Title: METHOD AND CONFIGURATION AGENT FOR PROVISIONING AN APPLICATION ON A SERVER

Abstract: The invention relates to a method and a configuration agent for provisioning an application (4) on a server (1). The server (1) hosts one or more applications (4) which provide services to clients (2) over a network (3) and at least one application-specific configuration agent (6). An administration means (7) is connected to all servers (1). To provision an application (4) on a server (1), a control command is transmitted from the administration means (7) to the server (1) and received by said configuration agent (6). The configuration agent (6) determines one or more configuration actions concerning the application (4) and/or its runtime environment (5) on the basis of the received control command and executes the configuration actions accordingly.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OA (BE, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— with international search report

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.
Description of the Invention

METHOD AND CONFIGURATION AGENT FOR PROVISIONING AN APPLICATION ON A SERVER

The invention relates to a method for provisioning an application on a server in a computer arrangement, the computer arrangement comprising a plurality of servers each hosting one or more applications which provide services to clients over a network, and more particularly to the use of an application-specific configuration agent.

Computer arrangements comprising a plurality of servers, often called server farms, are well known to provide a variety of different services to clients over a network. For example, clients can request the transmission of a file which is stored on the server or on a storage device associated with the server. Another example is a web service, where web pages are transmitted upon request. Also known are database services, where, in a similar fashion, data sets of a database are transmitted or processed upon request. Usually each individual service is provided by an application program running on a server. Depending on requirements or capacity, a server can host a single or also several instances of one or of a number of different application programs.

In general, a trend towards larger server farms is becoming apparent. Often a network service provider runs as many as hundreds of servers to be able to handle all incoming requests. Considering this growing sizes of modern server farms, there is an increasing demand to automate administrative tasks, for example to automatically allocate the available resources to different types of services. The
approach to automate administrative tasks is known as autonomous computing (AC). A central element in an AC environment is an administration means to perform administrative tasks which is sometimes referred to as a decision engine (DE). The decision engine is capable of providing and configuring new servers and eventually also to start applications on the servers afterwards.

Different approaches for the provisioning process are known. Often, an approach known as "bare metal provisioning" is chosen, where for each new task (different customer, different application, etc.) a whole boot image is transferred to a server over a network, saved on a local storage device (hard-disk) of the server and then the server is rebooted using the new boot image. In a deviation of this process, the new boot image is transferred over the network but not saved on the local storage device. This process, called "net-boot" might be slightly faster since no local copy of the boot image has to be made.

Both methods require a system shutdown and a reboot, which are rather time consuming processes that typically take several minutes in the best case. Another approach is to remotely start or stop instances of an application on a server by the administration means. This method is faster than the bare metal provisioning, but requires to include detailed information on the starting and stopping procedures of all applications in the administration means. Since the administration means is usually maintained by the provider who runs the server farm and the applications are userspecific, this approach is error-prone and in practice not feasible.
It is therefore an objective of the present invention to provide a method for provisioning an application on a server by an administration means where the method is fast and flexible and where the administration means is able to work with a diversity of applications. It is another object of the invention to present a configuration agent suitable to carry out the method.

The above object is accomplished by a method for provisioning an application on a server according to claim 1 and a configuration agent according to claim 8.

The basic idea behind the present invention is to provide at least one application-specific configuration agent for each server and to transmit control commands concerning an application to be provisioned on the server from the administration means to this configuration agent. After receiving the control command, the configuration agent determines one or more configuration actions concerning the application and/or its runtime environment on the basis of the received control command and executes the configuration actions by the configuration agent.

The method allows for rapid application provisioning without a server shutdown or reboot. The configuration agent encapsulates application specific knowledge (configuration actions) so that a administration means can simply send general directives (control commands). This allows the central administration means to be flexible and general.

The administration means is also able to work with a diversity of applications since the configuration agent isolates the administration means from application specific
details. Therefore, new applications or updates of applications can easily be controlled by the administration means. Application specific details have of course to be integrated into the configuration agent, but the less complex structure of the configuration agent compared to the administration means simplifies this task. Another advantage is that the configuration agent can be provided by the user of an application, who has detailed knowledge on how to provision the application, whereas the administration means would have to be maintained by the provider of the computer arrangement (server farm), who does not necessarily have such a detailed knowledge on that particular application.

A further advantage is that the configuration means does also care about the runtime environment of the application. The runtime environment comprises all resources needed and conditions to be fulfilled for a successful execution of the application. According to one embodiment of the invention, configuration actions concerning the runtime environment of the application comprise one or more of the following actions:

- changing settings of the server;
- establishing network connections;
- mounting file systems on the server;
- stopping other applications that could conflict with the application to be provisioned;
- starting other applications that are needed by the application to be provisioned.

These actions usually require knowledge of the local environment, for example information about the hard- and software resources of the server the application is executed on. Therefore, configuration actions concerning the runtime
environment of the application can be performed much easier by the local configuration agent. In some cases, an external access to resources of the server might not possible at all.

Other features which are considered as characteristic for the invention or which describe advantageous embodiments of the present invention are set forth in the appending claims.

The above and other objects, features and advantages of the present invention will become apparent from the following description in conjunction with the accompanying drawings.

In the drawings

Figure 1 shows a schematic representation of an embodiment of a computer arrangement which makes use of the invention,

Figure 2 shows a flow chart diagram of an embodiment of the method according to the invention.

Figure 1 shows several servers 1 (1a, 1b) which are set up to communicate to clients 2 via a network 3. Each server 1 comprises an application 4 (4a, 4b), its runtime environment 5 (5a, 5) and a configuration agent 6 (6a, 6b). The configuration agents 6 are connected to an administration means 7 via control connections 8.

The configuration shown in Figure 1 is typical for a server farm where a provider runs a plurality of servers. Often, blade servers, named after their geometric outline, are used in server farms. In the embodiment shown, each server 1 hosts just one application 4. However, the invention is not
restricted to a situation where each server 1 hosts one application 4 only. In a case where more than one applications 4 or instances of an application 4 are hosted on a server 1, either one configuration agent 6 would be provided for each application 4 or one common configuration agent 6 would be provided which would be responsible for more than one application 4.

Each configuration agent 6 is set up to provision the application 4 or the applications 4 it is specific to. The process of provisioning comprises configuring and controlling the application 4 itself. One possible way to configure and control an application 4 is to change the settings of the application 4, either via a local interface that the application provides (e.g. RMI - Remote Method Invocation) or via a configuration file used by the application 4. Another way of controlling is to stop or start or restart an instance of the application 4. Any control options that the application 4 provides can be used. These could for example include standardized interfaces like Active-X as well as scripting or macro options.

According to the invention, the process of provisioning application 4 also comprises configuring and controlling the runtime environment of the application 4. Despite the ability to configure the applications 4 itself, the configuration agent 6 is therefore able to manipulate the runtime environment 6 of the application 4. The runtime environment 6 includes all resources needed and conditions to be fulfilled for a successful execution of the application 4. A certain application 4 might, for example, need specific server settings to be executed, like a specific network address, a certain file system or a assured minimal amount of memory.
The configuration agent is therefore set up to be able to configure the runtime environment 6 accordingly, e.g. by selecting the desired network address, by mounting the needed file system or by reserving the required amount of memory. This can be achieved by running appropriate operating system commands or scripts. Another important part of the runtime environment 6 of the application 4 are other applications or processes or background-processes (demons) running on the server 1. The configuration agent 4 is thus also able to start other applications, processes or demons that are needed by the application 4 or to stop applications, processes or demons that could conflict with the application 4.

The method according to the invention and hence the operation of the configuration agent 6 will now be described in more detail in connection with Figure 2.

In step A of Figure 2, a control command concerning an application 4 is transmitted from the administration means 7 to the configuration agent 6 on the server 1 that hosts the application 4 to be provisioned. For the control command to be simple and general, it is beneficial to agree on a uniform format for the control commands, i.e. to define a uniform interface specification between administration means 7 and configuration agent 6. It is particularly favorable to abstain from application specific information as much as possible, so that, for instance, commands like "start xy" or "stop xy", where "xy" denotes the name of the application 4 to be started or stopped, are sufficient, even if starting or stopping the application 4 requires many actions to be performed on the server 1. The control command could be in form of a mark-up language based transaction, in particular in form of an XML-based (eXtensible Markup Language) transaction. The control connections 8 used to transmit the
control commands could form an independent network for security reasons, or the same network 3 that connects the server 1 and the clients 2 could be used.

In step B, the control command is received by the configuration agent 6 on the server 1 that hosts the application 4 to be provisioned. After having checked its syntax and validity, the configuration agent 6 analyzes the control command and determines appropriate configuration actions to carry out the control command. In a first approach, the control commands could be taken from a table which is stored in the configuration agent 6. For every possible control command, a list of configuration actions has to be given in that table. The table could be in form of a simple data structure or a database.

In some cases it could be important that a certain order, by which the configuration actions will be performed, can be defined and is kept. Another important requirement could be that the configuration agent is able to take actual conditions of the application 4 and/or the server 1 into consideration. To fulfill such needs, in a more sophisticated approach a script instead of a simple table can be used to define the control actions. The script can be written in XML or any other structured language and is processed by a kind of interpreter within the configuration agent. A script would offer more flexibility compared to a table.

After having determined the appropriate configuration actions, these actions are then performed in steps D and E. Usually, actions concerning the runtime environment 5 of the application 4 are performed first (step D). As mentioned earlier, examples for actions concerning the runtime
environment 5 of the application 4 are changing settings of the server 1, establishing network connections or mounting file systems on the server 1. Also, other applications or processes could be the target of such actions, i.e. stopping other applications or processes that could conflict with the application 4 or starting other applications or processes are needed by the application 4.

Finally, in step E, the application 4 is configured and/or controlled itself. Examples for actions concerning the application 4 itself are changing settings of the application 4, either via a local interface that the application provides or via a configuration file used by the application 4. Also, the application, or more specific, an instance of the application, could be stopped or started or restarted.

It is to be understood that the examples given in the disclosed embodiments are in all respects illustrative and not restrictive. The invention can be embodied in other specific forms without departing from the spirit or essential character thereof.
List of Reference Signs

1 Server
2 Client
3 Network
4 Application
5 Runtime environment
6 Configuration agent
7 Administration means
8 Control connection
Claims

1. A method for provisioning an application (4) on a server (1) in a computer arrangement, the computer arrangement comprising:
   - a plurality of servers (1), each server (1) hosting one or more applications (4) which provide services to clients (2) over a network (3) and each server (1) hosting at least one application-specific configuration agent (6);
   - an administration means (7) connected to all servers (1);
with the following steps:
   - transmitting a control command concerning an application (4) to be provisioned on one of the servers (1) from the administration means (7) to the configuration agent (6) specific to said application (4) on said server (1);
   - receiving the control command by said configuration agent (6);
   - determining one or more configuration actions concerning the application (4) and/or its runtime environment (5) on the basis of the received control command by said configuration agent (6);
   - executing the configuration actions by the configuration agent (6).

2. The method according to claim 1, where the configuration actions concerning the application (4) comprise one or more of the following steps:
   - changing settings of the application (4) to be provisioned;
   - stopping the application (4) to be provisioned;
   - starting the application (4) to be provisioned.
3. The method according to one of the claims 1 or 2, where the configuration actions concerning the runtime environment (5) of the application (4) comprise one or more of the following actions:
- changing settings of the server;
- establishing network connections;
- mounting file systems on the server;
- stopping other applications that could conflict with the application (4) to be provisioned;
- starting other applications that are needed by the application (4) to be provisioned.

4. The method according to one of the claims 1 to 3, where the control command concerning an application (4) follows a uniform interface specification which is not application-specific.

5. The method according to claim 4, where the control command concerning an application (4) is in form of a mark-up language based transaction, in particular in form of an XML-based (eXtensible Markup Language) transaction.

6. The method according to one of the claims 1 to 5, where a table is provided which comprises a list of configuration actions for possible control commands, and where the configuration actions for a received control commands are determining from said table.

7. The method according to claim 6, where the configuration actions are executed in the order they are listed in the table.
8. A configuration agent (6) for provisioning an application (4) on a server, the configuration agent (6) being set up to
- receive a control command concerning an application (4) to be provisioned hosted on the servers (1) over a network connection;
- determine one or more configuration actions concerning the application (4) and/or its runtime environment (5) on the basis of the received control command;
- executing the configuration actions.

9. The configuration agent (6) according to claim 8, where the configuration actions executed by the configuration agent (6) can comprise one or more of the following actions:
- changing settings of the application (4) to be provisioned;
- stopping the application (4) to be provisioned;
- starting the application (4) to be provisioned;
- changing settings of the server (1);
- establishing network connections;
- mounting file systems on the server (1);
- stopping other applications that could conflict with the application (4) to be provisioned;
- starting other applications that are needed by the application (4) to be provisioned.

10. The configuration agent (6) according to one of the claims 8 or 9, characterized in that a table is provided which comprises a list of configuration actions for possible control commands, and that the configuration actions for a received control command are determining from said table.
Fig. 2:

administration means 7:

transmit control command

server 1:

receive control command

B

determine configuration actions

C

configure runtime environment 5

D

configure application 4

E
INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER
G06F9/50

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)
G06F  H04L

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, INSPEC, IBM-TDB

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
</table>

Further documents are listed in the continuation of box C.

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 doubt 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.

"Z" document member of the same patent family

Date of the actual completion of the international search
11 November 2005

Date of mailing of the international search report
24/11/2005

Name and mailing address of the ISA
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

Authorized officer
Dieben, M

Form PCT/ISA/210 (second sheet) (January 2004)
<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
</table>
| A        | US 2003/126202 A1 (WATT CHARLES T)  
3 July 2003 (2003-07-03)  
paragraphs ‘0011! – ‘0014! | 1-10                 |
<table>
<thead>
<tr>
<th>Patent document cited in search report</th>
<th>Publication date</th>
<th>Patent family member(s)</th>
<th>Publication date</th>
</tr>
</thead>
<tbody>
<tr>
<td>US 2003126202 A1</td>
<td>03-07-2003</td>
<td>NONE</td>
<td></td>
</tr>
</tbody>
</table>