The two differences do not interact with each other. Operating system dependence and independence is a concept applying equally well to both shell commands and system calls.

As to the first difference, a skilled person would obviously wish to execute system calls remotely, as D1 does for shell commands or as the well-known technique named "remote procedure call" (RPC) does for normal subroutines.

As to the second difference, operating system independent system calls are a matter of common general knowledge in the field, as evidenced in D5 (page 217, 8.1, paragraphs 1-5): i.e. the POSIX API standard which defines wrapper routines that issue system calls on POSIX-compliant operating systems.

The board considers it obvious for a skilled person to use POSIX-compliant APIs at the servers side for implementing the remote execution of operating system independent system calls. Such an API implemented in the program library "libc" of D5 (page 217, 8.1, paragraph 2) corresponds to the virtual server agent (35A-35D) of the claim. The selection of the executing server at the client side by the "Job Scheduled Shell" of D1 (page 7, last paragraph) represents herein a virtual server client (30) in the sense of the claim.

As to the appellant's doubts that a skilled person would not combine D5 with D1, the board notes the paragraph entitled "Heterogeneous cluster makeup" (D1, middle of page 7) which discloses the wish for operating system independence in the framework of D1.

The skilled person would clearly be motivated to apply common general knowledge to realise that wish.

Thus, claim 1 of the main request is not inventive, in violation of Article 56 EPC.

3.2 First auxiliary request

Claim 1 of the first auxiliary request does not add an inventive step to claim 1 of the main request by explicitly stating that at least two of the servers have different operating systems. Operating system independence is already one of the core aspects of claim 1 of the main request, even if the latter does not exclude that the operating systems of all servers might be the same.

Thus, claim 1 of the first auxiliary request is not inventive, in violation of Article 56 EPC.

3.3 Second auxiliary request

Claim 1 of the second auxiliary differs from claim 1 of the main request by adding the steps of identifying, mapping and impersonating a user of the application program. These features are part of original claim 15 to which the objection of lacking inventive step was raised in section 4.7 of the summons of the examining division, i.e. the reasons for its decision according to the state of the file. Therein, it is argued that it is known that "rsh", the default command used in D1 to execute a shell command on the server, checks a file named "~/.k5login" to see whether the user can be authenticated to one of the principals named in that file, and if not whether "the authenticated principal

name of the user can be mapped to the local account name using specific mapping rules".

The appellant argued that said file ("~/k5login") would not imply a mapping of the user. With no documents at hand about what "rsh" does with said file, the board cannot decide whether the additional features of the claim do indeed belong to usual behaviour of "rsh". Furthermore, "rsh" is a tool for remotely executing shell commands, but - as said above - the application is about system calls.

The board notes that the newly added steps of identifying, mapping and impersonating a user are not independent from the rest of the claim, but interact with them and allow the execution of abstract system calls on the server with the permissions of the mapped user. This solves for example the problem of having different conventions for user names in different operating systems of the servers or compared with the client computer.

The board considers the argumentation in section 4.7 of the summons of the examining division concerning original dependent claim 15 not sufficient to prove a lack of inventive step of present claim 1. Therefore, the case is remitted to the first instance so that the examining division will have the possibility to procure evidence whether "rsh" really executes the newly added steps of the second auxiliary request. If such evidence turns out to be available, the examining division will also have to take into account in its considerations the differences between shell commands and system calls.

4. *Adaptations*

The board notes that in the event that the examining division comes to the conclusion that the claims of the second auxiliary request are inventive, there are still a number of amendments which will have to be made before a patent can be granted: e.g. the description needs to be adapted before a patent can be granted (see for example the serial numbers in section [1]). There are also clarity problems in the claims of the second auxiliary request: e.g. the last word of claim 1 should read "agent" instead of "agents". Furthermore, with respect to the virtual server claims, the dependencies have not been correctly adapted. For example, virtual server claim 25 refers back to method claim 23.

Order

For these reasons it is decided that:

1. The decision under appeal is set aside.
2. The application is remitted to the department of first instance for further prosecution on the basis of the second auxiliary request filed during oral proceedings.

The Registrar:

The Chairman:

B. Atienza Vivancos

D. H. Rees