METHOD AND SYSTEM FOR PATCHING A VIRTUAL IMAGE

An improved method for patching a virtual image comprises the steps: Modifying a selected dormant virtual image to be patched by injecting a corresponding patch logic, and patch material to be applied on next boot during an off-line preparation phase; downloading a boot medium (42) and creating a temporary disk (52) for a selected target virtual machine (50) with corresponding deployment data; changing a master boot record of said temporary disk (52) associated with said target virtual machine (50) to boot next on said boot medium (42); and executing said patch logic to install said patch material in case said target virtual machine (50) associated with said virtual image to be patched is booted.
(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), OAPI (BF, BJ, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), Published: with international search report (Art. 21(3))
DESCRIPTION

METHOD AND SYSTEM FOR PATCHING A VIRTUAL IMAGE

BACKGROUND OF THE INVENTION

Field of the Invention

The present invention relates in general to the field of virtualization, and in particular to a method for patching a virtual image and a system for patching a virtual image. Still more particularly, the present invention relates to a data processing program and a computer program product for patching a virtual image.

Description of the Related Art

While virtualization brought a lot of advantages in terms of optimization of resources utilization it also introduced new challenges. The more evident issue is strictly tied on how to manage and maintain an increasing number of virtual images. Typically, virtual images are captured and stored in a central image repository and are maintained through versioning and provenance control mechanisms. Among the different maintenance actions, a key issue is how to bring those images to the same patch level. The security policies that usually are applied to running virtual or physical machines need to be also applied to dormant images. The more is delayed the deployment of an operating system patch the greater is the risk of viruses infections once images are instantiated. The most common way to apply patches to dormant images is to instantiate them one by one in a segregated network just for the time required to deploy the change through standard deployment mechanisms; this approach has the major drawbacks, that it is inefficient that any dormant virtual image is re-instantiated for deploying the new patches.
even if it is not sure they will be used in the future; and even if the dormant virtual image is instantiated in a segregated network there is no guarantee to not have virus exposure.

The search was focused towards finding prior art which discloses a method or system of preparing dormant virtual images to be patched in an offline state and patching the virtual images during the next boot process when the virtual images are re-instantiated in the production environment. More specifically, the search was directed towards finding prior art which discloses the following features: Preparing dormant virtual images to be patched in an offline state by modifying the virtual images with the proper patch logic and the material (patch software module, pre-OS environment) to be applied on the next boot process; and Executing the patch logic on top of a pre-OS environment when the virtual machine associated with the virtual image boots (re-instantiation). This ensures that patches are applied just when actually needed.

In the Patent Publication US 7,823,145 B1 "UPDATING SOFTWARE ON DORMANT DISKS" by Le et al. a system and method for scanning and updating software on a dormant disk is disclosed. The disclosed a method of updating a dormant disk without requiring booting of the dormant disk uses an indirect mechanism, wherein the method includes the step of scanning a dormant disk to determine a current status of the dormant disk, determining whether the updates are available and applying the updates to the dormant disk using the indirect mechanism. The indirect mechanism includes means for storing a script on the dormant disk, wherein the script is configured to update the files upon booting of the dormant disk. All in the entire document describes the traditional way to patch off-line virtual images, and relies on the concept to re-instantiate the virtual image in a segregated network just for patching purposes.
Summary of the Invention

The technical problem underlying the present invention is to provide a method for patching a virtual image and a system for patching a virtual image, which are able to apply any change including device driver modification and to solve the above mentioned inefficiencies, shortcomings and pain points of prior art virtual image patching.

According to the present invention this problem is solved by providing a method for patching a virtual image having the features of claim 1, a system for patching a virtual image having the features of claim 9, a data processing program for patching a virtual image having the features of claim 14, and a computer program product for patching a virtual image having the features of claim 15. Advantageous embodiments of the present invention are mentioned in the subclaims.

Accordingly, in an embodiment of the present invention a method for patching a virtual image comprises the steps: Modifying a selected dormant virtual image to be patched by injecting a corresponding patch logic, and patch material to be applied on next boot during an off-line preparation phase; downloading a boot medium and creating a temporary disk for a selected target virtual machine with corresponding deployment data; changing a master boot record of the temporary disk associated with the target virtual machine to boot next on the boot medium; and executing the patch logic to install the patch material in case the target virtual machine associated with the virtual image to be patched is booted.

In further embodiments of the present invention, the patch material comprises a patch software module and information about
a pre-operating system environment to be used for patching the selected dormant virtual image.

In further embodiments of the present invention, the pre-operating system environment is started by the boot medium during booting of the target virtual machine associated with the virtual image to be patched and taking control of the patching process.

In further embodiments of the present invention, the patch software module is executed on top of the pre-operating system environment.

In further embodiments of the present invention, the pre-operating system environment starts an agent directly accessing the selected virtual image and applying changes according to the patch software module.

In further embodiments of the present invention, the patching method is selected during the off-line preparation phase.

In further embodiments of the present invention, the virtual image to be patched and the patch software module to apply are selected and bound during the off-line preparation phase only, if the selected patching method is a next reboot patching method and virtual image deployment with patching is selected.

In further embodiments of the present invention, the boot medium is an ISO image.

In another embodiment of the present invention, a system for patching a virtual image comprises a browser, an image provisioning server with an image repository holding at least one virtual image, and a virtualization infrastructure comprising at least one hypervisor running at least one virtual
machine; wherein the browser is used to select a dormant virtual image to be patched from the image repository, patch material to be applied, and a target virtual machine; wherein the selected dormant virtual image to be patched is modified by injecting a corresponding patch logic, and the patch material to be applied on next boot during an off-line preparation phase; wherein the image provisioning server contacts an on screen display an operating System deployment (OSD) tool of the virtualization infrastructure to download a boot medium and to create a temporary disk for the selected target virtual machine with corresponding deployment data; wherein the on screen display (OSD) tool changes a master boot record of the temporary disk associated with the target virtual machine to boot next on the boot medium; and executes the patch logic to install the patch material in case the target virtual machine associated with the virtual image to be patched is booted.

In further embodiments of the present invention, the patch material comprises a patch software module and information about a pre-operating system environment to be used for patching of the selected dormant virtual image.

In further embodiments of the present invention, the on screen display (OSD) tool starts the target virtual machine which boots on the ISO image; wherein the target virtual machine downloads the pre-operating system environment and the on screen display (OSD) tool from the hypervisor to the temporary disk, mounts the virtual image to be patched, downloads and deploys corresponding files.

In further embodiments of the present invention, the patch software module is executed on top of the pre-operating system.

In further embodiments of the present invention, the pre-operating system environment starts an agent directly accessing
the selected virtual image and applying changes according to the patch software module.

In another embodiment of the present invention, a data processing program for execution in a data processing system comprises software code portions for performing a method for patching a virtual image when the program is run on the data processing system.

In yet another embodiment of the present invention, a computer program product stored on a computer-readable medium, comprises computer-readable program means for causing a computer to perform a method for patching a virtual image when the program is run on the computer.

All in all, embodiments of the present invention rely on the assumption to just prepare off-line the dormant virtual image to be patched and postpone the actual patching when the virtual image is re-instantiated in the production environment. This is obtained by modifying the virtual image injecting off-line the proper patch logic and the material comprising a patch software module, and a pre-operating system environment to be applied on a next boot process. A key aspect of the present invention is that, once the virtual machine associated to the virtual image boots, the patch logic is executed on top of a pre-operating system environment, for example WinPE for Windows (a trademark of Microsoft corporation) and the pre-boot operating system for Altiris Deployment Solution (Altiris and Deployment Solution are trademarks of Symantec Corporation) for Linux (Linux is a trademark of Linus Torvalds). This would happen when the network is still not available preventing any virus attack. In addition to resolve the security concern, embodiments of the present invention guarantee patches are applied just when actually needed dramatically reducing the inefficiency of the traditional approaches.
While some of the prior art solutions, describe the step to do indirect/postponed patch preparing the dormant virtual image to be patched at boot time embodiments of the present invention have a significant difference that also brings evident advantages if compared with the prior art solutions.

While the prior art mechanism modify the dormant virtual image including directly in the virtual image itself the update scripts, embodiments of the present invention do not apply these changes but just modify the master boot record (MBR) to force the machine to boot on a loaded boot medium, for example an ISO file (image), instead of the hard disk. This means that at boot time the control is taken by the pre-operating system environment included in the loaded boot medium that would start an agent that directly accesses the virtual image itself and applies the changes to it. This implies that it is possible to apply any change including device driver modification that are not possible in the prior art solutions. So, the target system boots from an intermediate boot medium, for example the ISO disk, and the patch is applied only on the local target virtual machine (VM) disk.

The above, as well as additional purposes, features, and advantages of the present invention will become apparent in the following detailed written description.

**Brief Description of the Drawings**

A preferred embodiment of the present invention, as described in detail below, is shown in the drawings, in which

FIG. 1 is a schematic block diagram of a system for patching a virtual image, in accordance with an embodiment of the present invention; and
FIG. 2 is a schematic flow diagram of a method for patching a virtual image, in accordance with a first embodiment of the present invention.

**Detailed Description of the Preferred Embodiments**

FIG. 1 shows a system for patching a virtual image, in accordance with an embodiment of the present invention.

Referring to FIG. 1, the shown embodiment of the present invention employs a system 1 for patching a virtual image comprising a browser 10, an image provisioning server 20 with an image repository 30 holding at least one virtual image, and a virtualization infrastructure 5 comprising at least one hypervisor 40 running at least one virtual machine 50, 60, 70. In the shown embodiment the hypervisor 40 is implemented as, for instance, the VMware ESX hypervisor (VMware is a trademark of VMware Inc.) type comprising a boot medium 42, an on screen display (OSD) tool 44, a hypervisor (HYP) Kernel and a Linux Kernel, for example, and the image provisioning server 20 is implemented as Tivoli provisioning manager for images (TPMImages). Even if these hypervisor and image provisioning server types are mentioned, embodiments of the present invention may work without any difference varying the hypervisor and image provisioning server types.

The browser 10 is used to select a dormant virtual image to be patched from the image repository 30, patch material to be applied, and a target virtual machine 50; wherein the selected dormant virtual image to be patched is modified by injecting a corresponding patch logic, and the patch material to be applied on next boot during an off-line preparation phase.

The image provisioning server 20 contacts the on screen display (OSD) tool 44 of the virtualization infrastructure 5 to download
the boot medium 42 and to create a temporary disk 52 for the selected target virtual machine 50 with corresponding deployment data. The on screen display (OSD) tool 44 changes a master boot record (MBR) of the temporary disk 52 associated with the target virtual machine 50 to boot next on the boot medium 42; and executes the patch logic to install the patch material in case the target virtual machine 50 associated with the virtual image to be patched is booted.

The patch material comprises a patch software module and information about a pre-operating system environment 56 to be used for patching of the selected dormant virtual image.

In other words, during the patch preparation phase, an operator selects the virtual image to be patched, the software module to apply including the patch and the patching method like immediate, next reboot or scheduled. If "next reboot" and "image deployment with patching" are selected, the image provisioning server 20 binds the software module to the virtual image to be deployed.

During the virtual image und patch deployment phase, the operator selects the target virtual machine 50 and triggers a deployment action. In reaction to the trigger process the image provisioning server 20 contacts the on screen display (OSD) tool 44 running on the hypervisor 40. The on screen display (OSD) tool 44 downloads a network boot ISO image as boot medium, for example, and creates the temporary virtual machine (VM) disk 52 with the deployment data. The on screen display (OSD) tool 44 changes the master boot record (MBR) of the virtual machine (VM) disk 52 in order to boot on the ISO file (image) as boot medium 42. The on screen display (OSD) tool 44 starts the virtual machine (VM) 50. The virtual machine (VM) 50 boots on the ISO file (image) as boot medium 42, and the pre-operating system 56 and the on screen display (OSD) tool 54 are downloaded from the
hypervisor 40 and loaded in a ram disk. Then virtual images are mounted and the files are downloaded and deployed. The on screen display (OSD) tool 56 runs an agent to prepare the operating system (OS), to inject the device drivers and to install the patch software module. So the patch software module is executed on top of the pre-operating system 56.

FIG. 2 shows a method for patching a virtual image, in accordance with an embodiment of the present invention.

Referring to FIG. 2, the shown embodiment of the present invention employs a method for patching a virtual image. In step S100 a dormant virtual image to be patched, a patch software module to apply and a patching method is selected. In step S200 the patch software module is bound to the dormant virtual image to be patched, if "next reboot" and "image deployment with patching" are selected as patching method.

In step S300 the selected dormant virtual image to be patched is modified by injecting a corresponding patch logic, and patch material to be applied on next boot during an off-line preparation phase. In step S400 a boot medium 42 is downloaded and a temporary disk 52 for a selected target virtual machine 50 is created with corresponding deployment data. In step S500 a master boot record of the temporary disk 52 associated with the target virtual machine 50 is changed to boot next on the boot medium 42. In step S600 the patch logic is executed to install the patch material in case the target virtual machine 50 associated with the virtual image to be patched is booted.

The patch material comprises a patch software module and information about the pre-operating system environment 56 to be used for patching the selected dormant virtual image. The pre-operating system environment 56 is started by the boot medium 42 during booting of the target virtual machine 50 associated with
the virtual image to be patched and taking control of the patching process, wherein said patch software module is executed on top of said pre-operating system environment 56. As mentioned above the pre-operating system environment 56 starts an agent directly accessing the selected virtual image to be patched and applying changes according to the patch software module.

Embodiment of the present inventive can be implemented as an entirely software embodiment, or an embodiment containing both hardware and software elements. In a preferred embodiment, the present invention is implemented in software, which includes but is not limited to firmware, resident software, microcode, etc.

Furthermore, the present invention can take the form of a computer program product accessible from a computer-readable or computer-readable medium providing program code for use by or in connection with a computer or any instruction execution system. For the purposes of this description, a computer-readable or computer-readable medium can be any apparatus that can contain, store, communicate, propagate, or transport the program for use by or in connection with the instruction execution system, apparatus, or device.

The medium can be an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system (or apparatus or device) or a propagation medium. Examples of a computer-readable medium include a semiconductor or solid state memory, magnetic tape, a removable computer diskette, a random access memory (RAM), a read-only memory (ROM), a rigid magnetic disk, and an optical disk. Current examples of optical disks include compact disk - read only memory (CD-ROM), compact disk - read/write (CD-R/W), and DVD. A data processing system suitable for storing and/or executing program code will include at least one processor coupled directly or indirectly to memory elements through a system bus. The memory elements can include local
memory employed during actual execution of the program code, bulk storage, and cache memories which provide temporary storage of at least some program code in order to reduce the number of times code must be retrieved from bulk storage during execution.

Input/output or I/O devices (including but not limited to keyboards, displays, pointing devices, etc.) can be coupled to the system either directly or through intervening I/O controllers.

Network adapters may also be coupled to the system to enable the data processing system to become coupled to other data processing systems or remote printers or storage devices through intervening private or public networks. Modems, cable modems, and Ethernet cards are just a few of the currently available types of network adapters.
What is claimed is:

1. A method for patching a virtual image comprising:
   Modifying a selected dormant virtual image to be patched by injecting a corresponding patch logic, and patch material to be applied on next boot during an off-line preparation phase;
   downloading a boot medium (42) and creating a temporary disk (52) for a selected target virtual machine (50) with corresponding deployment data;
   changing a master boot record of said temporary disk (52) associated with said target virtual machine (50) to boot next on said boot medium (42); and
   executing said patch logic to install said patch material in case said target virtual machine (50) associated with said virtual image to be patched is booted.

2. The method according to claim 1, wherein said patch material comprises a patch software module and information about a pre-operating system environment (56) to be used for patching said selected dormant virtual image.

3. The method according to claim 2, wherein said pre-operating system environment (56) is started by said boot medium (42) during booting of said target virtual machine (50) associated with said virtual image to be patched and taking control of said patching process.

4. The method according to claim 3, wherein said patch software module is executed on top of said pre-operating system environment (56).
5. The method according to claim 3 or 4, wherein said pre-operating system environment (56) starts an agent directly accessing said selected virtual image and applying changes according to said patch software module.

6. The method according to one of the preceding claims 1 to 5, wherein said patching method is selected during said off-line preparation phase.

7. The method according to claim 6, wherein said virtual image to be patched and said patch software module to apply are selected and bound during said off-line preparation phase only, if said selected patching method is a next reboot patching method and virtual image deployment with patching is selected.

8. The method according to one of the preceding claims 1 to 7, wherein said boot medium (42) is an ISO image.

9. A system for patching a virtual image comprising:
   A browser (10), an image provisioning server (20) with an image repository (30) holding at least one virtual image, and a virtualization infrastructure (5) comprising at least one hypervisor (40) running at least one virtual machine (50, 60, 70);
   wherein said browser (10) is used to select a dormant virtual image to be patched from said image repository (30), patch material to be applied, and a target virtual machine (50);
   wherein said selected dormant virtual image to be patched is modified by injecting a corresponding patch logic, and said patch material to be applied on next boot during an off-line preparation phase;
wherein said image provisioning server \((20)\) contacts an OSD tool \((44)\) of said virtualization infrastructure \((5)\) to download a boot medium \((42)\) and to create a temporary disk \((52)\) for said selected target virtual machine \((50)\) with corresponding deployment data;

wherein said OSD tool \((44)\) changes a master boot record of said temporary disk \((52)\) associated with said target virtual machine \((50)\) to boot next on said boot medium \((42)\); and

does said patch logic to install said patch material in case said target virtual machine \((50)\) associated with said virtual image to be patched is booted.

10. The system according to claim 9, wherein said patch material comprises a patch software module and information about a pre-operating system environment \((56)\) to be used for patching of said selected dormant virtual image.

11. The system according to claim 10, wherein said OSD tool \((44)\) starts said target virtual machine \((50)\) which boots on said ISO image;

wherein said target virtual machine \((50)\) downloads said pre-operating system environment \((56)\) and said OSD tool \((54)\) from said hypervisor \((40)\) to said temporary disk \((52)\), mounts said virtual image to be patched, downloads and deploys corresponding files.

12. The system according to claim 11, wherein said patch software module is executed on top of said pre-operating system \((56)\).

13. The system according to claim 11 or 12, wherein said pre-operating system environment \((56)\) starts an agent
directly accessing said selected virtual image and applying changes according to said patch software module.

14. A data processing program for execution in a data processing system comprising software code portions for performing a method for creating a virtual appliance according to one of the preceding claims 1 to 8 when said program is run on said data processing system.

15. A computer program product stored on a computer-readable medium, comprising computer-readable program means for causing a computer to perform a method for creating a virtual appliance according to one of the preceding claims 1 to 8 when said program is run on said computer.
S100 Selecting
- a Dormant Virtual Image to be patched;
- a Patch Software Module to apply; and
- a Patching Method.

S200 Binding the Patch Software Module to the Dormant Virtual Image to be patched, if "Next Reboot" and "Image Deployment with Patching" is selected.

S300 Modifying the selected Dormant Virtual Image to be patched by injecting a corresponding Patch Logic, and Patch material on next boot during an Off-Line Preparation Phase.

S400 Downloading a Boot Medium and creating a Temporary Disk for a selected Target Virtual Machine with corresponding Deployment Data.

S500 Changing a Master Boot Record of the Temporary Disk associated with the Target Virtual Machine to boot next on the Boot Medium.

S600 Executing the Patch Logic to install the Patch Material in case the Target Virtual Machine associated with the Virtual Image to be patched is booted.

FIG. 2
## INTERNATIONAL SEARCH REPORT

**International application No.**
PCT/IB2012/056945

### A. CLASSIFICATION OF SUBJECT MATTER

**Int.Cl.** G06F11/00 (2006.01) i

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

### B. FIELD(S) SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

**Int.Cl.** G06F11/00

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Published examined utility model applications of Japan 1922-1996
Published unexamined utility model applications of Japan 1971-2013
Published registered utility model applications of Japan 1994-2013

Electronic database consulted during the international search (name of database and, where practicable, search terms used)

### 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>
<tbody>
<tr>
<td>A</td>
<td>WO 2007/136448 A1 (MICROSOFT CORPORATION) 2007.11.29, see the whole document</td>
<td>1 - 15</td>
</tr>
<tr>
<td>A</td>
<td>US 2010/0088699 A1 (Takayuki SASAKI) 2010.04.08, see the whole document</td>
<td>1 - 15</td>
</tr>
<tr>
<td></td>
<td>&amp; WO 2008/117500 A1</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>US 2009/0132683 A1 (Yusuke FURUYAMA) 2009.05.21, see the whole document</td>
<td>1 - 15</td>
</tr>
<tr>
<td></td>
<td>&amp; JP 2009-122963 A</td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>JP 2007-183747 A (HITACHI, LTD.) 2007.07.19, see the whole document (Family: none)</td>
<td>1 - 15</td>
</tr>
</tbody>
</table>

* Further documents are listed in the continuation of Box C. □ See patent family annex.

* "A" document defining the general state of the art which is not considered to be of particular relevance
  "E" earlier application or patent 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**
26. 02. 2013

**Date of mailing of the international search report**
05. 03. 2013

**Name and mailing address of the ISA/JP**
Japan Patent Office
3-4-3, Kasumigaseki, Chiyoda-ku, Tokyo 100-8915, Japan

**Authorized officer**
Hiroshi MAEDA

**Telephone No.** +81-3-3581-1101 Ext. 354 5

Form PCT/ISA/210 (second sheet) (July 2009)