(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
23 March 2017 (23.03.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/046830 A1

(51) International Patent Classification:
G06F 9/445 (2006.01) *G06F 9/46* (2006.01)

(21) International Application Number:
PCT/JP2015/004783

(22) International Filing Date:
17 September 2015 (17.09.2015)

(25) Filing Language: English

(26) Publication Language: English

(71) Applicant: **HITACHI, LTD.** [JP/JP]; 6-6, Marunouchi 1-chome, Chiyoda-ku, Tokyo, 1008280 (JP).

(72) Inventors: **LUO, Cheng**; c/o HITACHI, LTD., 6-6, Marunouchi 1-chome, Chiyoda-ku, Tokyo, 1008280 (JP). **MORIKI, Toshiomi**; c/o HITACHI, LTD., 6-6, Marunouchi 1-chome, Chiyoda-ku, Tokyo, 1008280 (JP). **IMADA, Takayuki**; c/o HITACHI, LTD., 6-6, Marunouchi 1-chome, Chiyoda-ku, Tokyo, 1008280 (JP).

(74) Agent: **WILLFORT INTERNATIONAL PATENT FIRM**; Nihonbashi TC Bldg. 1F, 19-7, Nihonbashi Koamicho, Chuo-ku, Tokyo, 1030016 (JP).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(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, ST, SZ, TZ, UG, ZM, ZW), 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), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

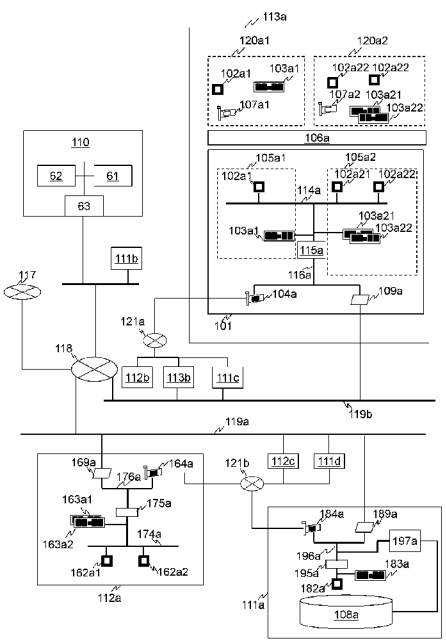
Published:

— *with international search report (Art. 21(3))*

WO 2017/046830 A1

(54) **Title**: METHOD AND SYSTEM FOR MANAGING INSTANCES IN COMPUTER SYSTEM INCLUDING VIRTUALIZED COMPUTING ENVIRONMENT

[Fig. 1]



(57) **Abstract**: A management system for managing instances in a computer system is built. The computer system includes a virtualized computing environment which is a virtualized compute node executing a virtualization program for defining virtualized servers, a bare metal environment which is a physical compute node as a bare metal server with an OS, and block storage systems configured to provide volumes storing images associated with OSs. The management system reserves one or more identifiers for one or more new instances, among identifiers (e.g. World Wide Port Names) which are used to access volumes. The management system requests one or more storage systems to build a connection between the reserved identifiers and one or more volumes. The storage systems have the one or more volumes. Each of the volumes is a template volume which stores the image associated with the OS for the new instances or an attached volume of the template volume. The management system activates each of the new instances to boot, from the image, the OS for the new instance associated with a reserved identifier among the reserved identifiers. The image is in the volume connected with the reserved identifier.

Description

Title of Invention: METHOD AND SYSTEM FOR MANAGING INSTANCES IN COMPUTER SYSTEM INCLUDING VIRTUALIZED COMPUTING ENVIRONMENT

Technical Field

[0001] The present invention generally relates to a technology for managing instances in a computer system including a virtualized computing environment.

Background Art

[0002] A virtualized computing environment, which is typically known as a cloud computing environment, provides a large scale computing functionality in applications such as a web service and scientific computing. The virtualized computing environment typically provides a platform for managing instances with different OS (Operating System) and running different applications within each OS. "Instance" is a set of computing resources with OS running on it. Specifically, for example, an instance can be a set of computing resources with a VM (Virtual Machine), a set of computing resources with a LPAR (Logical PARtition) and a bare metal server with an OS.

[0003] A server virtualization technology in the cloud computing environment involves partitioning or sharing of a physical server for multiple isolated virtual servers. The virtualized servers are managed by virtualization software which is well known as hypervisor or virtual machine monitor. The server virtualization technology distributes physical resources into multiple isolated virtualized servers.

[0004] A logical partitioning technology, which is known as one server virtualization technology with high efficiency, can divide hardware resources (physical computing resources) such as processors, memories and I/O devices into multiple subsets of resources named LPARs. The LPARs on the same physical server can be operated independently with its own OS and applications.

[0005] According to other server virtualization technologies, for example, a VM technology, VMs hold virtual computing resources such as virtual processors. Compared with other server virtualization technologies, the big difference is LPARs hold physical resources. With this basic difference, the logical partitioning technology can provide better performance, reliability and compatible management between virtualized environments and bare metal environments.

[0006] LPARs have less overhead caused by virtualizing physical computing resources compared to VMs to achieve better performance. In reliability aspect, partial hardware failure such as one processor failure only affects the LPARs using the failed physical

computing resources and other LPARs on the physical server can survive. But all VMs on the same physical server cannot survive from partial hardware failure due to the collapse of the software virtualization layer. Furthermore, a system running on LPARs can be migrated to a bare metal environment without any modification as both environments using physical computing resources. For a VM case, a system on VMs need to be re-configured for the environment change from using virtual computing resources to physical computing resources. A typical user case is an enterprise developing and testing a new service in a virtualized environment and publishing a product in a bare metal environment. The compatible management between virtualized environments and bare metal environments of LPARs can accelerate the speed from development to product for an enterprise.

[0007] Storage within a cloud computing environment is typically manually configured for an instance by a user such as a system operator using a management tool to configure a storage system coupled to the instance. With provided the storage, the instance can boot the OS within the storage and run a service. Two typical types of storage for booting an instance are an image located in a local disk and an image located in a volume provided from a block storage apparatus. Typically, the management of computing resources and storages are separated which means to deploy an instance requiring configuration from a server manager to configure a server and a storage manager to configure storage.

[0008] An "image" is a file associated with OS. Specifically the image can be typically a single file containing a virtual disk that has a bootable OS installed on it. As a file, the image can be easily transferred to anywhere via a remote network which resides outside a system including the target server. A typical usage to deploying an instance with an image is transferring a copy of the image from an image database server to the local disk of the target server, defining the instance and booting the instance from the copied image. There are some methods and tools to automate the image based deployment. A user selects an image from a web based interface and clicks a deploying button. Then the selected image will be automatically transferred to the target server for the new instance deployment which reduces much OPEX (Operating Expense) as the user does not need to manually configure at least one of a storage system and a computing server. The image based deployment provides easy management for users.

[0009] A volume from a block storage apparatus is another option for deploying a new instance. A storage administrator configures a bootable volume to enable access from the target server via a WWPN (World Wide Port Name), which is a worldwide name assigned to a port in a FC (Fibre Channel) fabric used on a storage area network and performs a function equivalent to the MAC address in Ethernet protocol as it is supposed to be an unique identifier in the network, and mounts to the target computing

server as a local block device. Then the server administrator defines the new instance and assigns the local block device to the new instance for booting. Compared to the image based deployment, the volume based deployment consumes less time cost due to no remote network transmission. The volume based deployment also provides better reliability compared to the image based deployment. In a case where the computing server fails, the attached volume can be quickly assigned to another computing server to recover the failed instance. However, the volume based deployment requires many manual configurations.

[0010] Recently there is a method that automates the process by implementing interfaces in both a hypervisor and a storage system. This method is not suitable for LPARs. The extra interface layer in both the hypervisor and the storage system make a big difference between a virtualized environment and a bare metal environment. In this case, a system on LPARs requires reconfiguring to work on a bare metal server which disables the LPAR feature of compatible management between virtualized environments and bare metal environments.

Citation List

Patent Literature

[0011] [PTL 1] US2013/0346615 A1

[PTL 2] EP2843549 A2

Summary of Invention

Technical Problem

[0012] As mentioned above, traditional image based deployment can provide easy management but has long deployment time, especially in a case of a large number of instance deployments. Traditional volume based deployment can provide fast deployment but require many manual configuration. There is a method to provide image-liked easy management with volumes. But the method is not suitable for the logical partitioning technology which can provide compatible management between virtualized environments and bare metal environments. Currently there is no method to provide easy management, deploying performance and compatible management at the same time.

[0013] The present invention targets at solution for proving image-liked easy management, volume-liked deploying performance and a compatible management feature.

Solution to Problem

[0014] A management system for managing instances in a computer system is built. The computer system includes a virtualized computing environment which is a virtualized compute node executing a virtualization program (e.g. a hypervisor or a virtual machine monitor) for defining virtualized servers (e.g. LPARs or VMs), a bare metal

environment which is a physical compute node as a bare metal server with an OS, and block storage systems configured to provide volumes storing images associated with OSs (e.g. including OSs or metadata of the OSs). The management system reserves one or more identifiers for one or more new instances, among identifiers which are used to access volumes. The management system requests one or more storage systems to build a connection between the reserved identifiers and one or more volumes. The storage systems have the one or more volumes. Each of the volumes is a template volume which stores the image associated with the OS for the new instances or an attached volume of the template volume. The management system activates each of the new instances to boot, from the image, the OS for the new instance associated with a reserved identifier among the reserved identifiers. The image is in the volume connected with the reserved identifier.

[0015] Specifically, for example, the management system includes WWPN management information such as a WWPN table, instance management information such as an instance table, and image management information such as an image table to achieve automatic management of instance on both LPAR and bare metal. The management system such as a control node automatically collects WWPNs from all the compute nodes which include both virtualized compute nodes with hypervisors and physical compute nodes (bare metal servers), and store the collected WWPNs in the WWPN table. The instance table is used, by the control node, to automatically maintain information of instances and keep a mapping between WWPNs, instances and volumes. The image table is used, by the control node, to maintain all images, each which only includes metadata of OS (or OS), and keep a mapping between images and volumes which are used as template volumes to generate volumes for new instances. The control node uses a reserved WWPN as a unique identifier to map with instance, physical compute node and volume. The control node provides a webpage to present available images for a user selection and automatically transfer selected image into corresponding volumes for the new instances. The control node sends the WWPNs and volume information to at least one storage system for building (or deleting) connection between instances and volumes automatically. The control node sends the WWPN to a hypervisor to assign corresponding virtual FC (Fibre Channel) port to new LPAR for the instance. The control node can presents LPAR migration between compute nodes using at least one of the tables. The control node can adopt heartbeat message to handle hardware failure and physical storage connection change.

Advantageous Effects of Invention

[0016] The present invention can provide image-liked easy management, deploying performance with volumes and compatible management between virtualized environments

and bare metal environments.

Brief Description of Drawings

[0017] [fig.1]FIG. 1 is a physical diagram illustrating a networked computer system in which techniques according to an embodiment of the present invention.

[fig.2]FIG. 2 is a logical diagram illustrating the networked computer system.

[fig.3]FIG. 3 illustrates a usage example of deploying an instance on a virtualized compute node according to the embodiment.

[fig.4]FIG. 4 illustrates an example of an instance catalog according to the embodiment.

[fig.5]FIG. 5 illustrates a WWPN table according to the embodiment.

[fig.6]FIG. 6 illustrates an instance table according to the embodiment.

[fig.7]FIG. 7 illustrates an image table according to the embodiment.

[fig.8]FIG. 8 is a flow chart for deploying an instance according to the embodiment.

[fig.9]FIG. 9 is a sequence chart for deploying an instance according to the embodiment.

[fig.10]FIG. 10 is a flow chart for deleting an instance according to according to the embodiment.

[fig.11]FIG. 11 is a sequence chart for deleting an instance according to the embodiment.

[fig.12]FIG. 12 is a flow chart for migrating an instance (a LPAR) from one compute node to another compute node according to the embodiment.

[fig.13]FIG. 13 is a sequence chart for migrating an instance (a LPAR) from one compute node to another compute node according to the embodiment.

[fig.14]FIG. 14 is a flow chart for initializing the said WWPN table according to the embodiment.

[fig.15]FIG. 15 is a sequence chart for initializing the said WWPN table according to the embodiment.

[fig.16]FIG. 16 is a flow chart for updating the said WWPN table and instance table based on heartbeat message according to the embodiment.

[fig.17]FIG. 17 is a sequence chart for updating the said WWPN table and instance table based on heartbeat message according to the embodiment.

[fig.18]FIG. 18 is a block diagram illustrating the mapping between the said image and template volumes according to the embodiment.

Description of Embodiments

[0018] Hereinafter, embodiments of the present invention are described with reference to the drawings. Same symbols indicate the same components.

[0019] It is noted that, in the following description, expressions such as “xxx table” are in

some cases used to describe information; however, the information may be expressed in a data configuration other than a table. In order to show that the information does not depend on a data configuration, at least one of the items in the "xxx table" can be called "xxx information".

[0020] Also, in the following description, a processing may be explained with "program" as a grammatical subject. However, a program is executed by a processor (for example, a CPU (Central Processing Unit)), thereby a predetermined processing being performing by properly using at least one of a storage resource (for example, a memory) and a communication interface device (for example, a communication port), etc., and thus a grammatical subject of processing may be a processor. The processing described while a program is treated as a grammatical subject may be processing performed by a processor or an apparatus having a processor. Additionally, the processor may include a hardware circuit that implements part or all of the processing. The program may be installed from a program source to an apparatus such as a computer. A program source may, for example, be a program distribution server or a computer-readable storage media. When the program source is the program distribution server, the program distribution server includes a processor (for example, a CPU) and a storage resource, and the storage resource further stores a distribution program and a program to be distributed. Then, when the processor of the program distribution server executes the distribution program, the processor of the program distribution server distributes the program to be distributed, to another computer.

[0021] Moreover, the management system may be configured of one or more computers. Specifically, for example, when a management computer displays information (specifically, when a management computer displays information on a display device thereof or a management computer transmits display information to a remote display computer), the management computer is the management system. Also, for example, when the functions identical or similar to those of the management computer are achieved by a plurality of computers, the plurality of computers (which may include a display computer when display is performed by the display computer) is the management system. It may be possible that the information is input and output, by an input/output device provided in the computer, to and from the computer. Examples of the input/output device may include a display device, a key board, and a pointing device; however, in place of or in addition to at least one of these, another device may be adopted. Further, as an alternative to the input/output device, a serial interface device or an Ethernet interface device (Ethernet is a registered trademark) may be adopted, such an interface device may be coupled to a display computer having a display device, a key board, and a pointer device, and the computer may transmit display information to the display computer and receive input information from the

display computer, whereby the information may be output (for example, displayed) and input. In the present embodiment, a control node 110 is a management computer having input devices such as a keyboard and a pointing device and output devices such as a display device.

[0022] Further, in the description below when the same type of elements are descried without being distinguished, the common number among the reference number respectively allocated to the elements may be used (for example, a storage system 111), and when the same type of elements are described with being distinguished, the reference numbers respectively allocated to the elements (for example, storage systems 111a, 111b, ..) may be used.

[0023] FIG. 1 is a physical diagram illustrating a networked computer system in which techniques according to an embodiment of the present invention.

[0024] The networked computer system comprises multiple storage systems 111, multiple compute nodes 112 and 113, and a control node 110 being coupled to them 111, 112 and 113. The compute nodes 112 are physical compute nodes such as bare metal servers (bare metal environments). The compute nodes 113 are virtualized compute nodes such as virtualized environments.

[0025] The storage systems 111 and the compute nodes 112 and 113 are divided into multiple groups 119 each including at least one compute node of physical compute nodes 112, virtualized compute nodes 113 and storage systems 111. Specifically, for example, the group 119a includes the physical compute nodes 112a and 112c and the storage systems 111a and 111d, and the groups 119b includes the physical compute node 112b, the virtualized compute nodes 113a and 113c and the storage systems 111c. In each group 119, there is a storage network 121 such as a SAN (Storage Area Network) and each compute node and each storage system are connected with the storage network 121.

[0026] Each of the compute nodes, each of the storage systems and the control node 110 are connected with an internal network 118. The internal network 118 is connected with an outside network 117. Each of the networks 117 and 118 can be IP (Internet Protocol) network. Specifically, for example, the internal network 118 can be a LAN (Local Area Network) or a WAN (Wide Area Network), and the outside network 117 can be a WAN or the internet. The groups 119 and the control node 110 are coupled via the inside network 118. All the components can access outside via the outside network 117.

[0027] Each of the compute nodes are a computer and comprises an interface device, a memory and a processor coupled to the interface device and the memory.

[0028] For example, the physical compute node 112a (one of the physical compute nodes 112) comprises interface devices such as a NIC (Network Interface Card) 169a for

communicating with outside apparatuses via the internal network 118 and a physical FC (Fibre Channel) port 164a for communicating with outside apparatuses via the storage network 121b. Furthermore, the physical compute node 112a comprises memories such as main memories 163a1 and 163a2. Furthermore, the physical compute node 112a comprises processors such as CPUs (Central Processing Units) 162a1 and 162a2. The NIC 169a and the physical FC port 164a are coupled to, via a bus (for example, PCI bus) 176a, a chip set 175a for controlling data transfer. The main memories 163a1 and 163a2 and the CPUs 162a1 and 162a2 are coupled to, via an interconnection 174a, the chip set 175a.

[0029] Also, for example, the virtualized compute node 113a (one of the virtualized compute nodes 112) comprises interface devices such as a NIC 109a for communicating with outside apparatuses via the internal network 118 and a physical FC port 104a for communicating with outside apparatuses via the storage network 121a. Furthermore, the virtualized compute node 113a comprises memories such as main memories 103a1, 103a2 and 103a3. Furthermore, the virtualized compute node 113a comprises processors such as CPUs 102a1, 102a2 and 102a3. The NIC 109a and the physical FC port 104a are coupled to, via a bus 116a, a chip set 115a for controlling data transfer. The main memories 103a1 to 103a3 and the CPUs 102a1 to 102a3 are coupled to, via an interconnection 114a, the chip set 115a.

[0030] Each of the virtualized compute nodes113, for example, the virtualized compute node 113a runs a hypervisor 106a partitions physical computing resources such as main memories 103a1 to 103a3 and CPUs 102a1 to 102a3 into subsets 105a1 and 105a2 and respectively allocates the subsets 105a1 and 105a2 to LPARs 120a1 and 120a2. The hypervisor 106a generates each of multiple virtual FC ports 107a1 and 107a2 from the physical FC port 104a. The hypervisor 106a can also respectively assign the virtual FC ports 107a1 and 107a2 to LPARs 120a1 and 120a2.

[0031] Each of the storage systems 111 such as a block storage system comprises a physical storage device such as a HDD (Hard Disk Drive) or a SSD (Solid State Drive) and a storage controller configured to perform data I/O (Input/Output) to/from the physical storage device.

[0032] For example, the storage system 111a (one of the storage systems 111) comprises a physical storage device 108a and a storage controller including a NIC 189a for communicating with outside apparatuses of the storage system 111a via the internal network 118, a physical FC port 184a for communicating with outside apparatuses of the storage system 111a via the storage network 121b, a storage interface 197a for communicating with the physical storage device 108a, a main memory 183a, and a CPU 182a. The NIC 189a, the physical FC port 184a, and the storage interface 197a are coupled to, via a bus 196a, a chip set 195a for controlling data transfer. The main

memory 183a and the CPU 182a are coupled to, via an interconnection, the chip set 195a.

[0033] The control node 110 can be a computer. The control node 110 comprises an interface device such as a NIC 63 for communicating with outside apparatuses via the internal network 118, a storage unit including at least a main memory 62, and a processor such as a CPU 61.

[0034] FIG. 2 is a logical diagram illustrating the networked computer system.

[0035] Each of the storage systems 111 provides logical volumes 201 to instances 205 which are hosts of the storage system 111. Each volume 201 can be a logical block storage device. Each volume 201 can be an actual volume which is a volume based on a physical storage device or a virtual volume such as a volume according to Thin Provisioning technology. For example, the storage system 111a provides volumes 201a1 and 201a2 based on the physical storage device 108a.

[0036] Each of the instances 205 can be allocated in a physical compute node 112 or a LPAR 120 in a virtualized compute node 113. Besides computing resources, each of the instances 205 includes a guest OS 203 and an application 204 running in the instance.

[0037] For example, the instance 205b is allocated in the physical compute node 112a. In other words, the instance 205b is the physical compute node 112a which is a bare metal server with an OS. The instance 205b holds WWPNs 202b1 and 202b2 each assigned to the physical FC port 164a. The instance 205b runs the guest OS 203b. In the instance 205b, at least one application 204 (e.g. the application 204b1) runs on the guest OS 203b. A WWPN 202 is an example of an identifier which is used to access a volume. Such an identifier is allocated to an interface (such as a FC port or a HBA (Host Bus Adapter) of a compute node. Such an identifier can be WWN (World Wide Name) or iSCSI name.

[0038] Also, for example, the instances 205a1 and 205a2 is allocated in LPARs 120a1 and 120a2 in the virtualized compute node 113a. In the virtualized compute node 113a, the hypervisor 106a holds WWPNs 202a1 and 202a2 each assigned to the virtualized FC port 104a and the WWPNs 202a1 and 202a2 can be respectively allocated to the LPARs 120a1 and 120a2 for connecting the storage system 111c. The instances 205a1 and 205a2 respectively runs the guest OS 203a1 and 203a2. In each of the instances 205a1 and 205a2, at least one application 204 (e.g. the application 204a11 or 204a21) runs on the guest OS 203a1 or 203a2.

[0039] In the control node 110, the storage unit stores a computer program such as a control program 241 executed by the CPU 61 and information such as a WWPN table 206, an instance table 207 and an image table 208. The WWPN table 206 is responsible for collecting and managing WWPNs 202 from all the physical compute nodes 112 and

the virtualized compute nodes 113. The instance table 207 is responsible for managing the mapping between instances 205, WWPNs 202 and volumes 201. The image table 208 is responsible for managing the mapping multiple volumes 201 to single image record in the table 208.

[0040] FIG. 3 illustrates a usage example of deploying an instance on the virtualized compute node 113a. FIG. 4 illustrates an example of an instance catalog screen.

[0041] The volumes 201c provided by the storage system 111c is OS pre-installed and can be used as a booting device. The contents which include pre-installed OS and applications in each of the volumes 201c can be different. The image table 208 maps multiple volumes 201 with the same content to a single virtual image 303. The control node 110 displays an instance catalog 400 which is a GUI (Graphical User interface) based on a webpage to present the virtual images 303 in the image table 208. The catalog 400 is configured to receive a selection (designation) of a virtual image (e.g. OS type (OS name)) and the number of instances from a user 302. The user 302, for a new instance deployment, select a virtual image 303 from the catalog 400 and input (designate) the number of instances about the selected virtual image 303. The number of instances can be designated by a system size. For example, if "small" is designated as a system size by the user 302, the control node 110 (the control program 214) can regard the number of instances as P (P is a natural number). Also, if bigger size than "small" is designated as a system size by the user 302, the control node 110 (the control program 214) can regard the number of instances as Q (Q is a natural number) which is larger than P.

[0042] In the example illustrated in FIGs. 3 and 4, the control node 110 reserves a WWPN 202a1 for a new instance 205a1 according to the state of WWPN 202a1 maintained in the WWPN table 206. Then control node 110 sends the WWPN 202a1 and the information of the volume 201c1 based on the selected virtual image 303 (OS name "OS-A") to the storage system 111c. The storage system 111c connects the WWPN 202a1 and the volume 201c1 to enable the new instance 205a1 to access the volume 201c1 via the virtualized FC port 104a that mapped to the WWPN 202a1. Finally the control node 110 sends the request to the hypervisor 106a to deploy and activate the instance 205a1 with the WWPN 202a1 on a LPAR 120 (120a1). The instance 205a1 accesses the storage system 111c via the storage network 121a and boots the guest OS from the volume 201c1.

[0043] FIG. 5 illustrates the WWPN table 206.

[0044] The WWPN table 206 comprises entries respectively corresponding to WWPNs 202. Each entry is constituted of fields. The fields includes a field for WWPN 202 where WWPN is stored, a field for state 401 where the state which shows the usage state of the WWPN 202 is stored, a field for PS_ID 402 where an identifier of a physical

compute node (where the WWPN 202 is located) is stored, a field for type 403 where the compute node type of the WWPN 202 is stored and a field for sys_ID 461 where an identifier of a storage system 111 connected to the WWPN 202 is stored. In the illustrated example, a WWPN "23140000876ec002" is idle and located in the virtualized compute node whose PS_ID is "Compute1". The storage systems whose sys_IDs are "sys01" and "sys02" are connected to the WWPN.

[0045] FIG. 6 illustrates the instance table 207.

[0046] The instance table 207 comprises entries respectively corresponding to instances 205. Each entry is constituted of fields. The fields includes a field for name 404 where the instance name is stored, a field for UUID (Universally Unique Identifier) 405 where the instance UUID is stored, a field for description 406 where the instance description is stored, a field for WWPN 202, a field for attached volume 407 where the location (identifier) of the attached volume of the instance is stored, a field for template volume 408 where the location (identifier) of the template volume for generating the attached volume is stored. In the illustrated example, an instance 205 whose name is "Tokyo" and whose UUID is "3245" has an OS whose name "OS-C". WWPN "23140000876ec006" is allocated to the instances 205. The attached volume whose identifier is "sys01:003" corresponds to the template volume whose identifier is "sys01:01", and is allocated to the instance 205. A template volume is a master volume and bootable volume with pre-installed OS inside. Each storage system 111 can generate an attached volume by copying data including OS from a template volume to another volume in the storage system 111 or each storage 111 system can use technologies such as copy-on-write to generate an attached volume. In the latter case, the attached volume is just a set of pointers to pointing to the template volume, in other words, the attached volume does not really store data..

[0047] FIG. 7 illustrates the image table 208.

[0048] The image table 208 comprises entries respectively corresponding to virtual images 303. Each entry is constituted of fields. The fields includes a field for name 409 where the virtual image name is stored, a field for UUID 410 where the image UUID is stored, a field for description 411 where the content description of the image is stored and a field for volume list 412 where all the locations (identifiers) of template volumes mapped to the image are stored. In the illustrated example, a virtual image 303 whose name is "client server" has UUID "314f-a3241-324". The description of the virtual image 303 shows the virtual image 303 includes the OS "OS-C" and the application "APP-A". The locations of the template volumes mapped to the virtual image 303 are "sys01:01" and "sys02:01".

[0049] FIGs. 8 and 9 illustrate processing of deploying an instance. Specifically, FIG. 8 is a flow chart for deploying an instance and FIG. 9 is a sequence chart for deploying an

instance. Prior to the start of the processing, the control node 110 has the WWPN table 206 storing all the WWPNs 202. The WWPNs 202 can be collected by the control program 241 (the control node 110) from all the compute nodes or generated by the control program 241. Steps of FIG. 8 are executed when the control program 241 receives, from the user 302 via the catalog 400, an instance deployment request associated with the identifier of the selected virtual image 303 and the number of instances.

[0050] In Step 501, the control program 241 reserves an unused WWPN (which is WWNP whose state is "unused") from the WWPN table 206 for the instance 205. The number of reserved WWPNs is equal to (or larger than) the number of instances input into the catalog 400 by the user 302. It is assumed that the number of reserved WWPNs is one because the number of instances input by the user 302 is one,

[0051] In Step 502, the control program 241 detects the type of the target compute node having the reserved WWPN, which is provided by the field of type 403 in the WWPN table 206. When the control program 241 detects the type is virtualized compute node 113, the control program 241 proceeds to Step 503. When the control program 241 detects the type is physical compute node 112, the control program 241 proceeds to Step 506.

[0052] In Step 503, the control program 241 sends the reserved WWPN 202 and volume information to the hypervisor 106 in the target virtualized compute node 113. The volume information includes information showing locations of volumes (attached volumes). The volumes are volumes storing the virtual image selected from the catalog 400 by the user 302. Specifically, The virtual image is the image including the OS whose OS type is the same as the OS type selected by the user 302. The control program 241 can find the virtual image whose description 411 includes the same OS type by referring to the image table 208 using OS type selected by the user 302. The locations of the volumes storing the virtual image can be obtained from the volume list 412 of the image table 208 by the control program 241.

[0053] In Step 504, the hypervisor 106 generates a request according to the received information from the control program 241. The request includes the reserved WWPN 202 and the volume information showing the locations of the volumes. The hypervisor 106 sends the request to build connection between the reserved WWPN 202 and the volumes to the storage system 111 having the volumes. Instead of the hypervisor 106, the control node 110 can generate and send the request to the storage system 111 as shown in FIG. 3.

[0054] In Step 505, the hypervisor 106 defines a new LPAR 120 for the instance 205. The hypervisor 106 allocates a sub set 105 and the virtual FC port 107 with the WWPN 202 to the new LPAR 120.

[0055] In Step 506, the control program 241 reserves all the other WWPNs 202 on the physical compute node 112 where the WWPN 202 reserved in Step 501 is located. In the physical compute node 112, there can be multiple WWPNs 202. And there can be only one instance located on the physical compute node 112. Once the physical compute node 112 is used to deploy for the instance 205 with the WWPN 202 reserved in Step 501, all the other WWPNs 202 on the physical compute node 112 cannot be used for any other instance 205.

[0056] In Step 507, the control program 241 generates a request to build connection between the WWPN 202 and the volumes. The request includes the reserved WWPN 202 and volume information showing the locations of the volumes (attached volumes). The control program 241 sends the request to the storage system 111 having the volumes.

[0057] In Step 508, the control program 241 generates a power-on command to turn on the target physical compute node 112, and sends the command to the physical compute node 112.

[0058] In Step 509, the control program 241 updates WWPN state of the reserved WWPN in the WWPN table 206 and creates the new entry (new record) for the new instance 205 in the instance table 207. As the result, the WWPN state is “unused” to “used”.

[0059] In Step 510, the storage system 111 receives the request and builds connection between the WWPN 202 and the volumes 201 in response to the request. The storage system 111 sets up the connection to enable the instance 205 access the volumes 201 for booting.

[0060] In Step 511, the target compute node activates the instance 205 and executes login sequence to the storage system 111, accessing the volumes 201 and booting the guest OS 203.

[0061] Through the above processing, the instance 205 can be deployed automatically to boot from the block volume 201 on either a virtualized compute node 113 or a physical compute node 112 after user 302 conducts selection via the catalog (webpage) 400.

[0062] FIGs. 10 and 11 illustrate processing for deleting an instance 205. Specifically, FIG. 10 is a flow chart for deleting an instance 205 and FIG. 11 is a sequence chart for deleting an instance. Steps of FIG. 10 are executed when the control program 241 receives an instance delete request from the user 302 via a webpage for delete processing like the catalog 400. The instance delete request can include at least one of the WWPN stored in the delete target instance and the identifier of the delete target instance. The webpage for delete processing can be configured to receive a selection of the WWPN or the delete target instance and an instruction to execute the delete processing.

[0063] In Step 601, the control program 241 detects the type of the target compute node having the WWPN in the received request, which is provided by the field of type 403

in the WWPN table 206. When the control program 241 detects the type is virtualized compute node 113, the control program 241 proceeds to Step 602. When the control program 241 detects the type is physical compute node 112, the control program 241 proceeds to Step 605.

[0064] In Step 602, the control program 241 releases the WWPN 202. Specifically, the control program 241 updates the field of state 401 in the WWPN table 206 (e.g. from "used" to "unused") and deletes the entry (the record) in the instance table 207 where the field of WWPN 202 is equal to the WWPN 202.

[0065] In Step 603, the control program 241 sends, to the target virtualized compute node 113, a request to delete the LPAR 120 which has the WWPN 202. The hypervisor 106 in the compute node 113 delete the LPAR in response to the request and retrieves the physical computing resource subset 105 as well as the virtual FC port 107 for future usage.

[0066] In Step 604, the hypervisor 106 (or the control program 241) generates a request to destroy the connection between the WWPN 202 and the attached volume 201, and sends the request to the storage system 111 having the attached volume 201.

[0067] In Step 605, the control program 241 releases all WWPNs 202 on the target physical compute node 112. In Step 506 of FIG 8, the control program 241 reserves all WWPNs on the physical compute node 112 for the new instance deployment on the physical compute node 112. Therefore, all the WWPNs reserved in Step 506 should be released to enable next instance deployment on the physical compute node 112.

[0068] In Step 606, the control program 241 generates a power-off command to turn off the target physical compute node 112, and sends the command to the physical compute node 112.

[0069] In Step 607, the control program 241 requests the storage system 111 to disconnect the WWPN 202 and the volume 201, and to delete the volume 201. The storage system 111 deletes the connection between the WWPN 202 and the volume 201 as well as the volume 201 to release storage space.

[0070] Through the above processing, the instance 205 and the attached volume can be deleted automatically after the user 302 conducts delete selection via the webpage for delete processing.

[0071] FIGs. 12 and 13 illustrate processing for migrating an instance 205 (LPAR 120) from a source virtualized compute node 113 to a destination virtualized compute node 113. Specifically, FIG. 12 is a flow chart for migrating an instance 205 (LPAR 120) between virtualized compute nodes 113. FIG. 13 is a sequence chart for migrating an instance 205 (LPAR 120) between virtualized compute nodes 113. Steps of FIG. 12 are executed when the control program 241 receives an instance migration request from the user 302 via a webpage for migration processing like the catalog 400. The instance

migration request can include at least one of the WWPN stored in the migration target instance, the identifier of the migration target instance, the identifier of the source virtualized compute node, and the identifier of the destination virtualized compute node. The webpage for migration processing can be configured to receive a selection of the WWPN (or the migration target instance), destination virtualized compute node and an instruction to execute the migration processing. The two virtualized compute node 113 can be in the same (single) group 119. The control program 241 can refer to group management table in the storage unit of the control node. The group management table shows, for each group 119, identifiers of compute nodes and storage systems in the group.

[0072] In Step 701, the control program 241 updates the field of physical server 402 in the WWPN table 206. Specifically the control program 241 selects an unused WWPN 202 of the target virtualized compute node 113 and swaps the field of physical server 402 of the two WWPNs 202.

[0073] In Step 702, the control program 241 sends information including information of the LPAR 120 in the migration target instance 205 and information of the destination virtualized compute node 113 to the hypervisor 106 of the source virtualized compute node 113.

[0074] In Step 703, the hypervisor 106 executes the migration of the LPAR 120 between the two virtualized compute nodes 113. The migration includes swapping the two WWPNs 202 between the hypervisors 106 on the two virtualized compute nodes 113.

[0075] Through the above processing, the instance 205 can be migrated between virtualized compute nodes 113 without any modification in the storage system 111 after the user 302 conducts migration selection via the webpage for migration processing. Through the above processing, the physical server change of the instance 205 can be observed by the user 302 via the change of the physical server field 402 in the WWPN table 206.

[0076] FIGs. 14 and 15 illustrate processing for initializing the WWPN table 206. Specifically FIG. 14 is a flow chart of collecting WWPNs from compute nodes 112/113 to initialize the WWPN table 206. FIG. 15 is a sequence chart of collecting WWPNs from compute nodes 112/113 to initialize the WWPN table 206. Steps of FIG. 14 are executed when the control program 241 is initialized.

[0077] In Step 801, the control program 241 accesses the compute node 112 or 113.

[0078] In Step 802, the control program 241 detects the type of the compute node, which is provided by the hypervisor 106 or management software such as baseboard management controller (BMC) on the compute node. When the control program 241 detects the compute node is virtualized compute node 113, the control program 241 proceeds to Step 803. When the control node detects the compute node is physical compute node 112, the control program 241 proceeds to Step 804.

[0079] In Step 803, the control program 241 gets WWPNs 202 from the hypervisor 106 of the virtualized compute node. As the compute node is the virtualized compute node 113, the hypervisor 116 holds all the WWPNs 202 on the node. The control program 241 can get all the WWPNs 202 that the hypervisor 116 holds by communicating with the hypervisor 116.

[0080] In Step 804, the control program 241 gets WWPNs 202 from software tool. The software tool such as BMC is pre-installed and provides an interface to get the information of physical resource and configure the physical resource.

[0081] In Step 805, the control program 241 registers the received WWPNs 202 into the WWPN table 206.

[0082] Through the above processing, the control node can collect all WWPNs 202 from all the virtualized compute nodes 113 and the physical compute nodes 112 to initialize the WWPN table 206 automatically when the control program 241 is initialized.

[0083] FIGs. 16 and 17 illustrate processing for updating the WWPN table 206 and instance table 207 based on heartbeat message. The heartbeat message is message sent timely from each compute node to control node. FIG. 16 is a flow chart for updating tables based on heartbeat message. FIG. 17 is a sequence chart for updating tables based on heartbeat message. Steps of FIG. 16 are executed by the control program 241 for each compute node after the node is added.

[0084] In Step 901, the control program 241 sets up an alive timer for a new added compute node 112 or 113. Specifically, the control program 241 sets up the alive timer with a valid time and the alive timer conducts self-count independently. The alive timer becomes expired and invalid if the timer is not reset during the preset valid time.

[0085] In Step 902, the control program 241detects the alive timer for a compute node 112 or 113 is valid or not. The detection can be achieved by checking the alive timer counter exceed the pre-set valid timer. When the control program 241 detects the alive timer is valid, the control program 241 proceeds to Step 903. When the control program 241 detects the said alive timer is invalid, the control program 241 proceeds to Step 909.

[0086] In Step 903, the control program 241 detects if the control program 241 has received heartbeat message from the compute node during the valid time period. When the control program 241 detects it received heartbeat message from the compute node, the control node proceeds to Step 904. When the control program 241 detects it did not receive heartbeat message from the compute node, the control node proceeds to Step 908.

[0087] In Step 904, the control program 241 gets WWPN information from the received heartbeat message. The compute node generates the heartbeat message including at least one of all WWPNs on the compute node and the state information of all the

WWPNs.

[0088] In Step 905, the control program 241 detects the received WWPN state is different from the WWPN table 206 or not. The WWPN on the compute node can be changed in case such as changing a physical FC port. The control program 241 can recognize the change by matching WWPN information from heartbeat message and the WWPN table 206. When the control program 241 detects there is difference between the received WWPN information and the WWPN table 206, the control program 241 proceeds to Step 906. When the control program 241 detects there is no difference between the received WWPN information and the WWPN table 206, the control program 241 proceeds to Step 907.

[0089] In Step 906, the control program 241 updates the WWPN table 206. Specifically the control program 241 updates the changed WWPN in the WWPN table 206 based on the different WWPN information detected in Step 905.

[0090] In Step 907, the control program 241 resets the alive timer. As the control program 241 received heartbeat message from the compute node, the compute node is considered alive during the past period and the control program 241 resets the alive timer.

[0091] In Step 908, the control program 241 sleeps. To save computing resources, the control program 241 proceeds to sleep for a while instead of polling.

[0092] In Step 909. the control program 241 deletes all WWPNs located on the compute node from the said WWPN table 206. As the alive timer is invalid, control program 241 considers the compute node is no longer available due to some reasons such as hardware failure. The control program 241 updates the WWPN table 206 by deleting the invalid WWPNs located on the compute node.

[0093] In Step 910, the control program 241 deletes all instances located on the compute node from the instance table 207. As the compute node is considered to be unavailable in Step 909, all instances 205 on the compute node are considered invalid. The control program 241 updates the instance table 207 by deleting the records of all the instances 205 located on the compute node.

[0094] Through the above processing, the control program 241 can automatically update the WWPN table and the instance table when there are changes either for the physical WWPN connection or the compute node.

[0095] FIG. 18 is a block diagram illustrating the mapping between the virtual image 303 and template volumes. Specifically FIG. 18 illustrates the mapping between a single virtual image and multiple template volumes.

[0096] Each of the copied (attached) volumes 201A can be generated from one of the template volumes 201T by technique provided by one of the storage systems 111 such copy-on-write.

[0097] The contents of the template volumes 201T within one storage system 111 are different. The contents of the template volumes 201T within different storage systems 111 can be the same. The template volumes 201T with the same contents but locate in different storage system 111 are mapped to the same virtual image 303.

[0098] The contents of the template volume 201T include OS, software and other possible configurations to make a different execution environment for user applications.

[0099] According to the above mentioned explanation, the following can be conducted.

[0100] The control program 241 collects all the WWPN from all the virtualized compute nodes 113 (hypervisors 106) and all the physical compute nodes 112.

[0101] The control program 241 receives a selection of OS type (virtual image) and the number of instances from the user 302 via the catalog 400. The catalog 400 is an example of a user interface. The catalog 400 is configured to receive simple information such as OS type and the number of instances.

[0102] The control program 241 reserves unused WWPNs among the collected WWPNs. The number of reserved WWPNs is the same as the number of instances input by the user 302. If there is a WWPN whose type is "virtual" (that is, a WWPN collected from a virtualized compute node 113) among the reserved WWPNs, the control program 241 generate a LPAR definition request. The LPAR definition request is a request to define a new LPAR with the reserved WWPN. The LPAR definition request includes the WWPN. The control program 241 sends the LPAR definition request to the hypervisor 106 in the virtualized compute node 113. The hypervisor 106 receives the LPAR definition request and defines a new LPAR in response to the request.

[0103] The storage systems 111 have template volumes 201T (original volumes) respectively storing virtual images. The control program 241 generates a building request. The building request is a request to build a connection between reserved WWPNs and volumes storing the target virtual image. The volumes include at least one of a template volume 201T and an attached volume 201A. The volumes typically include attached volumes 201A. The target virtual image is the virtual image including OS (or metadata) whose OS type is the designated OS type. The building request includes the reserved WWPNs and information. The information can include at least one of the designated OS type and locations of the (attached) volumes each storing the target virtual image. The control program 241 sends the building request to at least one of the storage systems 111 having the volumes each storing the target virtual image. If the number of volumes storing the virtual images is smaller than the number of the reserved WWPNS, at least one storage system 111 having the template volume 201T can generate, e.g. in response to the request, attached volumes of the template volume 201T in a storage system in the same group 119. In response to the building request, the storage systems having the attached volumes build a connection the reserved

WWPNs and the volumes. There can be as many volumes storing the target virtual image as the reserved WWPNs, therefore the volumes can be respectively connected to the reserved WWPNs.

[0104] The present invention can provide image-liked easy management, deploying performance with volumes and compatible management between virtualized environments and bare metal environments.

[0105] While the present invention has been described in detail and pictorially in the accompanying drawings, the present invention is not limited to such detail but covers various obvious modifications and equivalent arrangements.

[0106] For example, the WWPNs and the volumes which are connected each other can be in the same group 119.

[0107] Furthermore, for example, the processing for deleting an instance shown in FIGs. 10 and 11 can start when the LPAR included in the instance is shut down.

[0108] Furthermore, for example, the present invention can be adapted to another kind of systems such as iSCSI system. In the iSCSI system, a control node doesn't have to collect specific (unique) identifiers (e.g. WWPNs) used to access volumes. The control node can generate the specific identifiers and respectively allocate them to compute nodes.

Reference Signs List

[0109] 110: Control node

111: Storage system

112: Physical compute node

113: Virtualized compute node

Claims

[Claim 1] A method for managing instances in a computer system,
each instance being a set of computing resources with an OS (Operating System) running on it,
the computer system including a virtualized computing environment which is a virtualized compute node executing a virtualization program for defining virtualized servers, a bare metal environment which is a physical compute node as a bare metal server with an OS, and block storage systems configured to provide volumes storing images associated with OSs,
the method comprising:
reserving one or more unused identifiers for one or more new instances, among identifiers which are used to access volumes,
sending one or more building requests to one or more storage systems, the building requests being requests to build a connection between the reserved identifiers and one or more volumes, the storage systems having the one or more volumes, each of the volumes being a template volume storing a target image or an attached volume of the template volume, the target image being is the image associated with the OS for the new instances,
activating each of the new instances to boot, using the image, the OS for the new instance associated with a reserved identifier among the reserved identifiers, the image being in the volume connected with the reserved identifier.

[Claim 2] The method of claim 1, further comprising:
providing a GUI (Graphical User interface) based on webpage configured to receive a selection of OS type and the number of instances from a user,
wherein the target image is the image associated with the OS whose OS type is the same as the OS type selected by the user,
wherein the number of reserved identifiers are the same as the number of instances input by the user.

[Claim 3] The method of claim 2, wherein the volumes connected with the reserved identifiers include copied volumes of the volume storing the target image.

[Claim 4] The method of claim 3, wherein the copied volumes are prepared in response to the building request.

[Claim 5] The method of claim 3,
wherein the copied volumes are deleted when target instances are deleted, and
wherein the target instances are instances which are associated with identifiers connected with the copied volumes.

[Claim 6] The method of claim 1,
wherein at least one of the virtualized compute nodes is configured to define a LPAR (Logical PARtition) with OS by a hypervisor,
wherein the hypervisor is the virtualization program,
wherein the LPAR is the virtualized server,
wherein the identifiers are WWPNs (World Wide Port Names).

[Claim 7] The method of claim 6, the WWPNs are WWPNs collected from all available compute nodes.

[Claim 8] The method of claim 6, further comprising:
migrating an instance between a source compute node and a destination compute node by moving WWPN allocated to the LPAR in the source compute node to the destination compute node.

[Claim 9] The method of claim 6, further comprising:
releasing a target WWPN which is a WWPN allocated to a first delete target instance,
updating the WWPN state of the target WWN to a state which means unused WWPN,
requesting the hypervisor to delete the LPAR associated with the target WWPN
requesting the storage system to disconnecting the target WWPN and the volume and to delete the volume.

[Claim 10] The method of claim 6, further comprising:
releasing the all WWPNs on the physical compute node where a second delete target instance is located,
updating the WWPN states of the all WWPNs to states each which means unused WWPN,
sending, to the physical compute node, a power-off command to turn off the physical compute node,
requesting the storage system to disconnect the WWPN and the volume and to delete the volume.

[Claim 11] The method of claim 6, further comprising:
if any heartbeat message is not received from a compute node during a pre-set time period, deleting the WWPNs and information regarding

instances located in the compute node.

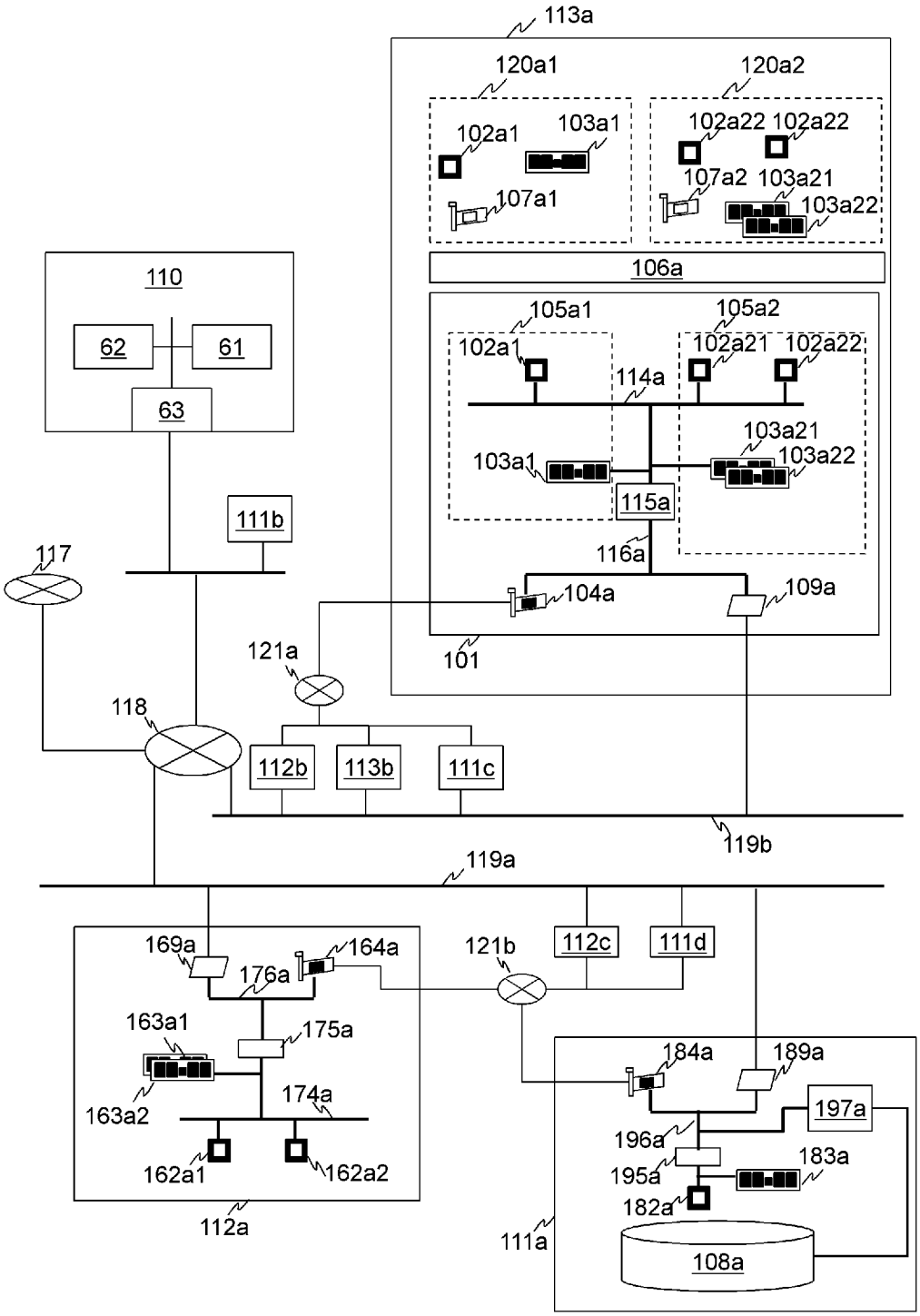
[Claim 12] A system for managing instances, each instance being a set of computing resources with an OS (Operating System) running on it, the system comprising:
a computer system including a virtualized computing environment which is a virtualized compute node executing a virtualization program for defining virtualized servers, a bare metal environment which is a physical compute node as a bare metal server with an OS, and block storage systems configured to provide volumes storing images associated with OSs, and
a management system coupled to the computer system,
the management system being configured to
reserve one or more unused identifiers for one or more new instances, among identifiers which are used to access volumes,
send a building request to one or more storage systems, the building request being a request to build a connection between the reserved identifiers and one or more volumes, the storage systems having the one or more volumes, each of the volumes storing a target image which is the image associated with the OS for the new instances,
activate each of the new instances to boot, using the image, the OS for the new instance associated with a reserved identifier among the reserved identifiers, the image being in the volume connected with the reserved identifier.

[Claim 13] A management system for managing instances in a computer system, each instance being a set of computing resources with an OS (Operating System) running on it, the system comprising:
the computer system including a virtualized computing environment which is a virtualized compute node executing a virtualization program for defining virtualized servers, a bare metal environment which is a physical compute node as a bare metal server with an OS, and block storage systems configured to provide volumes storing images associated with OSs, and
the management system comprising:
an interface coupled to the computer system,
a storage unit storing management information which is used to manage the instances, and
a processor coupled to the interface and the storage unit,
the processor being configured to

reserve one or more unused identifiers for one or more new instances, among identifiers which are used to access volumes,
send a building request to one or more storage systems, the building request being a request to build a connection between the reserved identifiers and one or more volumes, the storage systems having the one or more volumes, each of the volumes storing a target image which is the image associated with the OS for the new instances,
activate each of the new instances to boot, using the image, the OS for the new instance associated with a reserved identifier among the reserved identifiers, the image being in the volume connected with the reserved identifier.

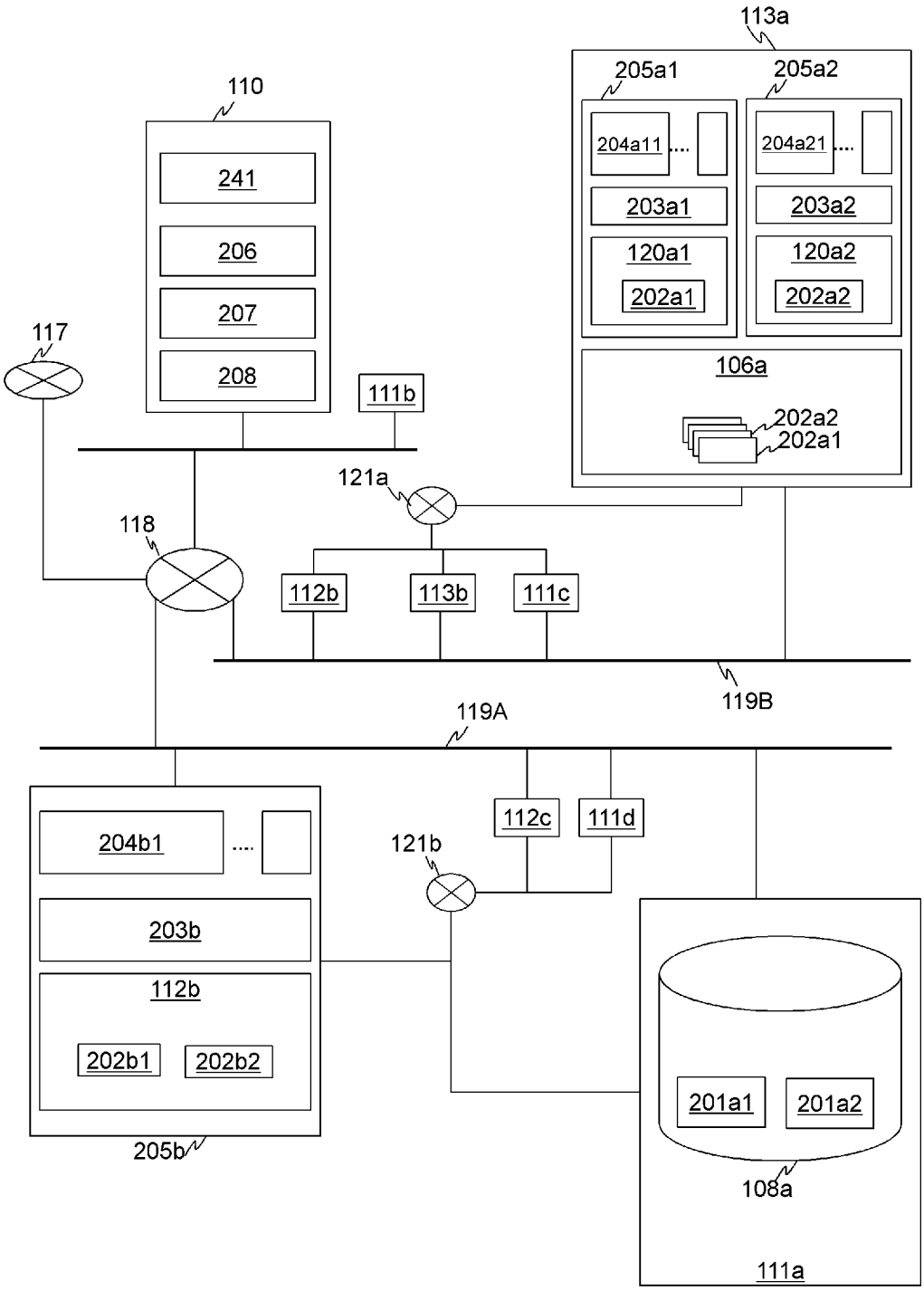
[Fig. 1]

FIG. 1


113a
120a1
120a2
102a1
103a1
102a22
102a22
107a2
103a21
107a1
103a22
106a
110
62
61
63
105a1
105a2
102a1
102a21
102a22
114a
103a21
103a1
103a22
115a
111b
116a
117
104a
109a
121a
101
118
112b
113b
111c
119b
119a
169a
164a
121b
112c
111d
176a
163a1
175a
184a
189a
174a
197a
163a2
196a
183a
195a
162a1
162a2
182a
111a
108a
112a

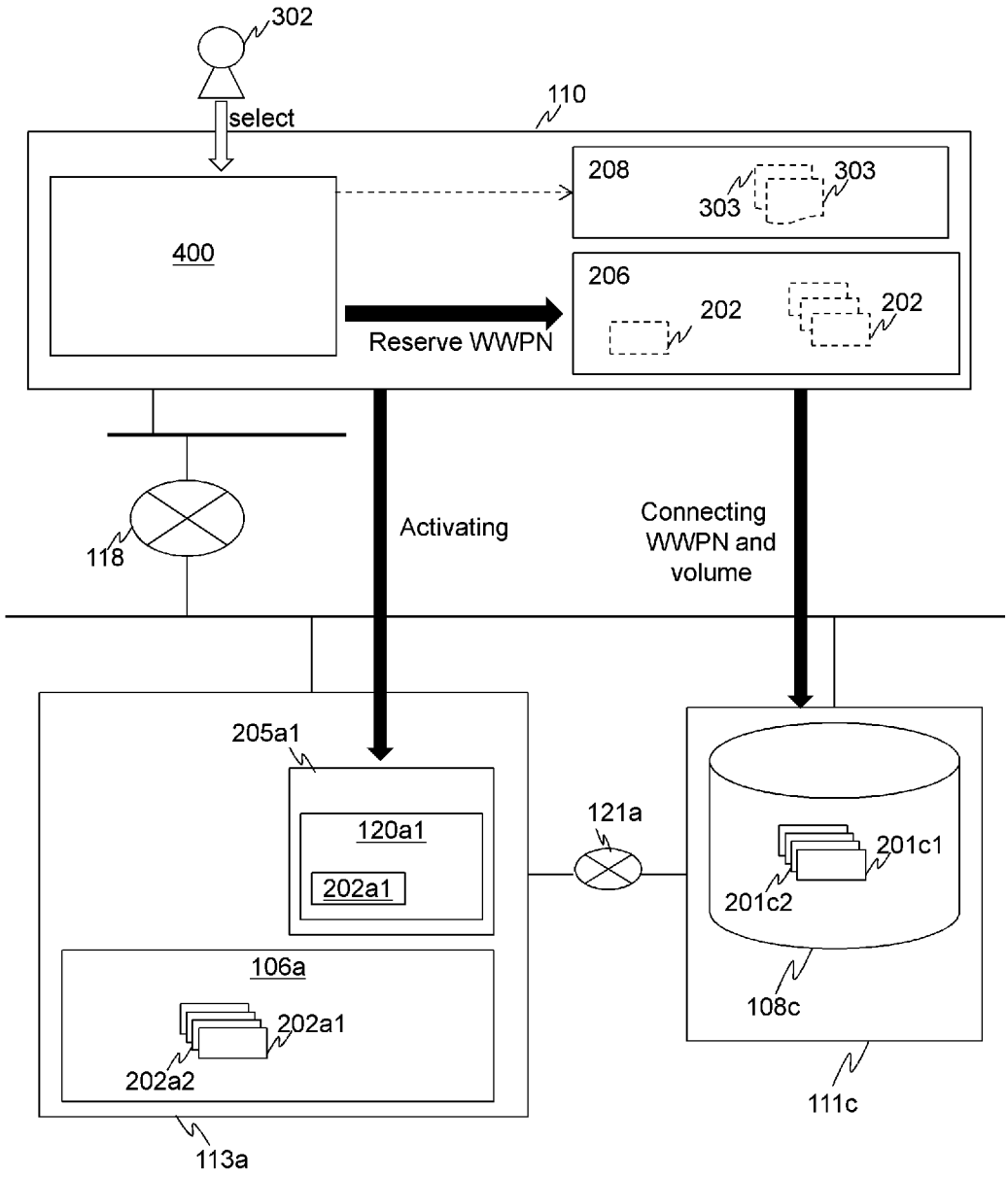
[Fig. 2]

FIG. 2


113a
205a1
205a2
110
204a11
204a21
241
203a1
203a2
206
120a1
120a2
207
202a1
202a2
208
106a
117
111b
202a2
202a1
121a
118
112b
113b
111c
119B
119A
204b1
112c
111d
121b
203b
112b
202b1
202b2
205b
201a1
201a2
108a
111a

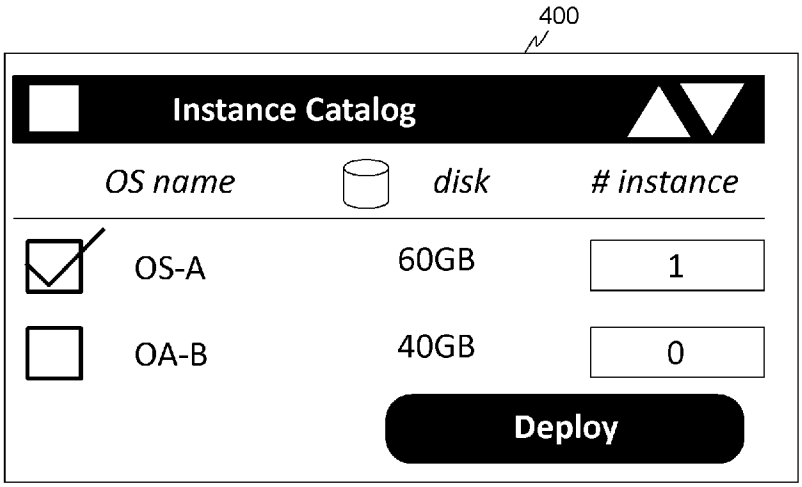
[Fig. 3]

FIG. 3


302
select
110
208
303
303
400
206
202
202
Reserve WWPN
118
Activating
Connecting
WWPN and
volume
205a1
120a1
202a1
121a
201c1
201c2
106a
202a1
202a2
108c
111c
113a

[Fig. 4]

FIG. 4


400
Instance Catalog
OS name
disk
instance
OS-A
60GB
1
OA-B
40GB
0
Deploy

[Fig. 5]

FIG. 5

WWPN table
206

202 WWPN	401 State	402 PS_ID	403 Type	461 Sys_ID
23140000876ec002	idle	Compute1	Virtual	sys01, sys02
23140000876ec006	used	Compute2	Physical	sys02,sys03
…	…	…	…	…

[Fig. 6]

FIG. 6

Instance table
207

404 Name	405 UUID	406 Description	202 WWPN	407 Attached volume (sys_ID:vol_ID)	408 Template volume (sys_ID:vol_ID)
Tokyo	3245	OS-C	23140000876ec006	sys01:003	sys01:01
…	…	…	…	…	…

[Fig. 7]

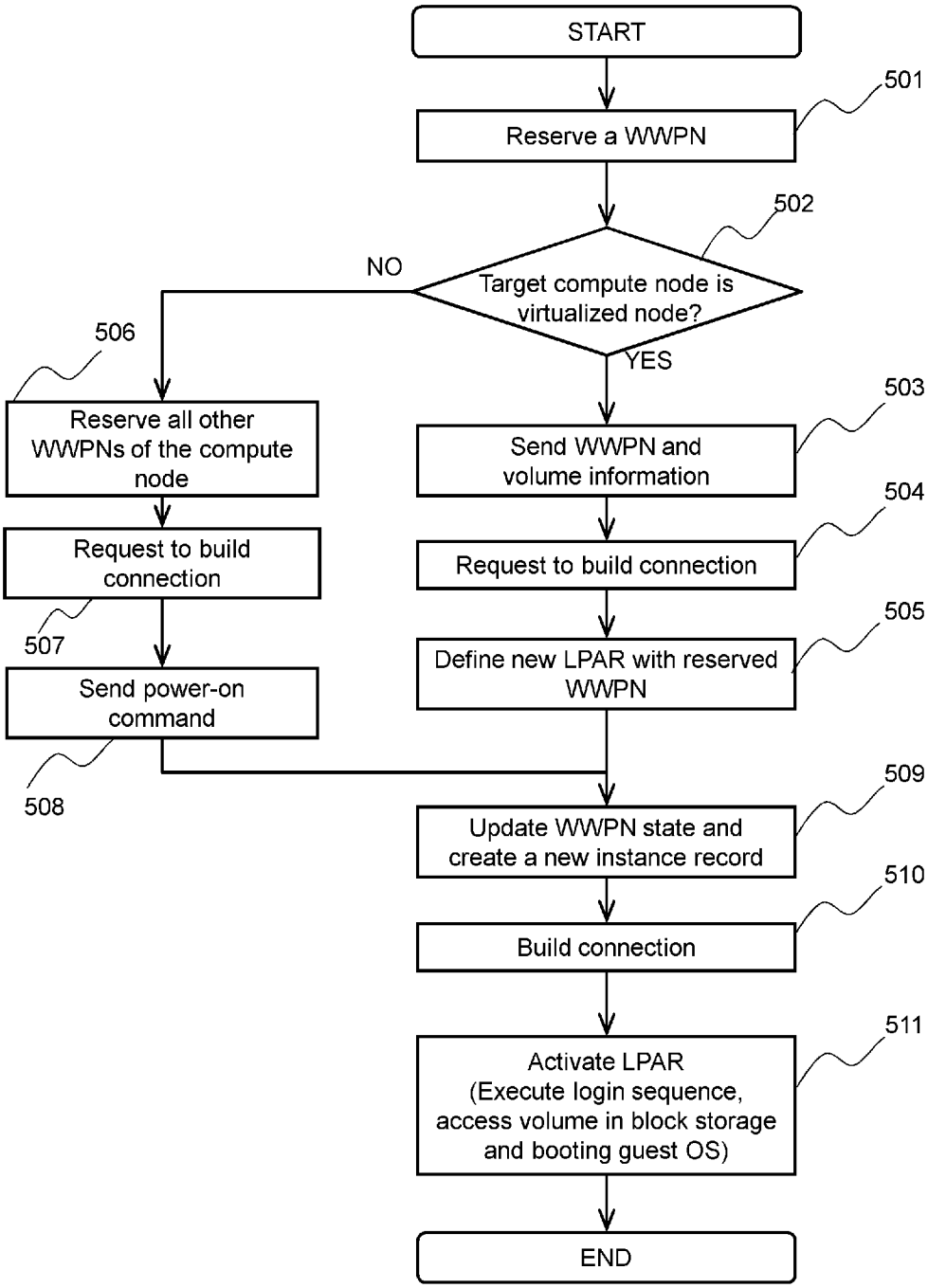
FIG. 7

Image table
208

409 Name	410 UUID	411 Description	412 Volume list
Client server	314f-a3241-324	OS-C APP-AI	sys01:01 sys02:01
…	…	…	…

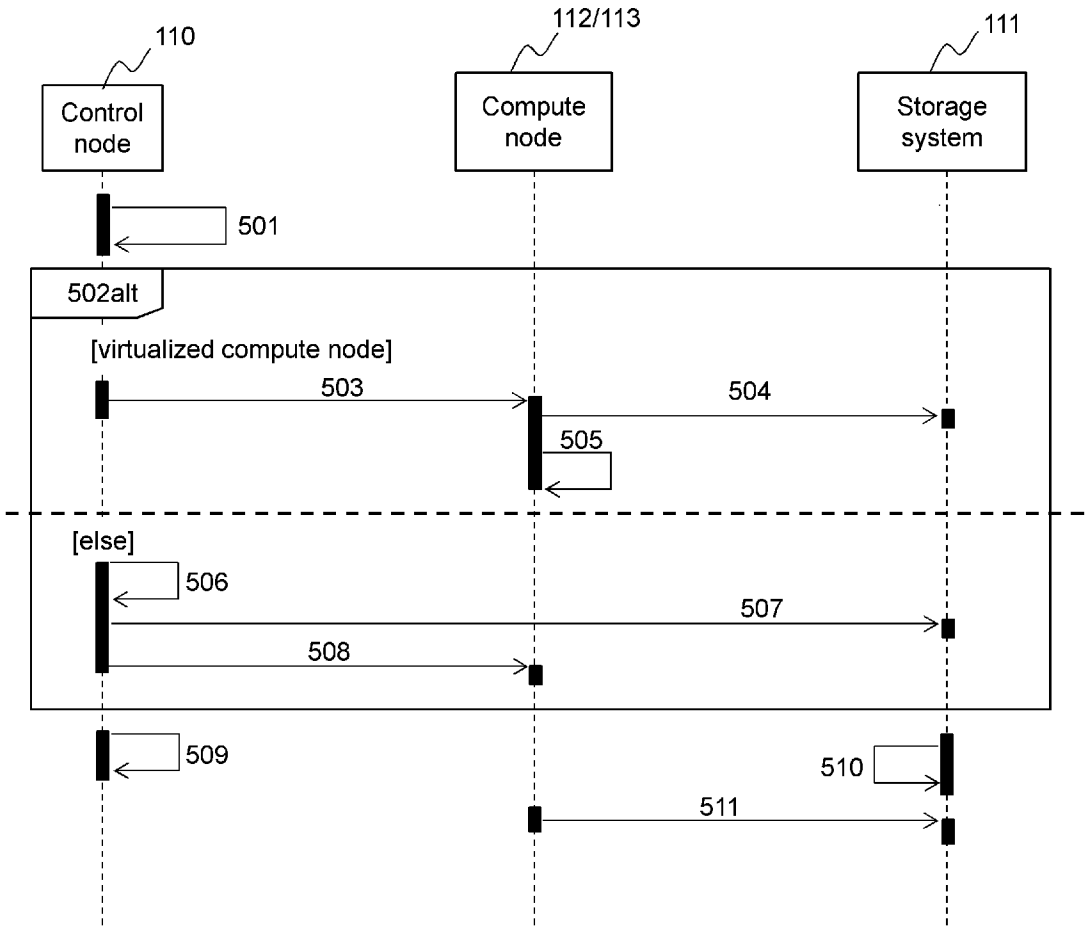
[Fig. 8]

FIG. 8


START
501
Reserve a WWPN
502
Target compute node is virtualized node?
NO
YES
506
Reserve all other WWPNs of the compute node
Request to build connection
507
Send power-on command
508
503
Send WWPN and volume information
504
Request to build connection
505
Define new LPAR with reserved WWPN
509
Update WWPN state and create a new instance record
510
Build connection
511
Activate LPAR (Execute login sequence, access volume in block storage and booting guest OS)
END

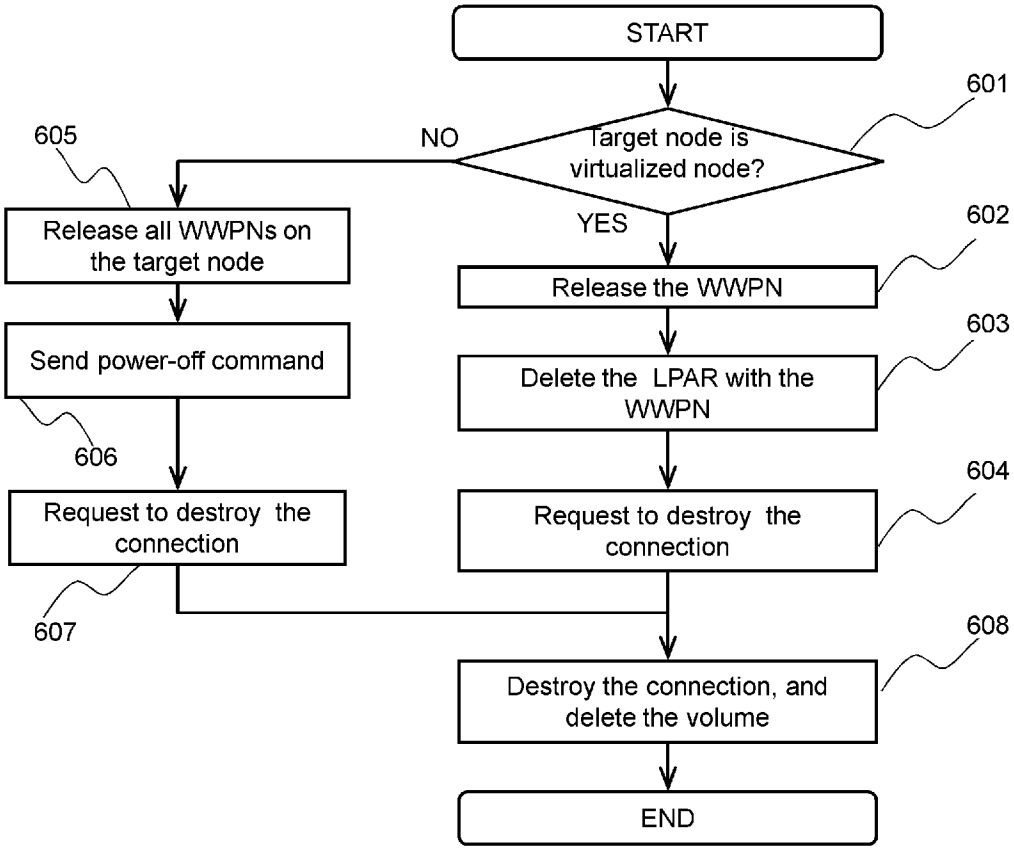
[Fig. 9]

FIG. 9


110
Control node
112/113
Compute node
111
Storage system
501
502alt
[virtualized compute node]
503
504
505
[else]
506
507
508
509
510
511

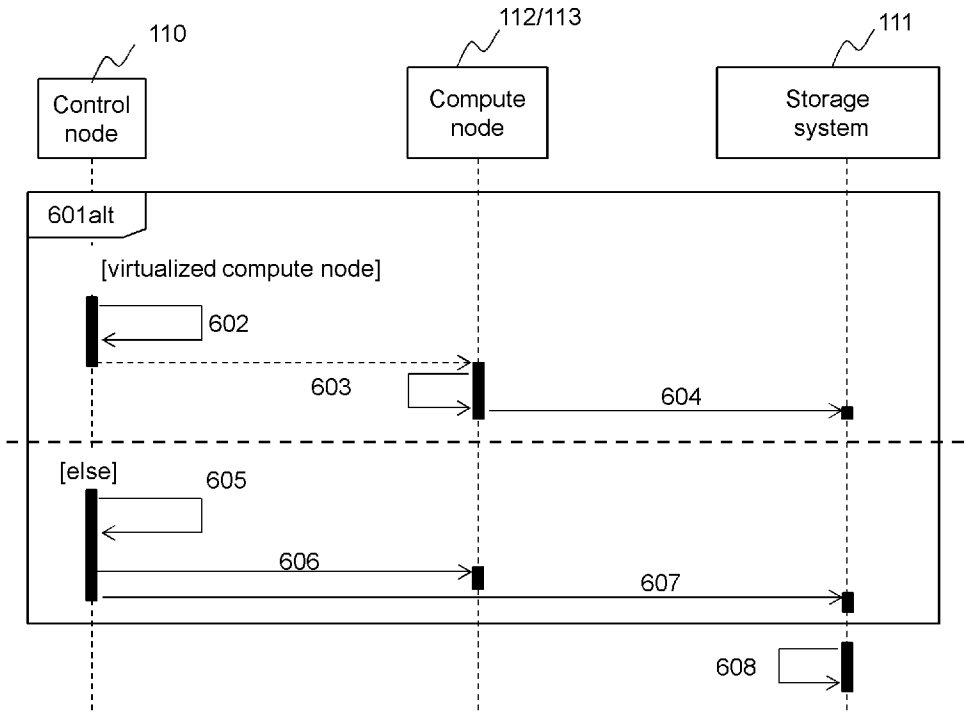
[Fig. 10]

FIG. 10


START
601
Target node is virtualized node?
NO
YES
605
Release all WWPNs on the target node
Send power-off command
606
Request to destroy the connection
607
602
Release the WWPN
603
Delete the LPAR with the WWPN
604
Request to destroy the connection
608
Destroy the connection, and delete the volume
END

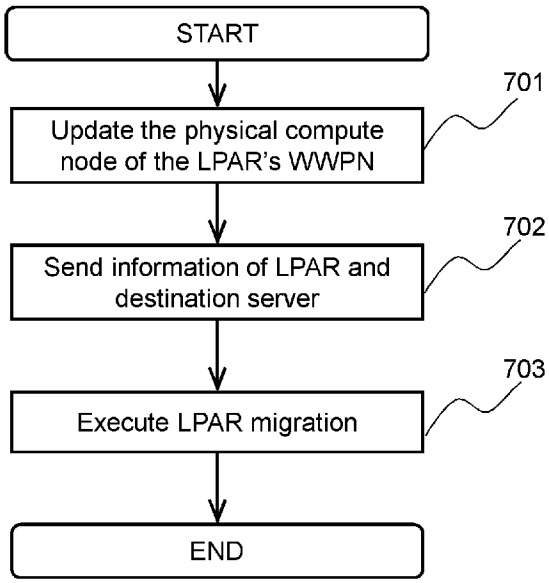
[Fig. 11]

FIG. 11



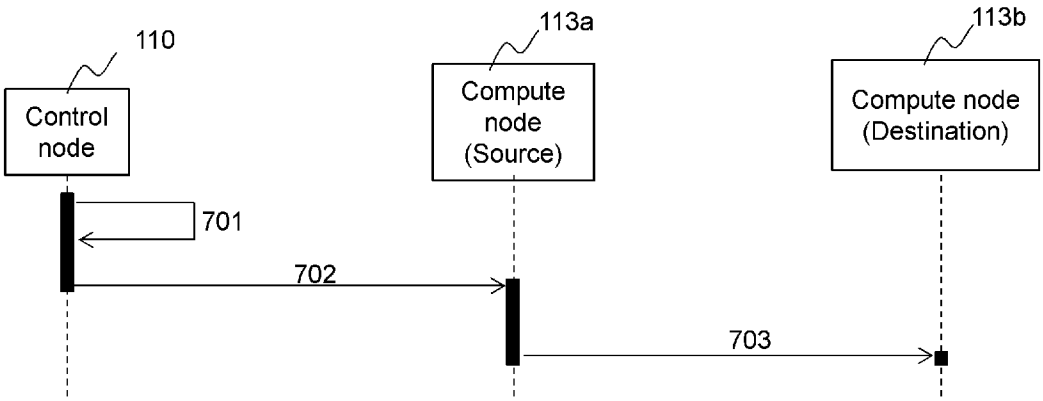
[Fig. 12]

FIG. 12



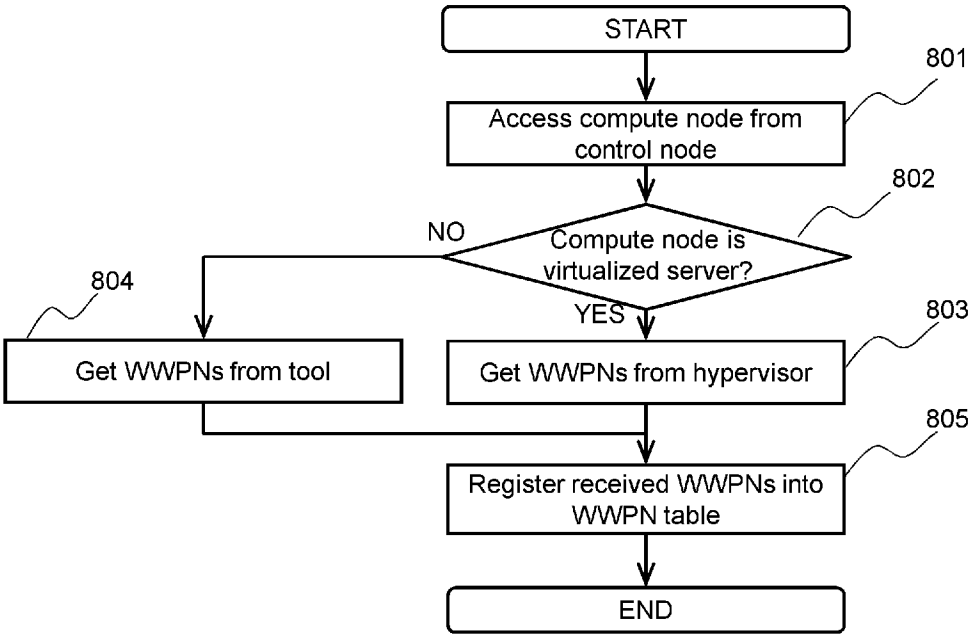
[Fig. 13]

FIG. 13



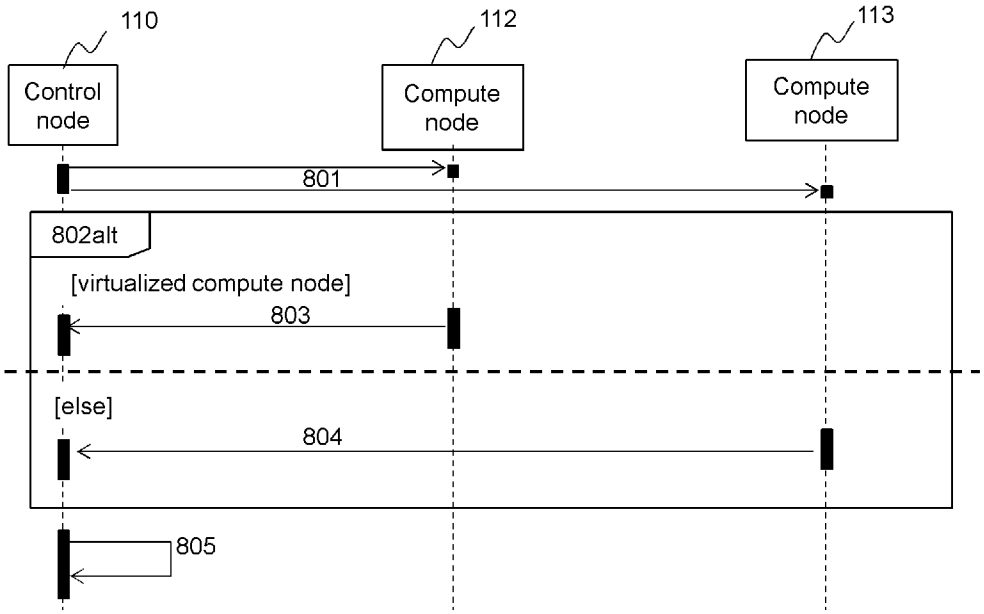
[Fig. 14]

FIG. 14



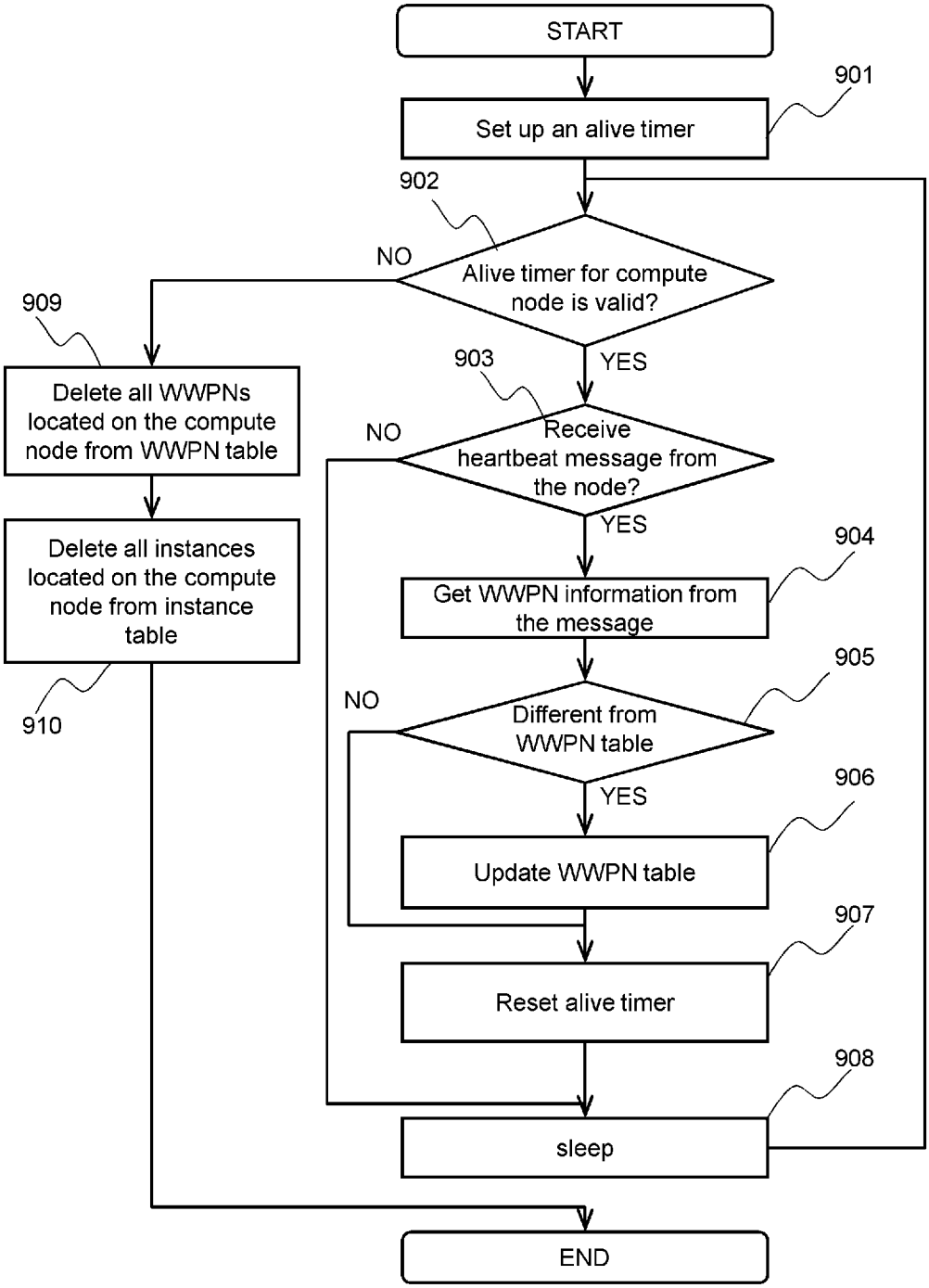
[Fig. 15]

FIG. 15


110
112
113
Control node
Compute node
Compute node
801
802alt
[virtualized compute node]
803
[else]
804
805

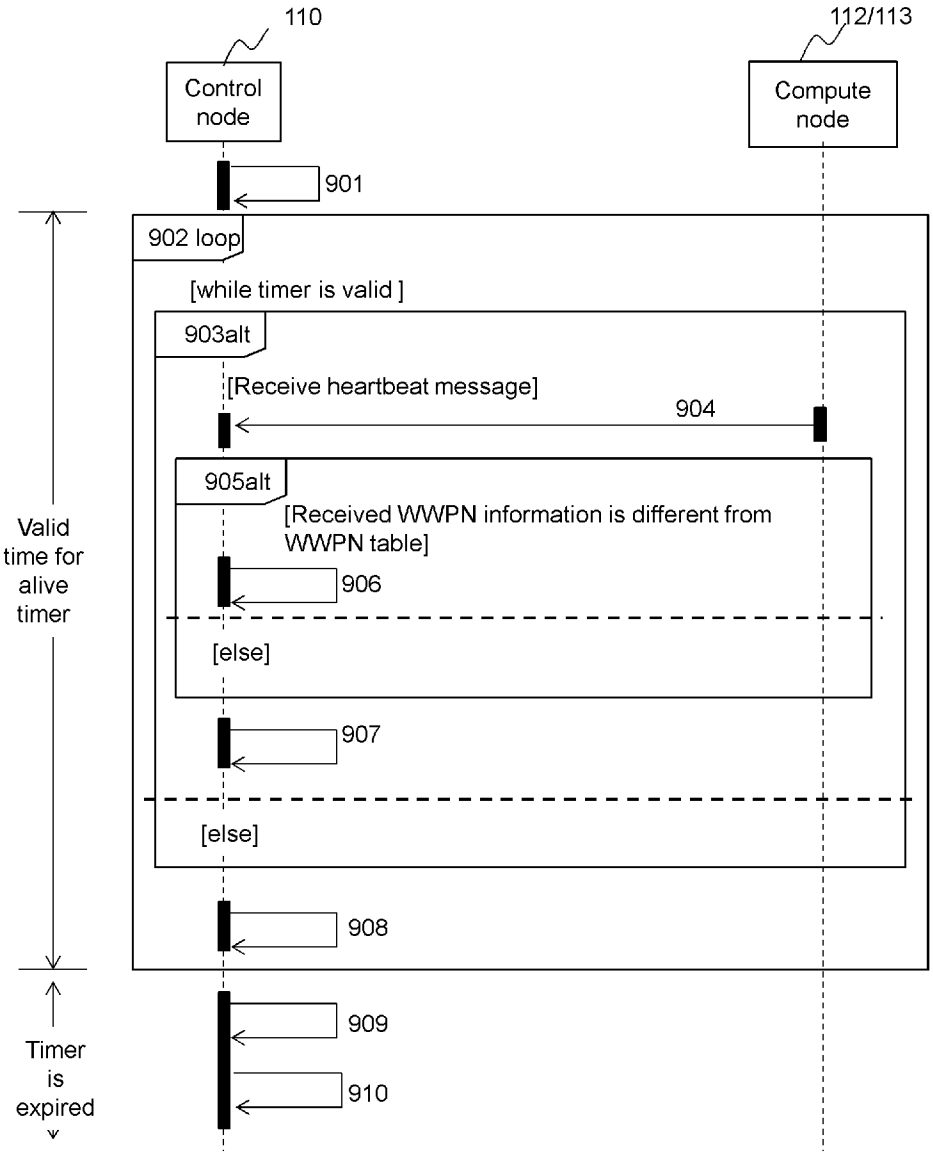
[Fig. 16]

FIG. 16


START
901
Set up an alive timer
902
Alive timer for compute node is valid?
NO
YES
909
Delete all WWPNs located on the compute node from WWPN table
903
Receive heartbeat message from the node?
NO
YES
Delete all instances located on the compute node from instance table
904
Get WWPN information from the message
910
905
Different from WWPN table
NO
YES
906
Update WWPN table
907
Reset alive timer
908
sleep
END

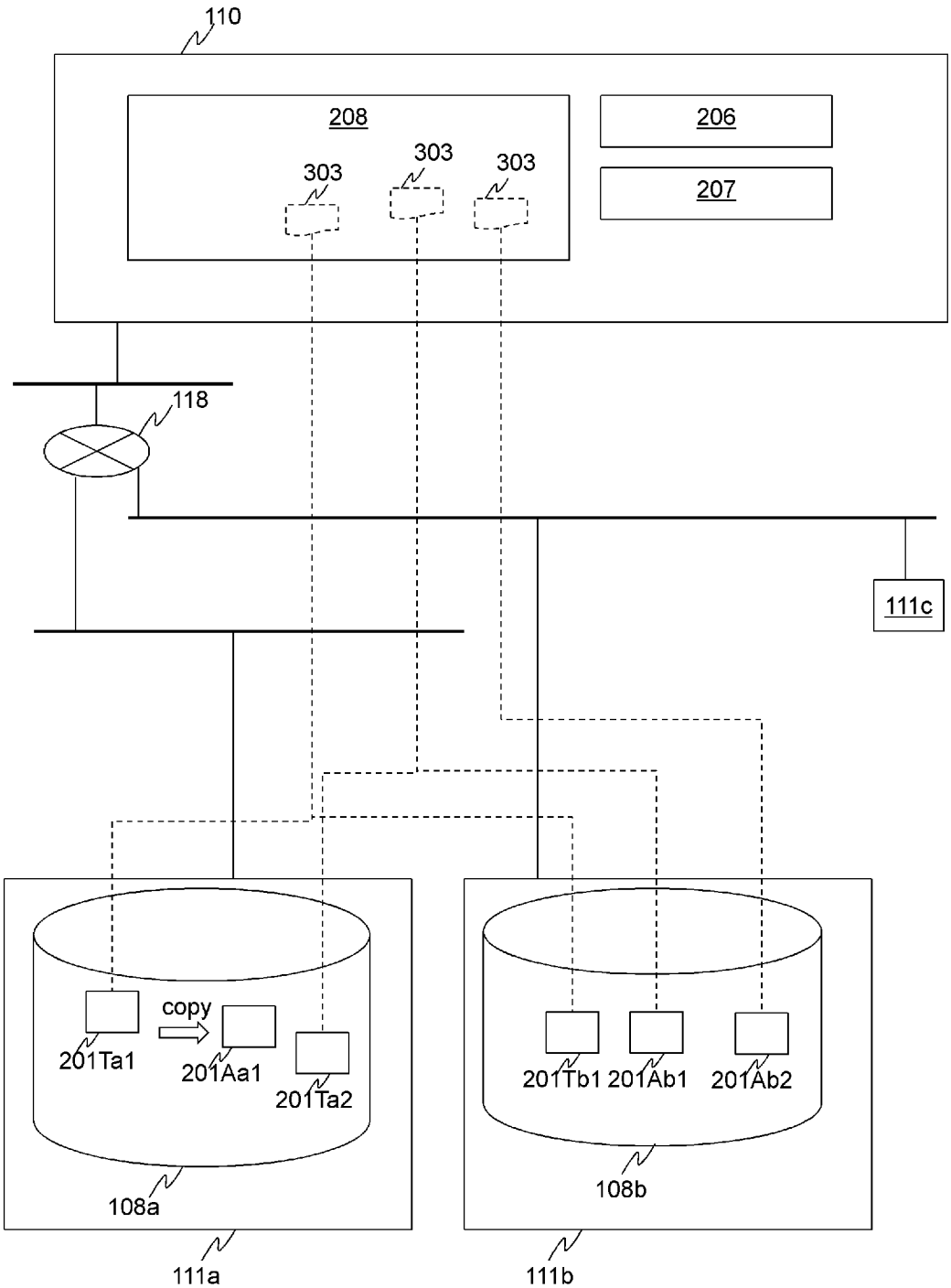
[Fig. 17]

FIG. 17


110
Control node
112/113
Compute node
901
902 loop
[while timer is valid]
903alt
[Receive heartbeat message]
904
905alt
[Received WWPN information is different from WWPN table]
906
[else]
907
[else]
908
909
910
Valid time for alive timer
Timer is expired

[Fig. 18]

FIG. 18


110
208
206
207
303
303
303
118
111c
copy
201Ta1
201Aa1
201Ta2
108a
111a
201Tb1
201Ab1
201Ab2
108b
111b

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2015/004783

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl. G06F9/445(2006.01)i, G06F9/46(2006.01)i

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)

Int.Cl. G06F9/445, G06F9/46

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 1971-2015
Registered utility model specifications of Japan 1996-2015
Published registered utility model applications of Japan 1994-2015

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2013/0097295 A1 (INTERNATIONAL BUSINESS MACHINES CORPORATION) 2013.04.18, Pars. [0021]-[0024],[0032]-[0034], Figs. 1-4 & US 2013/0219040 A1	1-13
A	US 2009/0282142 A1 (FUJITSU LIMITED.) 2009.11.12, Pars. [0060],[0069],[0079]-[0089], Figs. 1-5 & JP 2009-271834 A	1-13
A	US 2013/0346615 A1 (VMWARE, INC.) 2013.12.26, Full text; Figs. 1-9B (No Family)	1-13

☐ Further documents are listed in the continuation of Box C.

☐ See patent family 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 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	Date of mailing of the international search report
21.12.2015	28.12.2015

Name and mailing address of the ISA/JP	Authorized officer	
Japan Patent Office 3-4-3, Kasumigaseki, Chiyoda-ku, Tokyo 100-8915, Japan	KINUGAWA, Hironobu Telephone No. +81-3-3581-1101 Ext. 3545	5B 9557