



- (51) International Patent Classification:
G06F 15/16 (2006.01)
- (21) International Application Number:
PCT/US2014/036556
- (22) International Filing Date:
2 May 2014 (02.05.2014)
- (25) Filing Language: English
- (26) Publication Language: English
- (71) Applicant: **HITACHI, LTD.** [JP/JP]; 6-6, Marunouchi 1-chome, Chiyoda-ku, Tokyo 100-8280 (JP).
- (72) Inventor: **DEGUCHI, Akira**; 3595 Granada Ave, Unit 276, Santa Clara, California 95051 (US).
- (74) Agents: **LEUNG, Chun-Pok** et al.; Mattingly & Malur, PC, 1800 Diagonal Road, Suite 370, Alexandria, Virginia 22314 (US).
- (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, LT, 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, 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))

- (54) Title: STORAGE PROVISIONING METHOD AND APPARATUS USING DEVICE VIRTUALIZATION IN HETEROGENEOUS STORAGE ENVIRONMENT

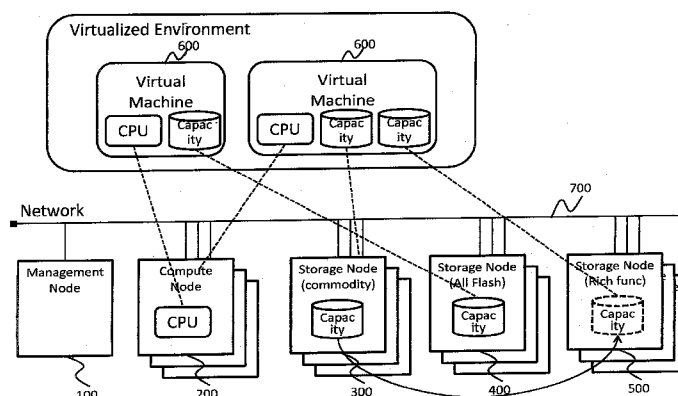


Fig. 1

- (57) Abstract: A management computer, coupled to a plurality of storage nodes, comprises a memory being configured to store first information indicating functionalities which can be executed by each of the storage nodes and second information showing types of storage capacity included in each of the storage nodes; and a processor, in response to a user request, configured to: select a storage node which has a functionality requested by the user request by referring to the first information; check whether the selected storage node has a type of storage capacity requested by the user request by referring to the second information; and provide, if the storage node does not have the type of storage capacity requested by the user request, a virtual device which is mapped to another storage capacity provided by another storage node, said another storage capacity satisfying the type of storage capacity requested by the user request.

STORAGE PROVISIONING METHOD AND APPARATUS USING DEVICE VIRTUALIZATION IN HETEROGENEOUS STORAGE ENVIRONMENT

BACKGROUND OF THE INVENTION

[0001] The present invention relates generally to storage systems and, more particularly, to storage capacity provisioning method and apparatus in a heterogeneous storage environment that includes, for example, enterprise storage systems and commodity server based storage.

[0002] U.S. Patent No. 8,375,223 for secure distributed storage describes commodity server based storage system which is configured from a commodity server that contains CPU, HDD, memory, etc. Ceph, Redhat storage, and Swift are examples of commodity server based storage. U.S. Patent No. 7,133,988 describes a storage device virtualization technology that virtualizes the capacity installed in another storage system physically and provides its capacity to the server as the capacity of the upper storage system. The storage system which has physical capacity is called external storage.

[0003] Currently, there are many types of the enterprise storage system such as, for example, high performance all flash storage system, storage functionalities rich storage system, and archive storage system. Cloud software can treat commodity server based storage and enterprise storage system. As a result, heterogeneous storage environments utilizing both types of storage will be popular. A storage system including commodity server based storage provides different features, performance, cost, access interface, and capacity from those of an enterprise storage system. The user has to choose the storage system depending on the user's requirement. To do

so, the user has to know which storage system has which features, and has to manage the free storage capacity of each storage system. It is a complicated task.

BRIEF SUMMARY OF THE INVENTION

[0004] Exemplary embodiments of the invention provide storage capacity provisioning method and apparatus in a heterogeneous storage environment in such a way that capacity management and feature management of each storage system will be unnecessary. For example, enterprise storage systems and commodity server based storage are provided in the heterogeneous storage environment. When two or more storage features are required and there is no storage having all the required features, the storage capacity of a first storage having one of the features is mapped to a second storage having another feature by using device virtualization technology. The capacity is provided from the second storage to the server. The provided storage can have all the required features. The result is improved ease of use in storage capacity provisioning and storage capacity manageability in the heterogeneous storage environment.

[0005] One aspect of the present invention is directed to a management computer coupled to a plurality of storage nodes. The management computer comprises a memory being configured to store first information indicating functionalities which can be executed by each of the plurality of storage nodes and second information showing types of storage capacity included in each of the plurality of storage nodes; and a processor, in response to a user request, configured to: select a storage node which has a

functionality requested by the user request by referring to the first information; check whether the selected storage node has a type of storage capacity requested by the user request by referring to the second information; and provide, if the storage node does not have the type of storage capacity requested by the user request, a virtual device which is mapped to another storage capacity provided by another storage node, said another storage capacity satisfying the type of storage capacity requested by the user request.

[0006] In some embodiments, the processor is configured: if the storage node has the type of storage capacity requested by the user request, to allocate the storage capacity from the storage node and to provide a storage device from the storage node having the allocated storage capacity; and if the storage node does not have the type of storage capacity requested by the user request, to search said another storage node having said another storage capacity and being connected to the storage node which has the functionality, and to allocate said another storage capacity from said another storage node.

[0007] In specific embodiments, the processor is configured to: check whether the requested functionality is storage media tiering or storage capacity virtualization, and if no, relate the storage device (if the storage node has the type of storage capacity requested by the user request) or the virtual device (if the storage node does not have the type of storage capacity requested by the user request) to a port connected to a virtual machine associated with the requested functionality and the requested type of storage capacity, and if yes, add the storage device (if the storage node has the type of storage capacity requested by the user request) or the virtual device (if the

storage node does not have the type of storage capacity requested by the user request) to a capacity pool of storage media tiering or storage capacity virtualization, make another device corresponding to the capacity pool, and relate said another device to the port connected to the virtual machine associated with the requested functionality and the requested type of storage capacity.

[0008] In some embodiments, the memory is configured to store third information indicating access interface; and wherein the processor is configured to check whether the access interface is file or object by referring to the third information, and if the access interface is file or object, the processor is configured to: if the access interface is file, make a file system in the storage node; if the access interface is object, create an object store in the storage node; if the storage node has the type of storage capacity requested by the user request, relate the storage device to an internet protocol port connected to a virtual machine associated with the requested functionality and the requested type of storage capacity; and if the storage node does not have the type of storage capacity requested by the user request, relate the virtual device to an internet protocol port connected to a virtual machine associated with the requested functionality and the requested type of storage capacity.

[0009] In specific embodiments, the processor is configured to determine whether the requested functionality is to reduce data size, and if yes, the processor is configured to estimate a needed physical capacity in consideration of the requested functionality to reduce data size. The checking comprises checking whether the selected storage node has a type of storage capacity requested by the user request with the estimated physical capacity.

The providing comprises providing, if the storage node does not have the type of storage capacity requested by the user request with the estimated physical capacity, a virtual device which is mapped to another storage capacity provided by another storage node, said another storage capacity satisfying the type of storage capacity requested by the user request with the estimated physical capacity.

[0010] In some embodiments, the processor is configured: if the storage node has the type of storage capacity requested by the user request with the estimated physical capacity, to allocate the storage capacity from the storage node; and if the storage node does not have the type of storage capacity requested by the user request with the estimated physical capacity, to search said another storage node having said another storage capacity with the estimated physical capacity and being connected to the storage node which has the functionality, and to allocate said another storage capacity from said another storage node.

[0011] In specific embodiments, the user request specifies multiple functionalities, and wherein the processor is configured to determine whether all the requested functionalities can be provided by one of the plurality of storage nodes, and if no, then determine whether all the requested functionalities can be provided by two or more storage nodes, and if yes, then: decide a device virtualization order among the two or more storage nodes having all the requested functionalities so as to provide an upper storage node and one or more lower storage nodes including a lowest storage node; and provide an upper virtual device for the upper storage node which corresponds to one or more lower virtual devices for the one or more lower

storage nodes. If the two or more storage node having all the requested functionalities do not have the type of storage capacity requested by the user request, a lowest virtual device for the lowest storage node is mapped to said another storage capacity provided by said another storage node, said another storage capacity satisfying the type of storage capacity requested by the user request.

[0012] In some embodiments, two storage nodes have all the requested functionalities, including the upper storage node and a middle storage node which do not have the type of storage capacity requested by the user request. The upper virtual device for the upper storage node corresponds to a middle virtual device of the middle storage node. The middle virtual device for the middle storage node is mapped to said another storage capacity provided by said another storage node, said another storage capacity satisfying the type of storage capacity requested by the user request.

[0013] In specific embodiments, the management computer further comprises an interface from which the user requests the functionality and total amount of storage capacity without specifying one or more storage nodes of the plurality of storage node.

[0014] Another aspect of the invention is directed to a storage provisioning method for a plurality of storage nodes. The method comprises: storing first information indicating functionalities which can be executed by each of the plurality of storage nodes and second information showing types of storage capacity included in each of the plurality of storage nodes; selecting a storage node which has a functionality requested by the user request by referring to the first information; checking whether the selected

storage node has a type of storage capacity requested by the user request by referring to the second information; and providing, if the storage node does not have the type of storage capacity requested by the user request, a virtual device which is mapped to another storage capacity provided by another storage node, said another storage capacity satisfying the type of storage capacity requested by the user request.

[0015] These and other features and advantages of the present invention will become apparent to those of ordinary skill in the art in view of the following detailed description of the specific embodiments.

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] FIG. 1 illustrates an example of a hardware configuration of a computer system in which the method and apparatus of the invention may be applied according to the first embodiment.

[0017] FIG. 2 illustrates an example of a node.

[0018] FIG. 3 illustrates an example of an enterprise storage system.

[0019] FIG. 4 illustrates an example of a traditional storage capacity provisioning screen provided by the management server.

[0020] FIG. 5 illustrates an example of the storage capacity provisioning screen in the present invention.

[0021] FIG. 5A illustrates an example of storage provisioning when the required physical capacity and required functionalities are provided by different storages according to the first embodiment.

[0022] FIG. 6 illustrates an example of the memory of the management server.

- [0023]** FIG. 7 illustrates an example of the memory of the storage system.
- [0024]** FIG. 8 illustrates an example of a functionality management table.
- [0025]** FIG. 9 illustrates an example of a capacity management table.
- [0026]** FIG. 10 shows an example of a flow diagram illustrating the process of a provisioning program (in the management server) to provision storage capacity.
- [0027]** FIG. 11 shows an example of a flow diagram illustrating the process of a capacity allocation program (1) (in the management server) to allocate physical storage capacity.
- [0028]** FIG. 12 shows an example of a flow diagram illustrating the process of a first device configuration program (1) (in the upper storage) to configure the device and storage functionalities.
- [0029]** FIG. 13 shows an example of a flow diagram illustrating the process of a second device configuration program (2) (in the upper storage).
- [0030]** FIG. 14 shows an example of a flow diagram illustrating the process of a replica configuration program executed in the management server.
- [0031]** FIG. 15 shows an example of a flow diagram illustrating the process of a snapshot configuration program executed in the management server.
- [0032]** FIG. 16 shows an example of a flow diagram illustrating the process of a remote copy configuration program executed in the management server.

[0033] FIG. 17 shows an example of a flow diagram illustrating the process of another capacity allocation program (2) executed in the management server.

[0034] FIG. 18 shows an example of a flow diagram illustrating the process of a third device configuration program (3).

[0035] FIG. 19 illustrates an example of storage provisioning when the required physical capacity and required functionalities are provided by different storages and the required functionalities are provided by two or more storages according to the second embodiment.

[0036] FIG. 20 shows an example of a flow diagram illustrating the process of a provisioning program (in the management server) to provision storage capacity according to the second embodiment.

[0037] FIG. 21 illustrates an example of a functionality table.

DETAILED DESCRIPTION OF THE INVENTION

[0038] In the following detailed description of the invention, reference is made to the accompanying drawings which form a part of the disclosure, and in which are shown by way of illustration, and not of limitation, exemplary embodiments by which the invention may be practiced. In the drawings, like numerals describe substantially similar components throughout the several views. Further, it should be noted that while the detailed description provides various exemplary embodiments, as described below and as illustrated in the drawings, the present invention is not limited to the embodiments described and illustrated herein, but can extend to other embodiments, as would be known or as would become known to those skilled in the art. Reference in the

specification to “one embodiment,” “this embodiment,” or “these embodiments” means that a particular feature, structure, or characteristic described in connection with the embodiment is included in at least one embodiment of the invention, and the appearances of these phrases in various places in the specification are not necessarily all referring to the same embodiment. Additionally, in the following detailed description, numerous specific details are set forth in order to provide a thorough understanding of the present invention. However, it will be apparent to one of ordinary skill in the art that these specific details may not all be needed to practice the present invention. In other circumstances, well-known structures, materials, circuits, processes and interfaces have not been described in detail, and/or may be illustrated in block diagram form, so as to not unnecessarily obscure the present invention.

[0039] Furthermore, some portions of the detailed description that follow are presented in terms of algorithms and symbolic representations of operations within a computer. These algorithmic descriptions and symbolic representations are the means used by those skilled in the data processing arts to most effectively convey the essence of their innovations to others skilled in the art. An algorithm is a series of defined steps leading to a desired end state or result. In the present invention, the steps carried out require physical manipulations of tangible quantities for achieving a tangible result. Usually, though not necessarily, these quantities take the form of electrical or magnetic signals or instructions capable of being stored, transferred, combined, compared, and otherwise manipulated. It has proven convenient at times, principally for reasons of common usage, to refer to these signals as

bits, values, elements, symbols, characters, terms, numbers, instructions, or the like. It should be borne in mind, however, that all of these and similar terms are to be associated with the appropriate physical quantities and are merely convenient labels applied to these quantities. Unless specifically stated otherwise, as apparent from the following discussion, it is appreciated that throughout the description, discussions utilizing terms such as “processing,” “computing,” “calculating,” “determining,” “displaying,” or the like, can include the actions and processes of a computer system or other information processing device that manipulates and transforms data represented as physical (electronic) quantities within the computer system’s registers and memories into other data similarly represented as physical quantities within the computer system’s memories or registers or other information storage, transmission or display devices.

[0040] The present invention also relates to an apparatus for performing the operations herein. This apparatus may be specially constructed for the required purposes, or it may include one or more general-purpose computers selectively activated or reconfigured by one or more computer programs. Such computer programs may be stored in a computer-readable storage medium including non-transitory medium, such as, but not limited to optical disks, magnetic disks, read-only memories, random access memories, solid state devices and drives, or any other types of media suitable for storing electronic information. The algorithms and displays presented herein are not inherently related to any particular computer or other apparatus. Various general-purpose systems may be used with programs and modules in accordance with the teachings herein, or it may prove

convenient to construct a more specialized apparatus to perform desired method steps. In addition, the present invention is not described with reference to any particular programming language. It will be appreciated that a variety of programming languages may be used to implement the teachings of the invention as described herein. The instructions of the programming language(s) may be executed by one or more processing devices, e.g., central processing units (CPUs), processors, or controllers.

[0041] Exemplary embodiments of the invention, as will be described in greater detail below, provide apparatuses, methods and computer programs for storage capacity provisioning in a heterogeneous storage environment.

[0042] Some embodiments will be described with reference to drawings in the following. Any embodiments that will be described in the following do not restrict the invention in accordance with the claims, and all of elements and all of pairs of the elements that will be described in the embodiments are not necessarily essential for the means for solving the problems of the invention. In the following descriptions, the processing will be described while a "program" is handled as a subject in some cases. In the case in which the program is executed by a processor, the predetermined processing is executed. Consequently, a subject of a processing can also be a processor. The processing that is disclosed while a program is handled as a subject can also be a processing that is executed by a processor that executes the program or an apparatus that is provided with the processor (for instance, a control device, a controller, and a storage system). Moreover, a part or a whole of a processing that is executed when the processor executes a

program can also be executed by a hardware circuit as substitute for or in addition to a processor.

[0043] First Embodiment

[0044] FIG. 1 illustrates an example of a hardware configuration of a computer system in which the method and apparatus of the invention may be applied according to the first embodiment. The computer system includes management node 100, compute node 200, storage node (commodity) 300, storage node (all flash) 400, and storage node (rich func) 500, and network 700 physically. Virtual machine 600 is created by using these nodes. CPU resource of each compute node constitutes computing resource pool and capacity of each node constitutes storage capacity pool. This is one example; other types of nodes can be added to the computer system.

[0045] FIG. 2 illustrates an example of a node. The node (100, 200, or 300) has program, memory, CPU, HDD (hard disk drive), and network interface (IF). This is a physical configuration of the management server, compute node, and storage node (commodity). The amount of the resource depends on the node type. The compute node may have a large number of the processors and the storage node may have a large amount of the HDD capacity. Some management program(s) may be installed in the management server. The HDD is an example of storage capacity. SSD (Solid State Drive) and other physical drives can be installed as storage capacity.

[0046] FIG. 3 illustrates an example of an enterprise storage system. The storage system 400 or 500 includes cache unit 201, storage I/F 202, processor 203, disk I/F 204, volume 205, disks 206, storage control

information 207, and storage program 208. The storage I/F 202 is coupled to the other nodes via the network 700, and mediates communication with the other nodes. The processor 203 executes a wide variety of processing by executing a wide variety of programs that have been stored into the storage program 208. Moreover, the processor 203 executes a wide variety of processing by using a wide variety of information that has been stored into the storage control information 207. The disk I/F 204 is coupled to at least one disk 206 via a bus. The volume 205 that is configured to manage data is configured by at least one storage region of the disk(s) 206. One example of a disk 206 is HDD. The disk 206 is not restricted to an HDD and can also be an SSD (Solid State Drive) or a DVD, for instance. Moreover, at least one disk can be collected up in a unit of a parity group, and a high reliability technique such as a RAID (Redundant Arrays of Independent Disks) can also be used. The volume 205 is provided as disk 206 to an operating system using the volume. The storage control information 207 stores a wide variety of information used by a wide variety of programs. The storage program 208 stores a wide variety of programs, such as read processing program or write processing program and so on. The cache unit 201 caches the data stored in the disk 206 for performance boost.

[0047] FIG. 4 illustrates an example of a traditional storage capacity provisioning screen provided by the management server 100. In this example, the storage node (commodity) 300 has 300TB, and storage system A has 10 TB and storage system B has 90TB of free storage capacity. Storage system A is product name. The user is making provisioning 50TB from commodity and attaching it to VM1 in the example. From the example,

free storage capacity is managed by each storage system. The user has to specify which storage system provides storage capacity. To do so, the user has to know the features of each storage system. For example, storage system A is a high performance storage system which has flash as physical storage media. The free capacity of each storage system is managed individually, because the storage systems each have different features and different performances.

[0048] Another example of a traditional storage capacity provisioning screen for user interface can be considered. For example, features of each storage system are not managed in the management server. Only total free storage capacity is managed and displayed in the screen, without specifying storage node(s). It is 400TB in the example of FIG. 4. In this case, the high performance all flash storage system and storage functionalities rich storage system are treated as commodity server based storage. As such, features of these storage systems are not utilized.

[0049] FIG. 5 illustrates an example of the storage capacity provisioning screen in the present invention. In this example, the total free capacity is displayed instead of the free capacity of each storage system. Provisioning capacity and destination VM correspond to the total free capacity. The user is making provisioning 50 TB from 400TB free capacity and attaching it to VM1 in this example. In the required media cost radio button, the user chooses the storage media type. High means high cost and high performance storage media such as a flash storage (Tier 1). Mid means mid cost and mid performance storage media such as an SAS HDD (Tier 2). Low means low cost and low performance storage media such as a SATA

HDD (Tier 3). In the required functionalities check box, the user chooses the needed functionalities and access interface. The user is not conscious of which storage system has which feature and performance. To achieve this storage provisioning, when two or more storage features are required and there is no storage having all required features, the storage capacity of a first storage having one of the features is mapped to a second storage having another feature by using device virtualization technology. The capacity is provided from the second storage to the server. The provided storage can have all required features. Details of the processing are described in FIG. 10 to FIG.19.

[0050] FIG. 5A illustrates an example of storage provisioning when the required physical capacity and required functionalities are provided by different storages according to the first embodiment. In this example, the request for storage for a VM in a host is issued to the storage having the required functionalities (upper storage). The storage having the required physical capacity may be referred to as the external storage and the physical capacity is mapped to the upper storage having the required functionalities. Examples of the processes of capacity allocation (FIG. 11) device configuration (FIGS. 12 and 13) are discussed below.

[0051] FIG. 6 illustrates an example of the memory of the management server. The memory includes a control information unit and a program. Functionality management table (FIG. 8), capacity management table (FIG. 9), and node distance table are managed in the control information unit. The node distance table manages the distance between nodes. Provisioning program (FIG. 10), capacity allocation program (FIG. 11), replica configuration

program (FIG. 14), snapshot configuration program (FIG. 15), and remote copy configuration program (FIG. 16) are stored in the program unit. Details of the tables and programs are described later.

[0052] FIG. 7 illustrates an example of the memory of the storage system. The memory includes a control information unit, a program unit, and a cache unit. Replication pair table, remote copy pair table, snapshot table and so on are managed in the control information unit. The replication pair table manages copy source device ID and copy destination device ID, and so on. The remote copy pair table manages copy source device ID, copy source storage system ID, copy destination device ID, copy destination storage system ID, path information which is used for data transfer, and so on. The snapshot table manages the snapshot pool information which manages the device configuring the snapshot pool and the relationship between the device used by the server and the snapshot pool. Device configuration program (FIGS. 12 and 13), device discover program (FIGS. 12 and 13), pair creation program (FIG. 14), snapshot pool configuration program (FIG. 15), snapshot initialization program (FIG. 15), and remote copy pair creation program (FIG. 16) are stored in the program unit. Details of the tables and programs are described later.

[0053] FIG. 8 illustrates an example of a functionality management table. Node ID, data protection, performance boost, cost reduction, and access I/F are managed. In this example, storage functionalities are categorized into data protection, performance boost, cost reduction, and access I/F. In this example, the storage system specified node ID 4 has de-duplication functionality and can be accessed with block I/F from the server.

The storage system specified node ID 5 has replication, remote copy, snapshot, and cache unit, and can be accessed with block, file or object I/F from the server. By the functionality management table, the management server knows which server has which functionalities.

[0054] FIG. 9 illustrates an example of a capacity management table. Node ID, media, free capacity, and used capacity are managed in the table. Media manages the disk type installed in the node specified by the node ID. By using media information, the management server can know the storage cost. The storage cost is dependent not only on storage media type but system. Therefore, the storage cost can be estimated in consideration of the type of system.

[0055] FIG. 10 shows an example of a flow diagram illustrating the process of a provisioning program (in the management server) to provision storage capacity according to the first embodiment. This program searches storage which has the required storage capacity and searches storage which has the required functionalities. The result can be two or more storages. If the storage having the required capacity differs from the storage having the required functionalities, the capacity is mapped to the storage having the required functionalities. Finally, it is attached to the VM as capacity of storage having functionalities.

[0056] In step S100, the provisioning program receives a capacity provisioning request from the user. The program searches storage which has all the requested functionalities (S101). If there is no storage having all the requested functionalities (S102), the program proceeds to step S103, reports error at S103, and terminates the processing (S109). If there is storage

having all requested functionalities, the program calls the capacity allocation program at the step S104 (FIG. 11). In step S105, if the result of the capacity allocation program is error, the program proceeds to step S103. If the result of the capacity allocation program is not error, the program issues the device configuration request to the storage (S106). If the required physical capacity and required functionalities are provided by different storages, the request is issued to the storage having the required functionalities. The storage which receives the request executes the device configuration program (FIG. 12 or 13 or 20). The details of the program are described later. In step S107, the program calls a program to configure storage functionalities. The appropriate program to call depends on the requested storage functionalities (FIG. 14 for replication, FIG. 15 for snapshot, and FIG. 16 for remote copy). After that, the program attaches the capacity to the specified VM (S108) and terminates the processing (S109).

[0057] FIG. 11 shows an example of a flow diagram illustrating the process of a capacity allocation program (1) (in the management server) to allocate physical storage capacity. In step S200, the program checks whether the storage having the required functionalities also has the requested cost of storage media or not. If the result is yes, the program allocates physical capacity from the same storage at step S201. If the result is no, the program searches the storage having a requested cost of storage media from the storage systems connected to the storage having the required functionalities (S202). Since the physical capacity is mapped to the storage having the required functionalities, the search object is limited to the connected storages. If there is no storage fulfilling the condition in step S203, the program reports

the error at the step S204. If there is storage fulfilling the condition in step S203, the program allocates the physical capacity from the storage in step S205. The program terminates the processing in step S206 after S201, S204, or S205.

[0058] FIG. 12 shows an example of a flow diagram illustrating the process of a first device configuration program (1) (in the upper storage) to configure the device and storage functionalities. The device configuration program (1) is called after receiving the device configuration request from the step S106 of provisioning program in FIG. 10. In step S300, the program checks whether the physical storage capacity is provided by some other storage or not. If the result is yes, the program obtains device information from the external storage which has the physical capacity (S301). The storage having the required capacity receives the request for device information from the upper storage which has functionalities and reports the device information to the upper storage by executing a device discover program (S302). Then the device configuration program (1) defines a virtual device corresponding to the physical capacity (S303). If the result of step S300 is no, the device configuration program (1) skips step S301 to step S303. In step S304, the program relates the device to the port connected to the VM. If the physical capacity is in external storage, the virtual device defined in step S303 is related to the port. The program terminates the processing in step S305. If the required functionality is cache, the cache of the upper storage can be used by the I/O processing.

[0059] FIG. 13 shows an example of a flow diagram illustrating the process of a second device configuration program (2) (in the upper storage).

When the requested functionalities are storage media tiering or storage capacity virtualization, the device configuration program (2) is executed instead of the device configuration program (1) of FIG. 12. Steps S300, S301, S302, and S303 are the same as the steps in FIG. 12. After step S300 or S303, step S400 is executed. The program adds allocated device to the capacity pool of storage media tiering or storage capacity virtualization. If the physical capacity is in external storage, the virtual device defined in step S303 is added. Then, the program makes another device corresponding to the capacity pool (S401). Next, the program relates the device made in previous step to the port connected to the VM (S402). Finally, the program terminates the processing (S403). Different from the example in FIG. 12, the program of FIG. 13 adds the device to the capacity pool instead of attaching it to the VM. As a result, some region of the device attached to the VM may correspond to another physical device belonging to the capacity pool. If the storage has the capacity pool including the storage media requested by the user when the user requests the storage provisioning, allocation of new physical capacity can be avoided. If the pool consisting of the requested cost of storage media does not exist, the pool is made and allocated device is added to the made pool. If the pool consisting of the requested cost of storage media exists and the free capacity more than the predetermined value is in the pool, allocation of physical capacity can be skipped. In particular, the steps S104, S105, S300, S301, S302 are skipped to avoid physical capacity allocation. If the physical capacity of the upper storage is contained in the pool, physical capacity allocation to the device defined in step S401 can be limited to the requested cost of storage media.

[0060] FIG. 14 shows an example of a flow diagram illustrating the process of a replica configuration program executed in the management server. When replication is requested, the replica configuration program is called in step S107 of the provisioning program in FIG. 10. In step S500, the program calls a capacity allocation program to allocate physical capacity for replicated data (FIG. 11). The program checks whether the device for replicated data is in the same storage which has the replication functionality (S501). If the result is yes, the program skips step S502 and proceeds to the step S503. If the result is no, the program specifies execution of the device configuration program to the storage which has the replication functionality in step S502. The storage executes device configuration program (1) or (2) (FIG. 12 or 13). Then, the replica configuration program issues a pair creation command to the storage which has the replication functionality (S503). The storage which receives the pair creation command executes the pair creation program. The pair creation program receives the command from the management server (S504) and updates the replication pair table (S505). After that, the pair creation program initializes the replication processes such as an initial copy and responds to the management server (S506). The replica configuration program receives the response and terminates the processing (S507).

[0061] FIG. 15 shows an example of a flow diagram illustrating the process of a snapshot configuration program executed in the management server. When snapshot is requested, the snapshot configuration program is called in step S107 of the provisioning program in FIG. 10. In step S600, the program queries whether there is a snapshot pool or not to the storage which

has the snapshot functionality. The storage which receives the query checks the existence of the snapshot pool and reports it to the management server (S601). The snapshot configuration program checks whether there is snapshot pool to store differentiated data of snapshot (S602). If the result is yes, the program skips steps S603, S604, and S605 and proceeds to step S606. If the result is no, the program calls the capacity allocation program to store the differentiated data in step S603 (FIG. 11). Then, the program checks whether the allocated capacity is in the same storage which has the snapshot functionality (S604). If the result is yes, the program skips step S605 and proceeds to step S606. If the result of the step S604 is no, the program specifies execution of the device configuration program to the storage (S605). The storage which receives the request executes the device configuration program (1) or (2) (FIG. 12 or 13). Then, the snapshot configuration program issues the snapshot initialization command to the storage having the snapshot functionality (S606). If the snapshot pool configuration is needed, a snapshot pool configuration request can be included in the command. The storage which receives the snapshot initialization command executes the snapshot initialization program. The snapshot initialization program checks whether the command includes the snapshot pool configuration request or not. If the result is no, the program skips step S608 and proceeds to step S609. If the result is yes, the program makes a snapshot pool and adds the specified device to the snapshot pool (S608). Then, the program relates the source device to the snapshot pool (S609) and responds to the management server (S610). Finally, the snapshot

configuration program receives the response from the storage and terminates the processing (S611).

[0062] FIG. 16 shows an example of a flow diagram illustrating the process of a remote copy configuration program executed in the management server. When remote copy is requested, the remote copy configuration program is called in step S107 of the provisioning program in FIG. 10. In step S700, the program searches the destination storage of remote copy. The destination storage of remote copy is one that has the remote copy functionality and is installed a long distance away from the source storage, and is connected to source storage selected by the provisioning program. The distance between storage systems can be checked by using the node distance table. Then, the program calls the capacity allocation program for the remote copy destination device in step S701. The capacity allocation program described in FIG. 11 is executed. The "same storage" in step S200 of the capacity allocation program in FIG. 11 means the destination storage. After capacity allocation, the program checks whether the destination device is in the destination storage (S702). If the result is yes, the program skips step S703 and proceeds to step S704. If the result is no, the program specifies execution of the device configuration program to the destination storage (S703). The storage which receives the request executes the device configuration program (1) or (2) (FIG. 12 or 13). Then, the program issues a remote copy pair creation command to the source storage (S704). The source storage which receives the remote copy pair creation command executes the remote copy pair creation program in the source storage. The remote copy pair creation program issues the pair creation request to the

destination storage (S706). The destination storage executes the remote copy pair creation program in the destination storage and updates the replication pair table (S707). Next, the remote copy pair creation program in source storage updates the replication pair table in the source storage (S708) and initializes the replication processes such as an initial copy and so on (S709). Finally, the remote copy configuration program receives the response from the remote copy pair creation program in the source storage and terminates the processing (S710). Some conditions for remote copy pair creation can be checked before steps S707 and S708.

[0063] FIG. 17 shows an example of a flow diagram illustrating the process of another capacity allocation program (2) executed in the management server. When the functionalities to reduce the data size such as compression or de-duplication are requested, the capacity allocation program (2) of FIG. 17 is executed instead of the capacity allocation program (1) of FIG. 11. When the data reduction functionalities is used, the needed physical capacity will be reduced. As a result, the capacity allocation program (2) allocates the physical capacity in consideration of the effect of the data reduction functionalities. In the capacity allocation program (2), step S800 is inserted just before step S200 of FIG. 11. In step S800, the capacity allocation program (2) estimates the needed physical capacity in consideration of the effect of compression or de-duplication. The effect of compression or de-duplication can be decided from the statistics in the storage or general known effect. The remaining process is identical to that of FIG. 11.

[0064] FIG. 18 shows an example of a flow diagram illustrating the process of a third device configuration program (3). When file or object is requested as access interface, the device configuration program (3) is executed (instead of (1) in FIG. 11 or (2) in FIG. 12). The process of FIG. 18 is similar to the process of FIG. 12. Step S900 is added and step S901 replaces step S303. In this program, the device configuration program (3) makes file system or configure object store in step S900 after step S302. In step S901, the program relates the capacity to the port connected to the VM. The capacity is related to the IP (internet protocol) port which is connected to the VM. Generally, the file system or object storage is accessed by using IP network.

[0065] Second Embodiment

[0066] FIG. 19 illustrates an example of storage provisioning when the required physical capacity and required functionalities are provided by different storages and the required functionalities are provided by two or more storages according to the second embodiment. In this example, the request for storage for a VM in a host is issued to the storage having a first required functionality of remote copy (upper storage). A middle storage provides a second required functionality of replication. The storage having the required physical capacity may be referred to as the external storage and the physical capacity is mapped to the middle storage having the replication functionality and to the upper storage having the remote copy functionality. FIG. 19 shows a device virtualization order between the upper storage and the lower storage. It is possible to have an upper storage and two or more lower storages needed to provide a plurality of functionalities. In general, the physical

capacity is mapped to the lowest storage of the multiple storages having the required functionalities. The volume is provided from the upper storage to the VM.

[0067] FIG. 20 shows an example of a flow diagram illustrating the process of a provisioning program (in the management server) to provision storage capacity according to the second embodiment. This program searches storage which has the required storage capacity and searches storages which have the required functionalities. The result can be two or more storages to provide the required functionalities. If the storage having the required capacity differs from the storage having the required functionalities, the capacity is mapped to the storage(s) having the required functionalities. Finally, it is attached to the VM as capacity of storage having functionalities.

[0068] FIG. 20 (second embodiment) is similar to FIG. 10 (first embodiment), but adds steps S1000 and S1001 and replaces S106 with steps S1002 to S1006. S100 to S105 and S107 to S109 remain the same. After step S102, if there is no storage having all the requested functionalities, the program proceeds to step S1000 and searches storages which can provide at least one of the requested functionalities. In step S1001, if all the requested functionalities are not provided by one or more storages, the program proceeds to step S103, reports error at S103, and terminates the processing (S109). If all the requested functionalities are provided by one or more storages, the program calls the capacity allocation program at the step S104 (FIG. 11). In step S105, if the result of the capacity allocation program is error, the program proceeds to step S103. If the result of the capacity allocation program is not error, the program determines whether all the

requested functionalities are provided by one storage. If yes, the program skips steps S1003 to S1006 and proceeds to step S107. If no, the program performs steps S1003 to S1006 before performing step S107.

[0069] In step S1003, the program decides the device virtualization order. The order of the storages having the requested functionalities is decided (e.g., upper storage and middle storage). For the example shown in FIG. 19, the order is upper storage, middle storage, and external storage. In step S1004, the program specifies execution of the device configuration program to the middle storage. The virtual volume corresponding to the physical capacity is defined. In step S1005, the program specifies execution of the device configuration program to the upper storage. This defines another virtual volume corresponding to the virtual volume defined in the middle storage in step S1004. In step S1006, the program updates the functionality table (FIG. 21). This table manages which storage system provides which functionality and is used for issuing command for functionality. In step S107, the program calls a program to configure storage functionalities. This step is executed for all required functionalities. In the example of FIG. 19, the remote copy configuration program for upper storage and the replica configuration program for middle storage are executed. After that, the program attaches the capacity to the specified VM (S108) and terminates the processing (S109).

[0070] FIG. 21 illustrates an example of a functionality table. The functionality table as columns of volume ID (1, 2, etc.), functionalities (remote copy, replication, etc.), and storage (node ID numbers). Volume ID is identification of volume to be provided to the VM. In this example, remote

copy and replication are applied to volume ID no. 1. Remote copy processing is executed in the storage system specified by node ID 1. Replication processing is executed in the storage system specified by node ID 2. When the operation command is issued, the management server can detect the storage system executing the functionality and issue the command.

Examples of operation command include suspend, re-sync, and the like.

[0071] Of course, the system configuration illustrated in FIG. 1 is purely exemplary of information systems in which the present invention may be implemented, and the invention is not limited to a particular hardware configuration. The computers and storage systems implementing the invention can also have known I/O devices (e.g., CD and DVD drives, floppy disk drives, hard drives, etc.) which can store and read the modules, programs and data structures used to implement the above-described invention. These modules, programs and data structures can be encoded on such computer-readable media. For example, the data structures of the invention can be stored on computer-readable media independently of one or more computer-readable media on which reside the programs used in the invention. The components of the system can be interconnected by any form or medium of digital data communication, e.g., a communication network. Examples of communication networks include local area networks, wide area networks, e.g., the Internet, wireless networks, storage area networks, and the like.

[0072] In the description, numerous details are set forth for purposes of explanation in order to provide a thorough understanding of the present invention. However, it will be apparent to one skilled in the art that not all of

these specific details are required in order to practice the present invention. It is also noted that the invention may be described as a process, which is usually depicted as a flowchart, a flow diagram, a structure diagram, or a block diagram. Although a flowchart may describe the operations as a sequential process, many of the operations can be performed in parallel or concurrently. In addition, the order of the operations may be re-arranged.

[0073] As is known in the art, the operations described above can be performed by hardware, software, or some combination of software and hardware. Various aspects of embodiments of the invention may be implemented using circuits and logic devices (hardware), while other aspects may be implemented using instructions stored on a machine-readable medium (software), which if executed by a processor, would cause the processor to perform a method to carry out embodiments of the invention. Furthermore, some embodiments of the invention may be performed solely in hardware, whereas other embodiments may be performed solely in software. Moreover, the various functions described can be performed in a single unit, or can be spread across a number of components in any number of ways. When performed by software, the methods may be executed by a processor, such as a general purpose computer, based on instructions stored on a computer-readable medium. If desired, the instructions can be stored on the medium in a compressed and/or encrypted format.

[0074] From the foregoing, it will be apparent that the invention provides methods, apparatuses and programs stored on computer readable media for storage capacity provisioning in a heterogeneous storage environment. Additionally, while specific embodiments have been illustrated

and described in this specification, those of ordinary skill in the art appreciate that any arrangement that is calculated to achieve the same purpose may be substituted for the specific embodiments disclosed. This disclosure is intended to cover any and all adaptations or variations of the present invention, and it is to be understood that the terms used in the following claims should not be construed to limit the invention to the specific embodiments disclosed in the specification. Rather, the scope of the invention is to be determined entirely by the following claims, which are to be construed in accordance with the established doctrines of claim interpretation, along with the full range of equivalents to which such claims are entitled.

WHAT IS CLAIMED IS:

1. A management computer coupled to a plurality of storage nodes, the management computer comprising:

a memory being configured to store first information indicating functionalities which can be executed by each of the plurality of storage nodes and second information showing types of storage capacity included in each of the plurality of storage nodes; and

a processor, in response to a user request, configured to:

select a storage node which has a functionality requested by the user request by referring to the first information;

check whether the selected storage node has a type of storage capacity requested by the user request by referring to the second information; and

provide, if the storage node does not have the type of storage capacity requested by the user request, a virtual device which is mapped to another storage capacity provided by another storage node, said another storage capacity satisfying the type of storage capacity requested by the user request.

2. The management computer according to claim 1, wherein the processor is configured:

if the storage node has the type of storage capacity requested by the user request, to allocate the storage capacity from the storage node and to

provide a storage device from the storage node having the allocated storage capacity; and

if the storage node does not have the type of storage capacity requested by the user request, to search said another storage node having said another storage capacity and being connected to the storage node which has the functionality, and to allocate said another storage capacity from said another storage node.

3. The management computer according to claim 2, wherein the processor is configured to:

check whether the requested functionality is storage media tiering or storage capacity virtualization, and if no, relate the storage device (if the storage node has the type of storage capacity requested by the user request) or the virtual device (if the storage node does not have the type of storage capacity requested by the user request) to a port connected to a virtual machine associated with the requested functionality and the requested type of storage capacity, and if yes, add the storage device (if the storage node has the type of storage capacity requested by the user request) or the virtual device (if the storage node does not have the type of storage capacity requested by the user request) to a capacity pool of storage media tiering or storage capacity virtualization, make another device corresponding to the capacity pool, and relate said another device to the port connected to the virtual machine associated with the requested functionality and the requested type of storage capacity.

4. The management computer according to claim 2, wherein the memory is configured to store third information indicating access interface; and wherein the processor is configured to check whether the access interface is file or object by referring to the third information, and if the access interface is file or object, the processor is configured to:

if the access interface is file, make a file system in the storage node;

if the access interface is object, create an object store in the storage node;

if the storage node has the type of storage capacity requested by the user request, relate the storage device to an internet protocol port connected to a virtual machine associated with the requested functionality and the requested type of storage capacity; and

if the storage node does not have the type of storage capacity requested by the user request, relate the virtual device to an internet protocol port connected to a virtual machine associated with the requested functionality and the requested type of storage capacity.

5. The management computer according to claim 1, wherein the processor is configured to determine whether the requested functionality is to reduce data size, and if yes, the processor is configured to estimate a needed physical capacity in consideration of the requested functionality to reduce data size;

wherein said checking comprises checking whether the selected storage node has a type of storage capacity requested by the user request with the estimated physical capacity; and

wherein said providing comprises providing, if the storage node does not have the type of storage capacity requested by the user request with the estimated physical capacity, a virtual device which is mapped to another storage capacity provided by another storage node, said another storage capacity satisfying the type of storage capacity requested by the user request with the estimated physical capacity.

6. The management computer according to claim 5, wherein the processor is configured:

if the storage node has the type of storage capacity requested by the user request with the estimated physical capacity, to allocate the storage capacity from the storage node; and

if the storage node does not have the type of storage capacity requested by the user request with the estimated physical capacity, to search said another storage node having said another storage capacity with the estimated physical capacity and being connected to the storage node which has the functionality, and to allocate said another storage capacity from said another storage node.

7. The management computer according to claim 1, wherein the user request specifies multiple functionalities, and wherein the processor is configured to determine whether all the requested functionalities can be provided by one of the plurality of storage nodes, and if no, then determine whether all the requested functionalities can be provided by two or more storage nodes, and if yes, then:

decide a device virtualization order among the two or more storage nodes having all the requested functionalities so as to provide an upper storage node and one or more lower storage nodes including a lowest storage node; and

provide an upper virtual device for the upper storage node which corresponds to one or more lower virtual devices for the one or more lower storage nodes;

wherein if the two or more storage node having all the requested functionalities do not have the type of storage capacity requested by the user request, a lowest virtual device for the lowest storage node is mapped to said another storage capacity provided by said another storage node, said another storage capacity satisfying the type of storage capacity requested by the user request.

8. The management computer according to claim 7,

wherein two storage nodes have all the requested functionalities, including the upper storage node and a middle storage node which do not have the type of storage capacity requested by the user request;

wherein the upper virtual device for the upper storage node corresponds to a middle virtual device of the middle storage node; and

wherein the middle virtual device for the middle storage node is mapped to said another storage capacity provided by said another storage node, said another storage capacity satisfying the type of storage capacity requested by the user request.

9. The management computer according to claim 1, further comprising:
an interface from which the user requests the functionality and total amount of storage capacity without specifying one or more storage nodes of the plurality of storage node.

10. A storage provisioning method for a plurality of storage nodes, the method comprising:

storing first information indicating functionalities which can be executed by each of the plurality of storage nodes and second information showing types of storage capacity included in each of the plurality of storage nodes;

selecting a storage node which has a functionality requested by the user request by referring to the first information;

checking whether the selected storage node has a type of storage capacity requested by the user request by referring to the second information;
and

providing, if the storage node does not have the type of storage capacity requested by the user request, a virtual device which is mapped to another storage capacity provided by another storage node, said another storage capacity satisfying the type of storage capacity requested by the user request.

11. The storage provisioning method according to claim 10, further comprising:

if the storage node has the type of storage capacity requested by the user request, allocating the storage capacity from the storage node and

providing a storage device from the storage node having the allocated storage capacity; and

if the storage node does not have the type of storage capacity requested by the user request, searching said another storage node having said another storage capacity and being connected to the storage node which has the functionality, and allocating said another storage capacity from said another storage node.

12. The storage provisioning method according to claim 11, further comprising:

checking whether the requested functionality is storage media tiering or storage capacity virtualization, and if no, relating the storage device (if the storage node has the type of storage capacity requested by the user request) or the virtual device (if the storage node does not have the type of storage capacity requested by the user request) to a port connected to a virtual machine associated with the requested functionality and the requested type of storage capacity, and if yes, adding the storage device (if the storage node has the type of storage capacity requested by the user request) or the virtual device (if the storage node does not have the type of storage capacity requested by the user request) to a capacity pool of storage media tiering or storage capacity virtualization, making another device corresponding to the capacity pool, and relating said another device to the port connected to the virtual machine associated with the requested functionality and the requested type of storage capacity.

13. The storage provisioning method according to claim 11, further comprising storing third information indicating access interface and checking whether the access interface is file or object by referring to the third information, and if the access interface is file or object, then:

if the access interface is file, making a file system in the storage node;

if the access interface is object, creating an object store in the storage node;

if the storage node has the type of storage capacity requested by the user request, relating the storage device to an internet protocol port connected to a virtual machine associated with the requested functionality and the requested type of storage capacity; and

if the storage node does not have the type of storage capacity requested by the user request, relating the virtual device to an internet protocol port connected to a virtual machine associated with the requested functionality and the requested type of storage capacity.

14. The storage provisioning method according to claim 10, further comprising:

determining whether the requested functionality is to reduce data size, and if yes, estimating a needed physical capacity in consideration of the requested functionality to reduce data size;

wherein said checking comprises checking whether the selected storage node has a type of storage capacity requested by the user request with the estimated physical capacity; and

wherein said providing comprises providing, if the storage node does not have the type of storage capacity requested by the user request with the estimated physical capacity, a virtual device which is mapped to another storage capacity provided by another storage node, said another storage capacity satisfying the type of storage capacity requested by the user request with the estimated physical capacity.

15. The storage provisioning method according to claim 14, further comprising:

if the storage node has the type of storage capacity requested by the user request with the estimated physical capacity, allocating the storage capacity from the storage node; and

if the storage node does not have the type of storage capacity requested by the user request with the estimated physical capacity, searching said another storage node having said another storage capacity with the estimated physical capacity and being connected to the storage node which has the functionality, and allocating said another storage capacity from said another storage node.

16. The storage provisioning method according to claim 10, wherein the user request specifies multiple functionalities, and the method further comprises determining whether all the requested functionalities can be provided by one of the plurality of storage nodes, and if no, then determining whether all the requested functionalities can be provided by two or more storage nodes, and if yes, then:

deciding a device virtualization order among the two or more storage nodes having all the requested functionalities so as to provide an upper storage node and one or more lower storage nodes including a lowest storage node;

providing an upper virtual device for the upper storage node which corresponds to one or more lower virtual devices for the one or more lower storage nodes; and

if the two or more storage node having all the requested functionalities do not have the type of storage capacity requested by the user request, mapping a lowest virtual device for the lowest storage node to said another storage capacity provided by said another storage node, said another storage capacity satisfying the type of storage capacity requested by the user request.

17. The storage provisioning method according to claim 16,

wherein two storage nodes have all the requested functionalities, including the upper storage node and a middle storage node which do not have the type of storage capacity requested by the user request;

wherein the upper virtual device for the upper storage node corresponds to a middle virtual device of the middle storage node; and

wherein the middle virtual device for the middle storage node is mapped to said another storage capacity provided by said another storage node, said another storage capacity satisfying the type of storage capacity requested by the user request.

18. The storage provisioning method according to claim 10, further comprising:

providing an interface from which the user requests the functionality and total amount of storage capacity without specifying one or more storage nodes of the plurality of storage node.

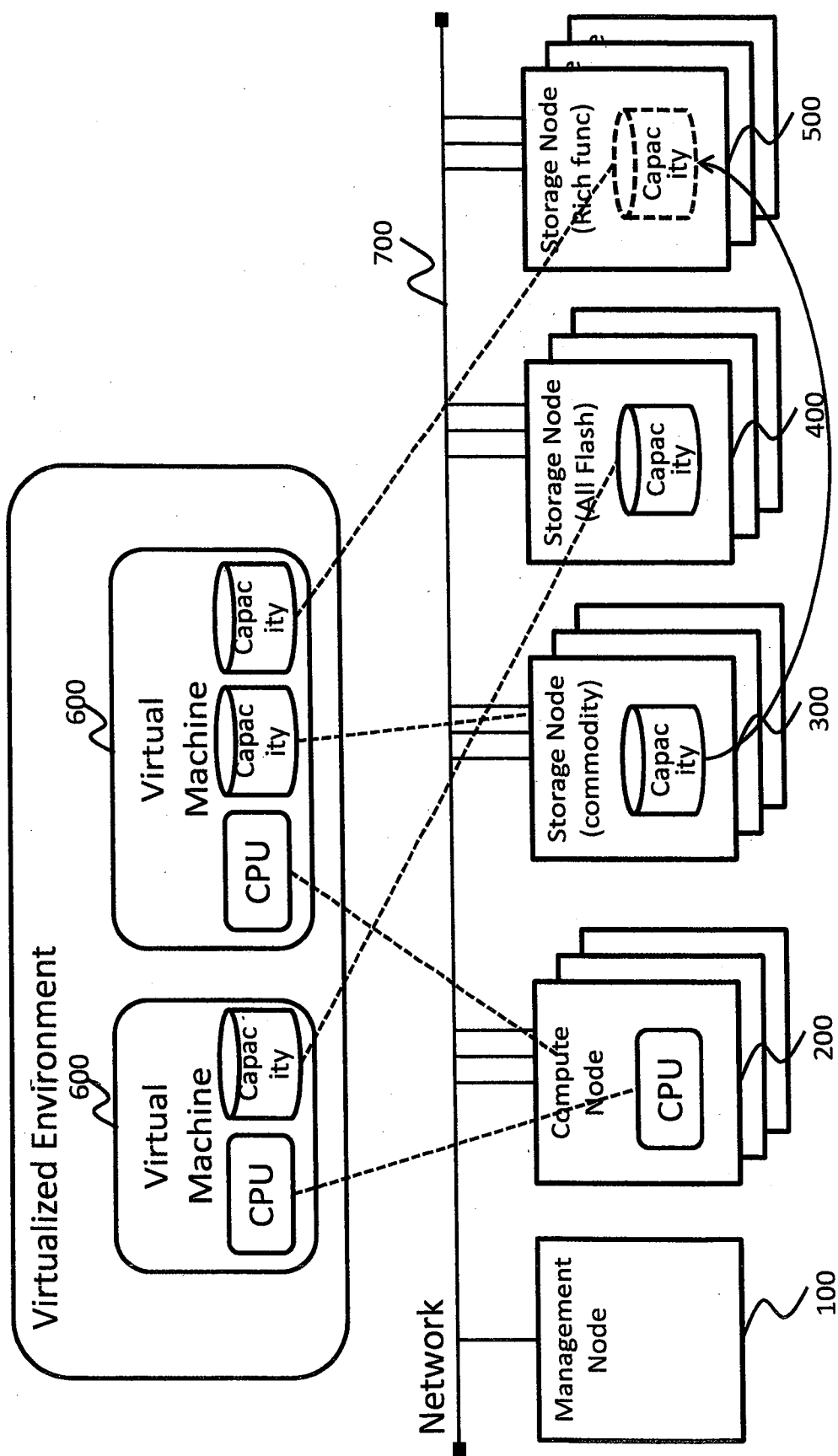


Fig.1

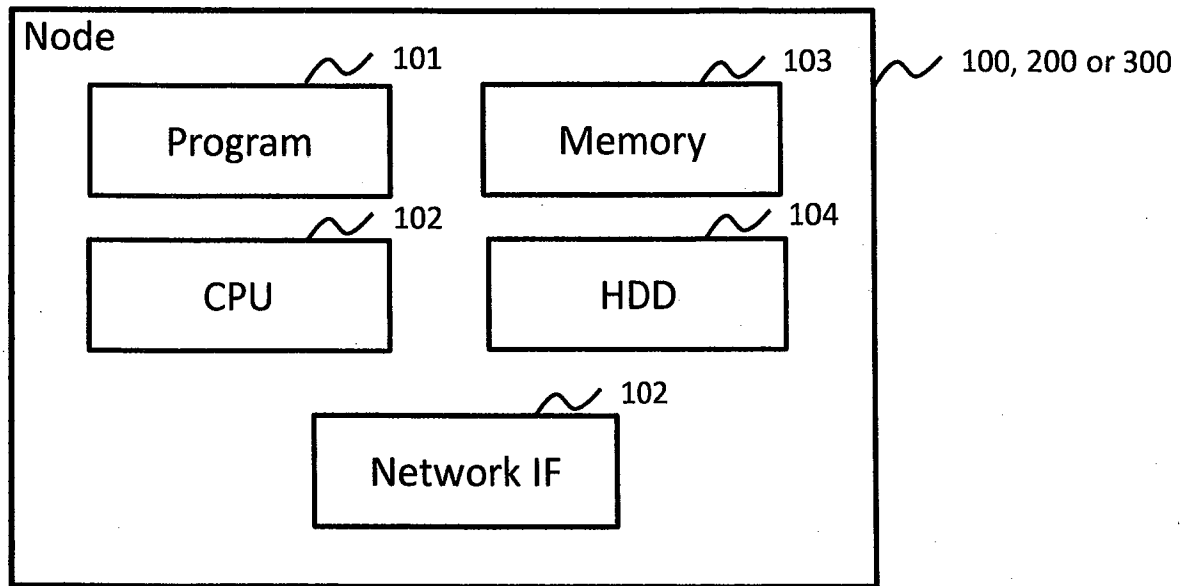


Fig.2

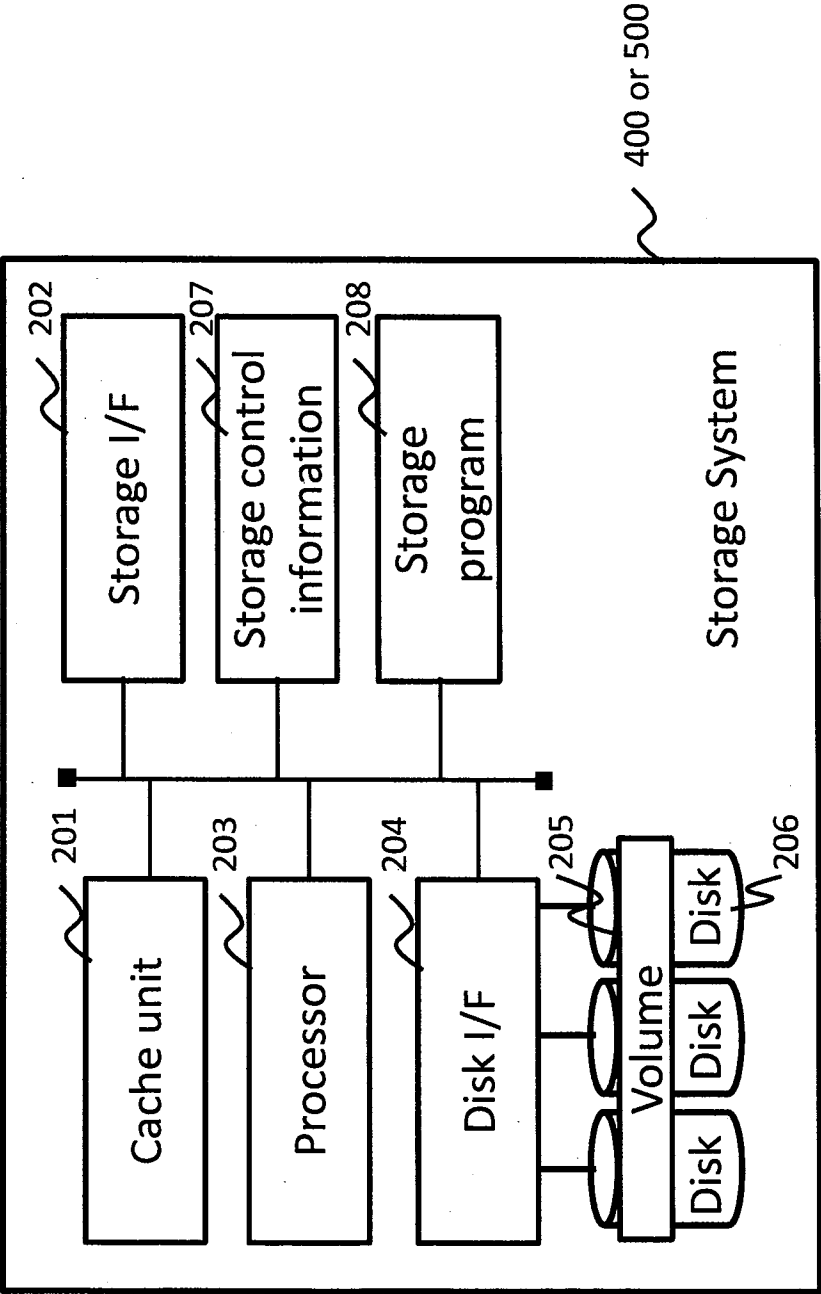


Fig.3

Capacity Provisioning

	Storage	Free Capacity	Provisioning Capacity	Destination VM
☒	<u>Commodity</u>	300TB	50TB	VM1
☐	<u>Storage System A</u>	10TB		
☐	<u>Storage System B</u>	90TB		

Provisioning

Fig.4

Capacity Provisioning		Destination VM	Required Media Cost	Required Functionalities	
Free Capacity	Provisioning Capacity			Data Protection	Performance Boost
400TB	50TB	VM1	☐ <u>High</u> (Tier1)	☐ <u>Replication</u>	☐ <u>Media Tiering</u>
			☐ <u>Mid</u> (Tier2)	☐ <u>Remote Copy</u>	☒ <u>Cache</u>
			☒ <u>Low</u> (Tier3)	☒ <u>Snapshot</u>	
				Cost Reduction	Access I/F
				☐ <u>De-duplication</u>	☒ <u>Block</u>
				☐ <u>Compression</u>	☐ <u>File</u>
					☐ <u>Object</u>
Provisioning					

Fig.5

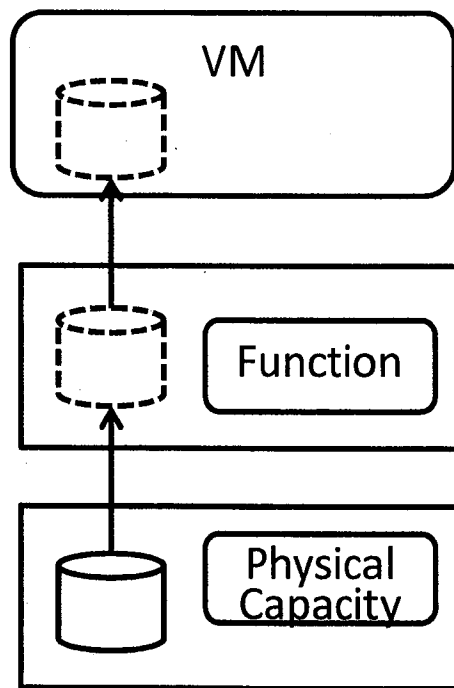


Fig.5A

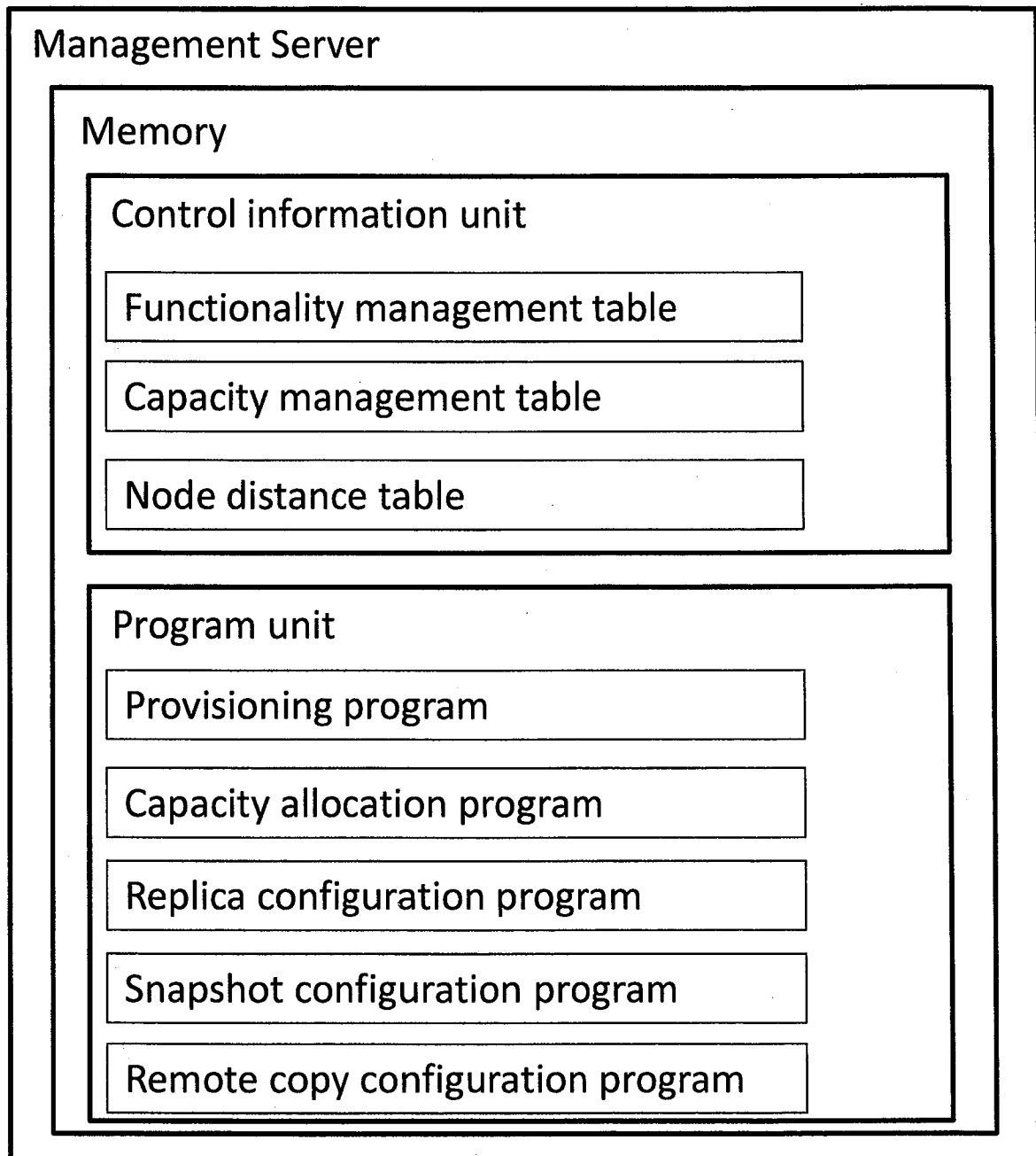


Fig. 6

8/22

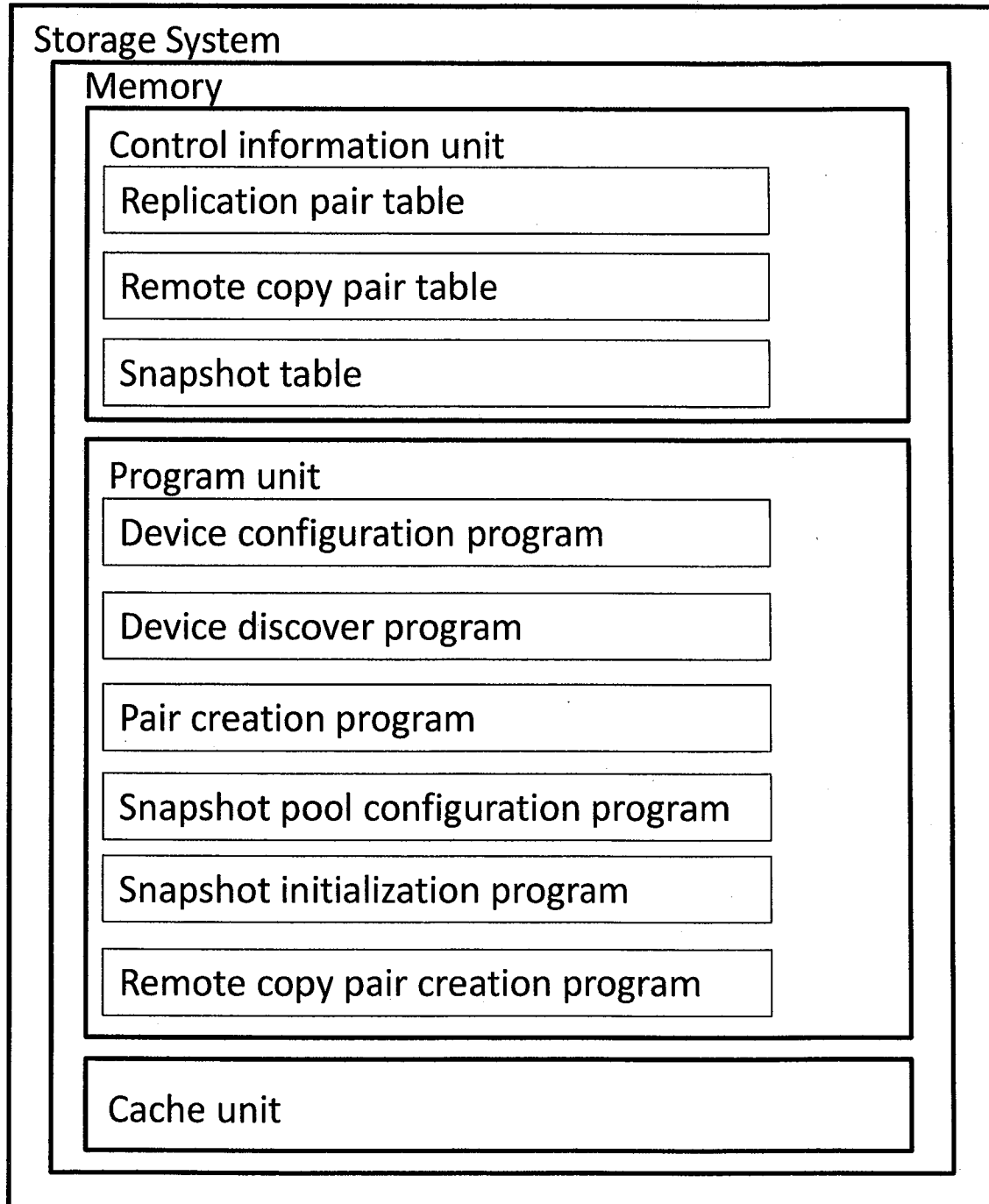


Fig. 7

9/22

Node ID	Data Protection	Performance Boost	Cost Reduction	Access I/F
1	—	—	—	Block
2	—	—	—	Block
3	—	—	—	Block
4	—	—	De-duplication	Block
5	Replication, Remote Copy, Snapshot	Cache	—	Block, File, Object

Fig.8

10/22

Node ID	Media	Free capacity	Used capacity
1	SATA	100TB	20TB
2	SATA	100TB	15TB
3	SATA	100TB	30TB
4	Flash	10TB	1TB
5	SAS	70TB	10TB
5	Flash	20TB	5TB
6	Tierd	50TB	5TB

Fig.9

11/22

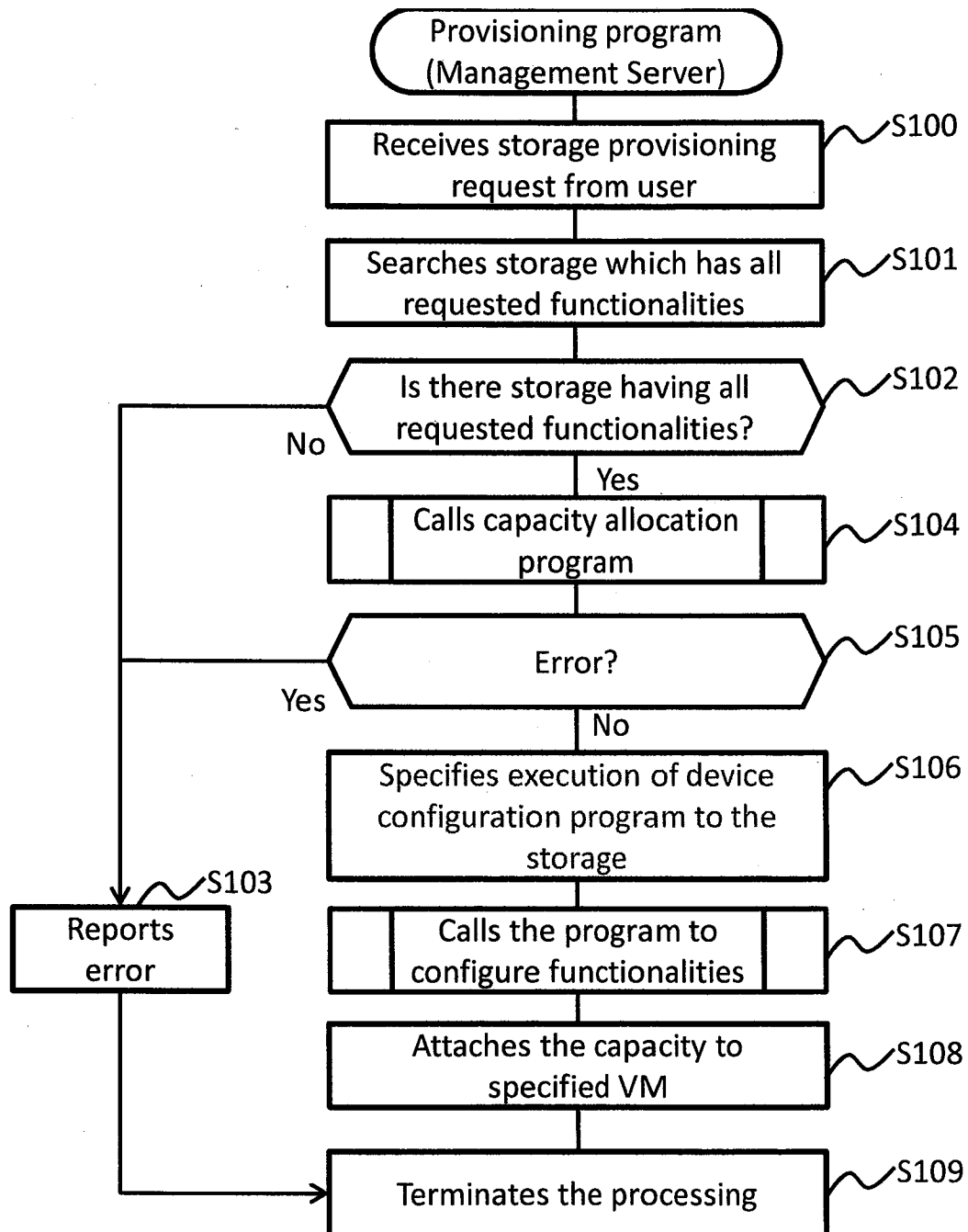


Fig. 10

12/22

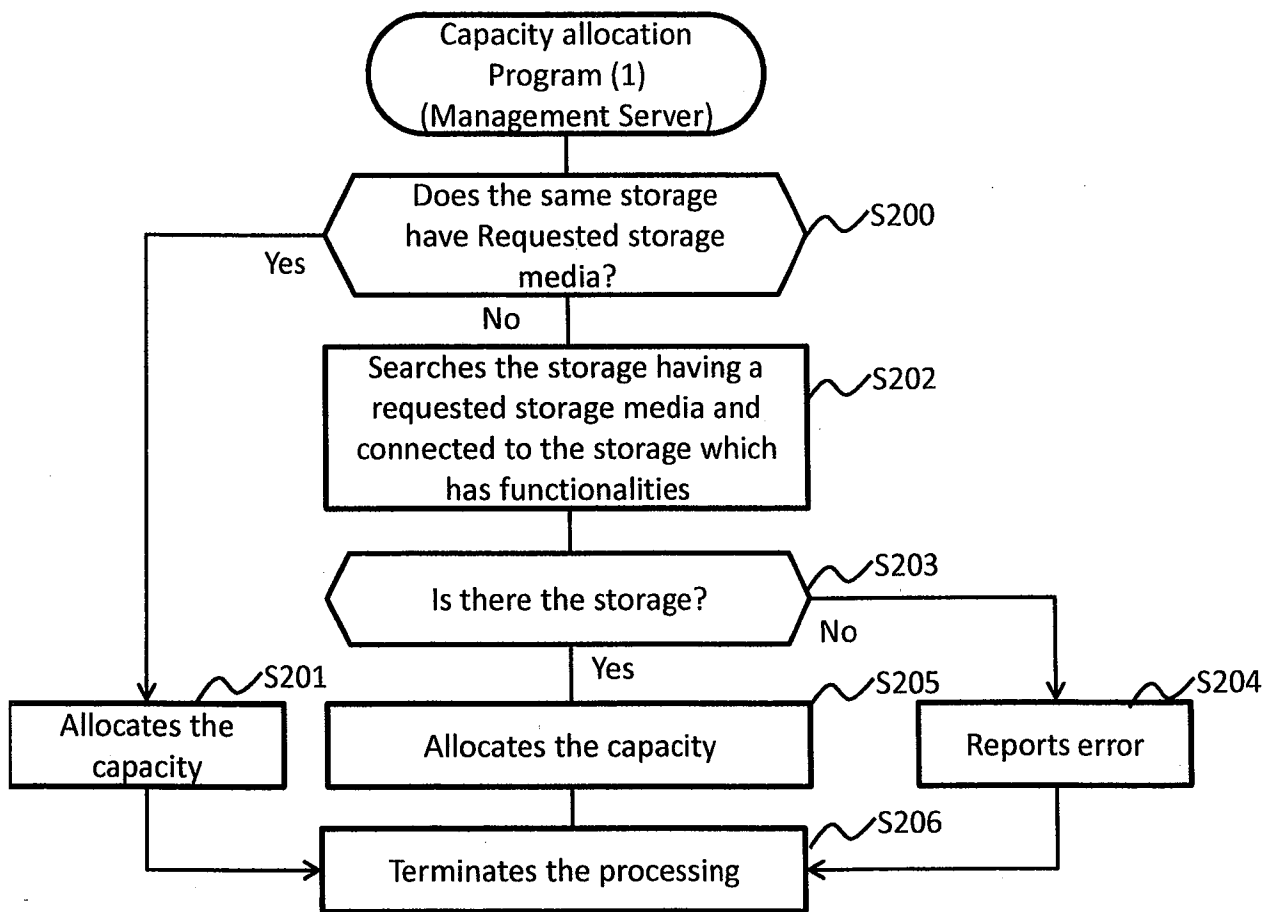


Fig.11

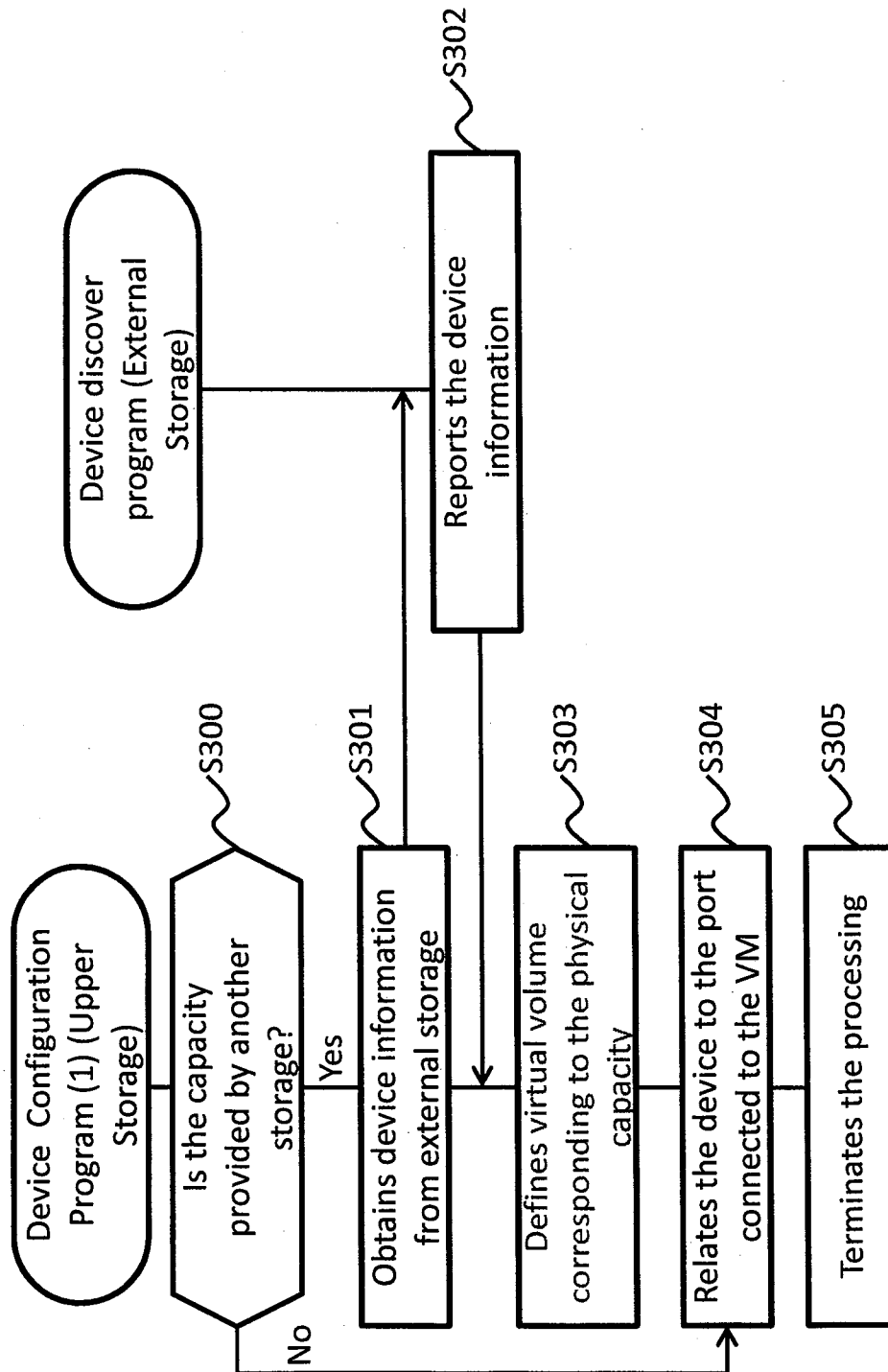


Fig.12

14/22

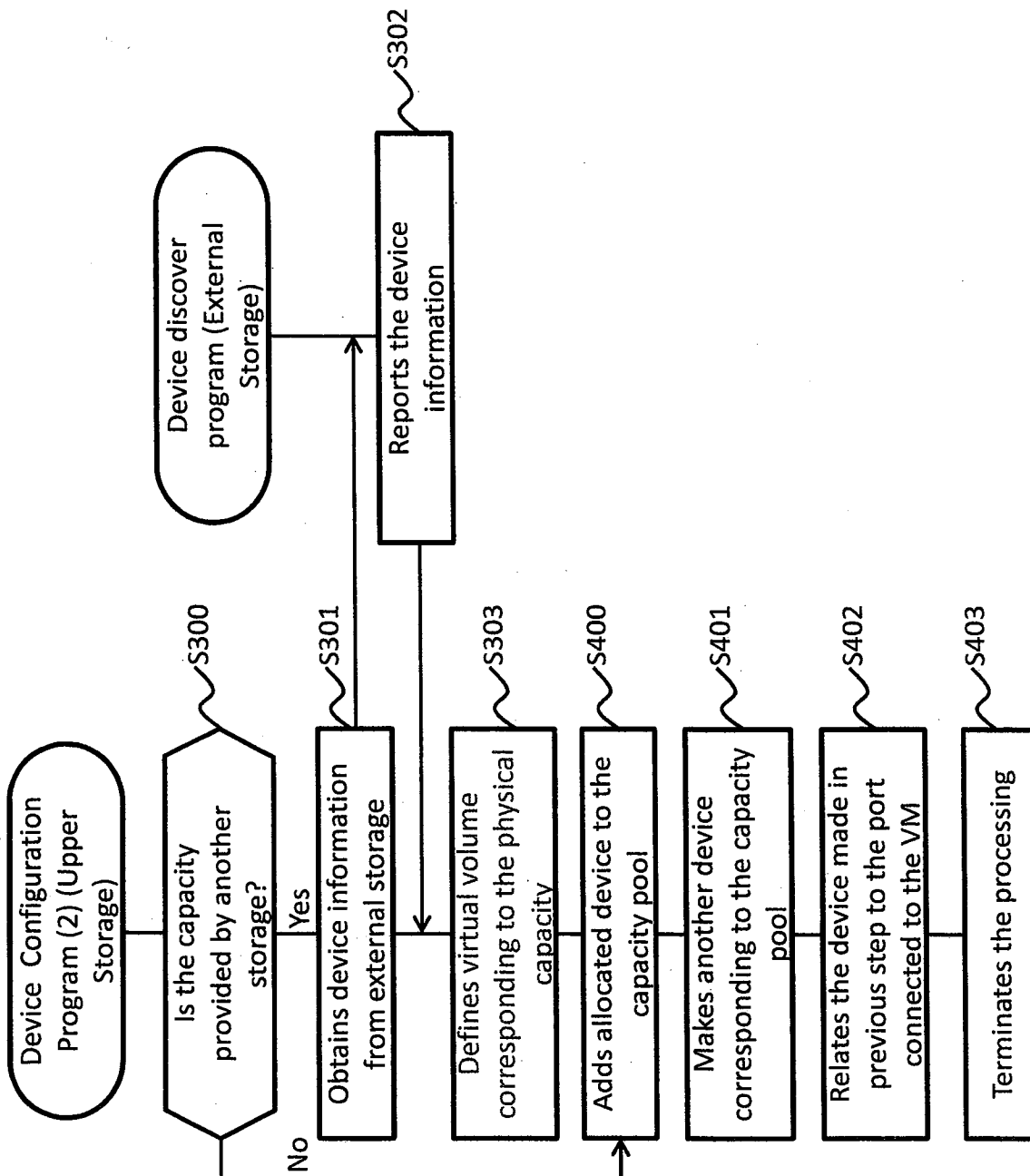


Fig. 13

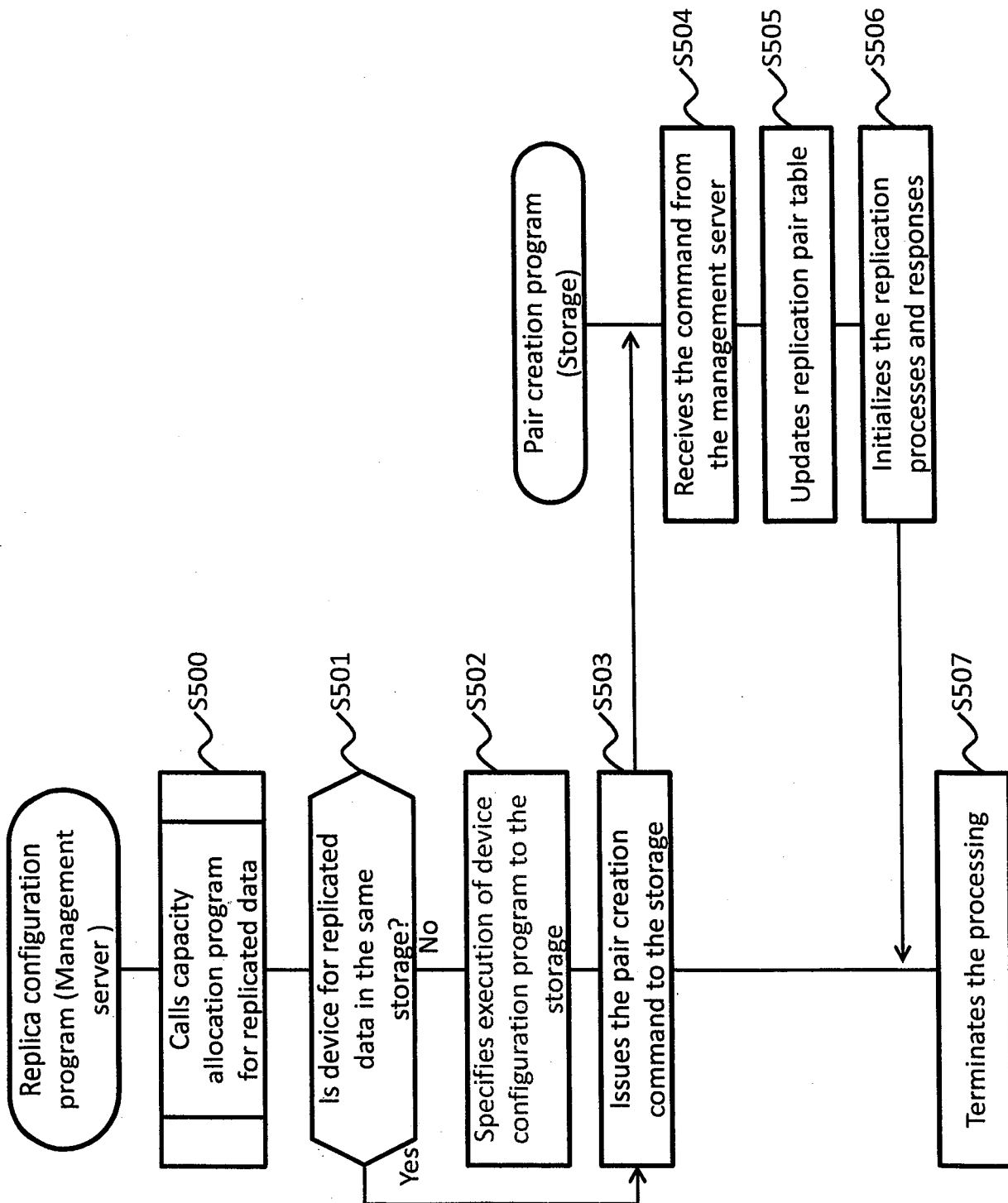


Fig. 14

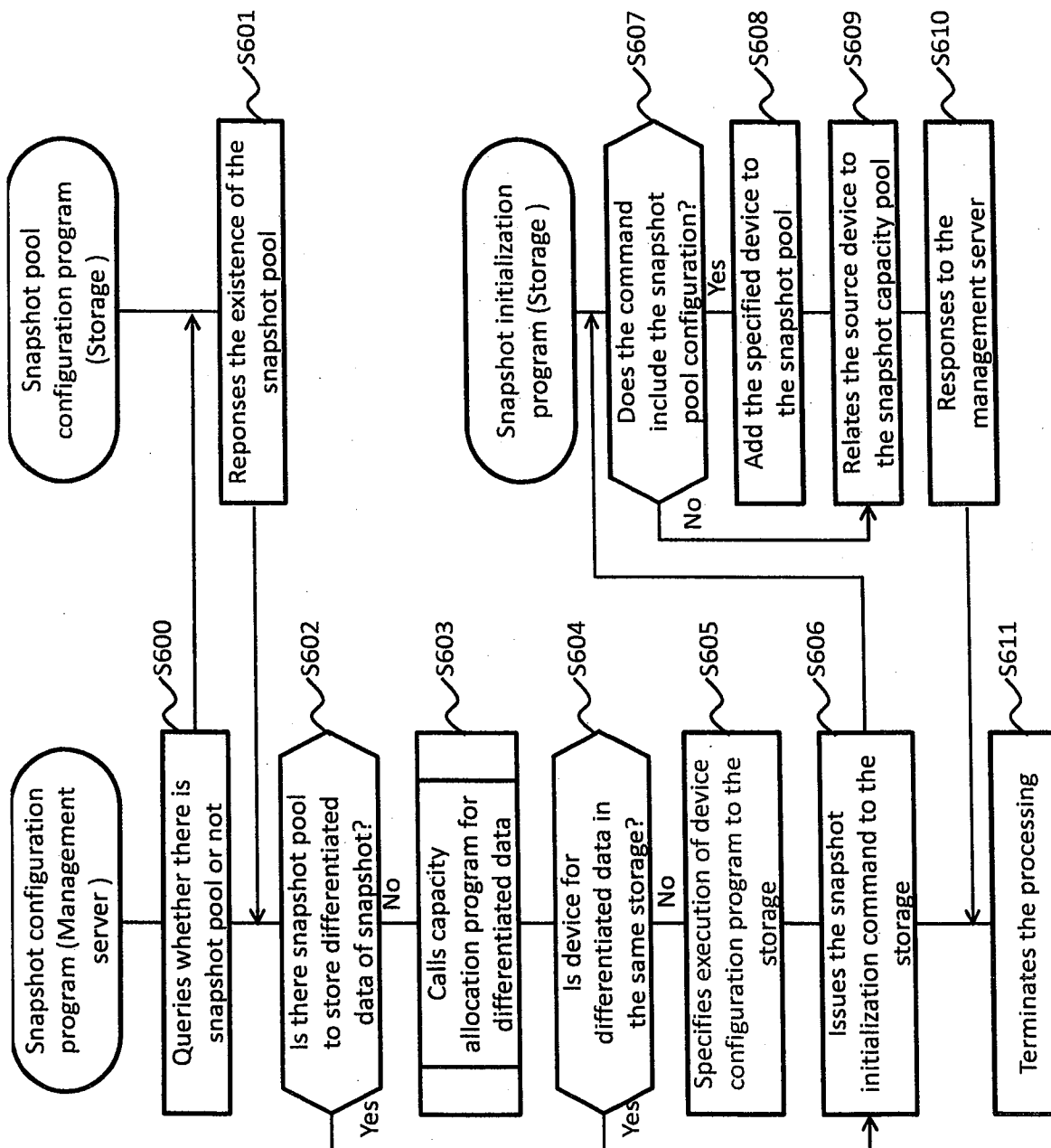


Fig. 15

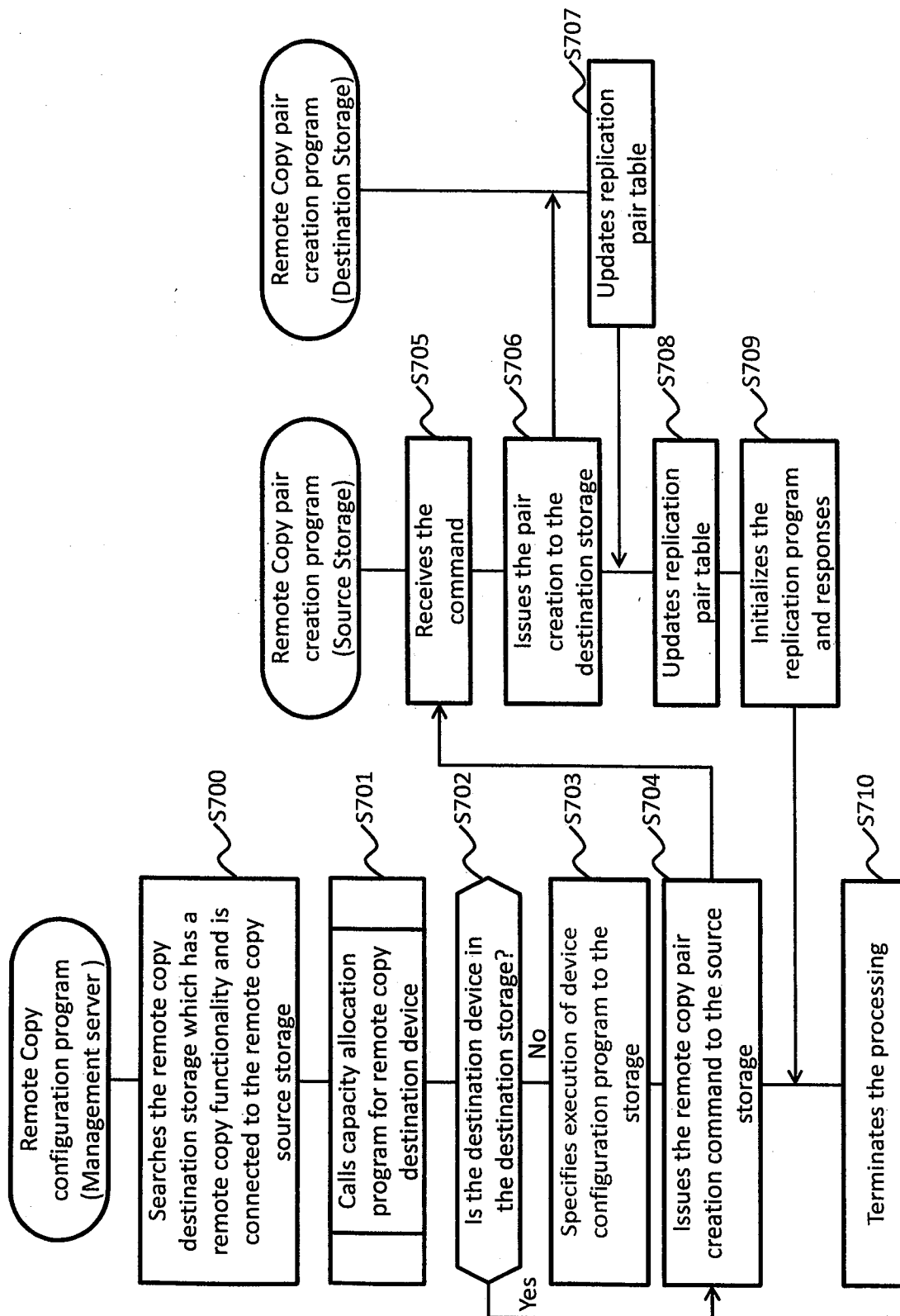


Fig. 16

18/22

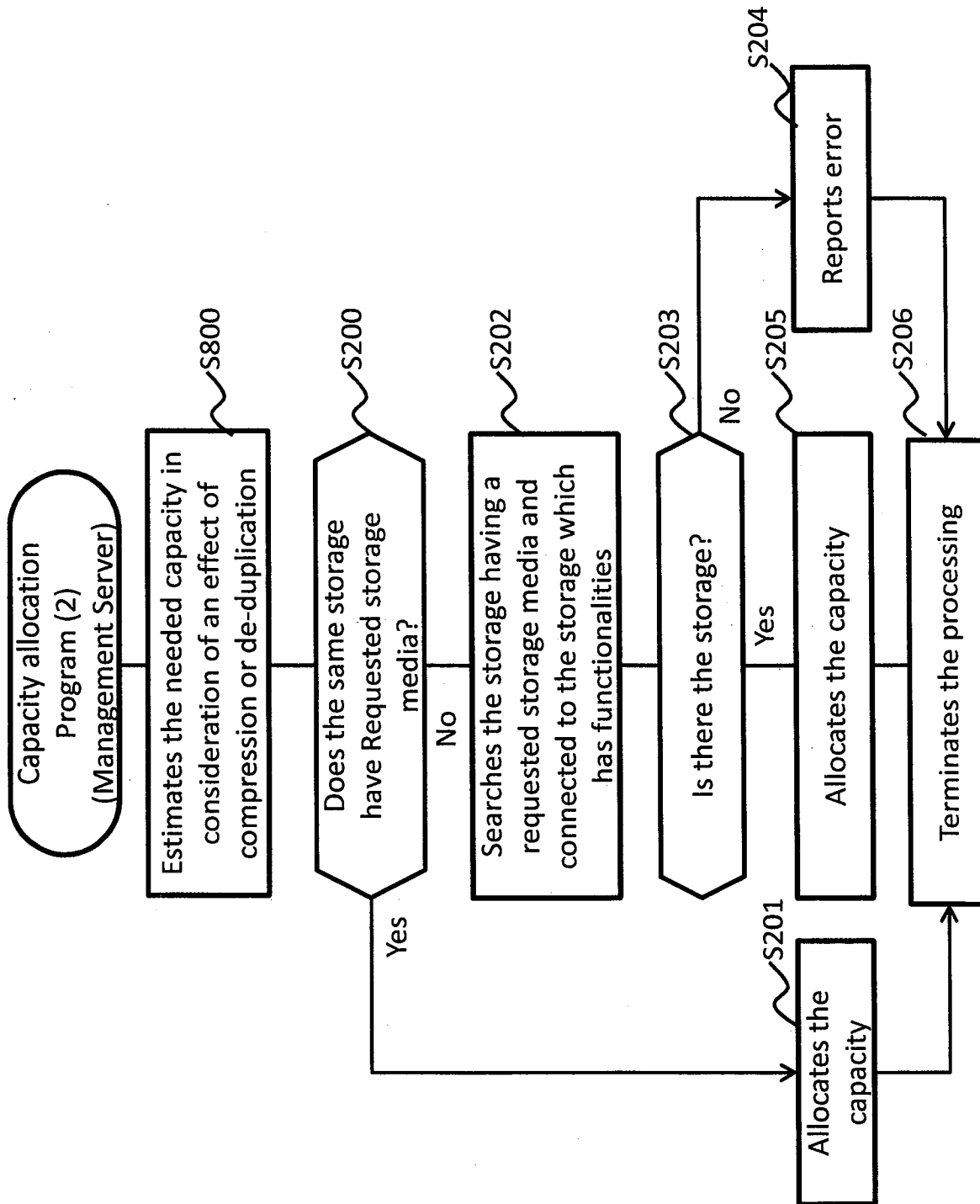


Fig.17

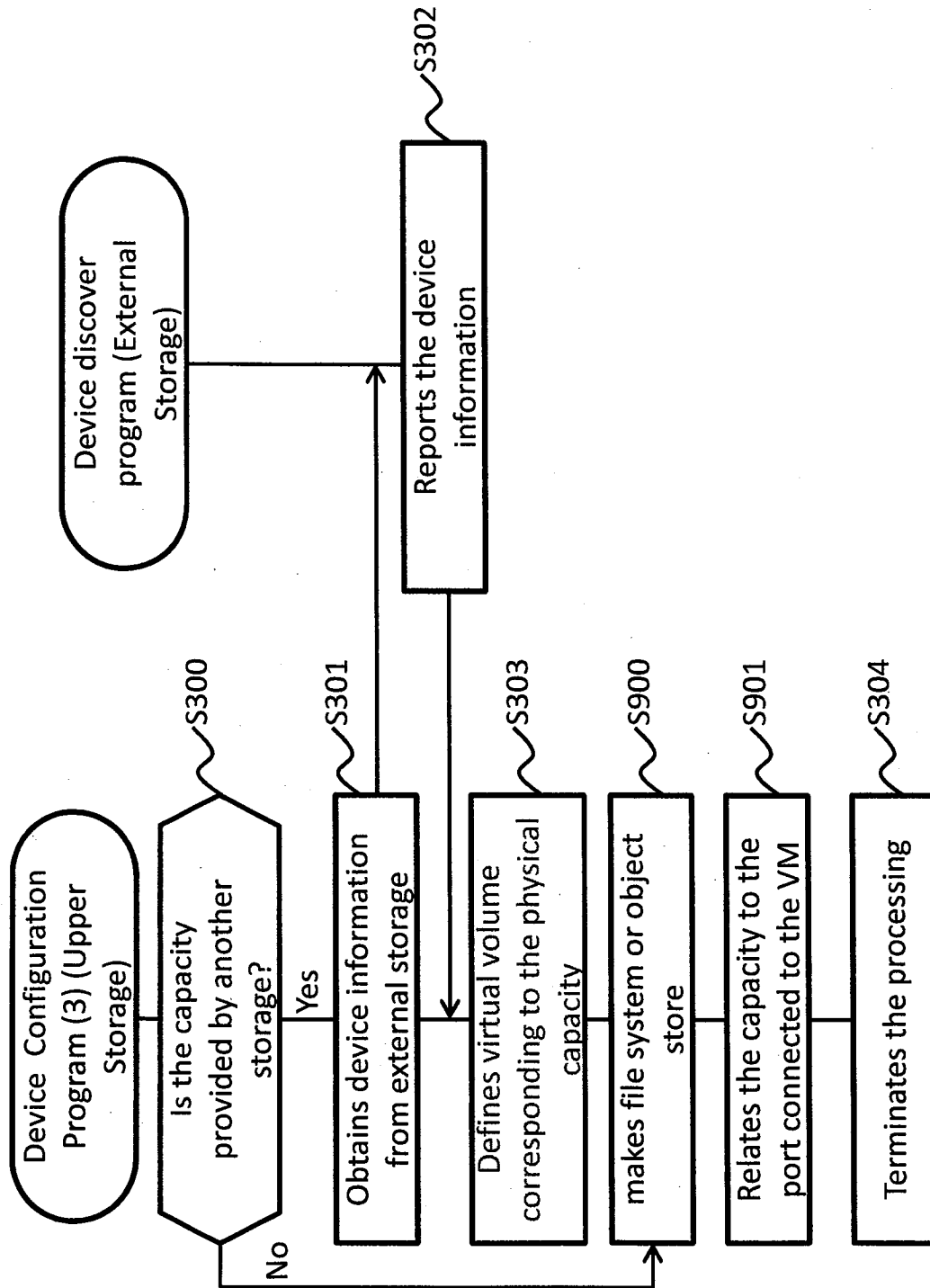


Fig.18

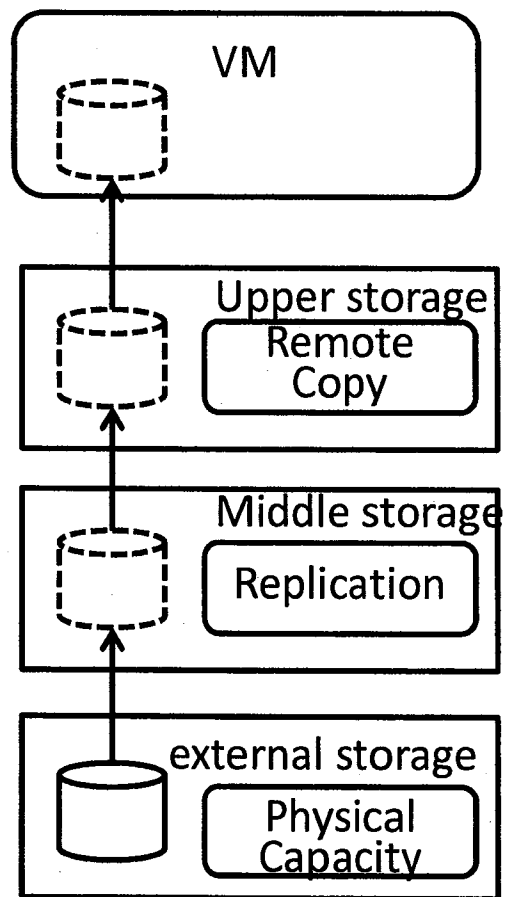


Fig.19

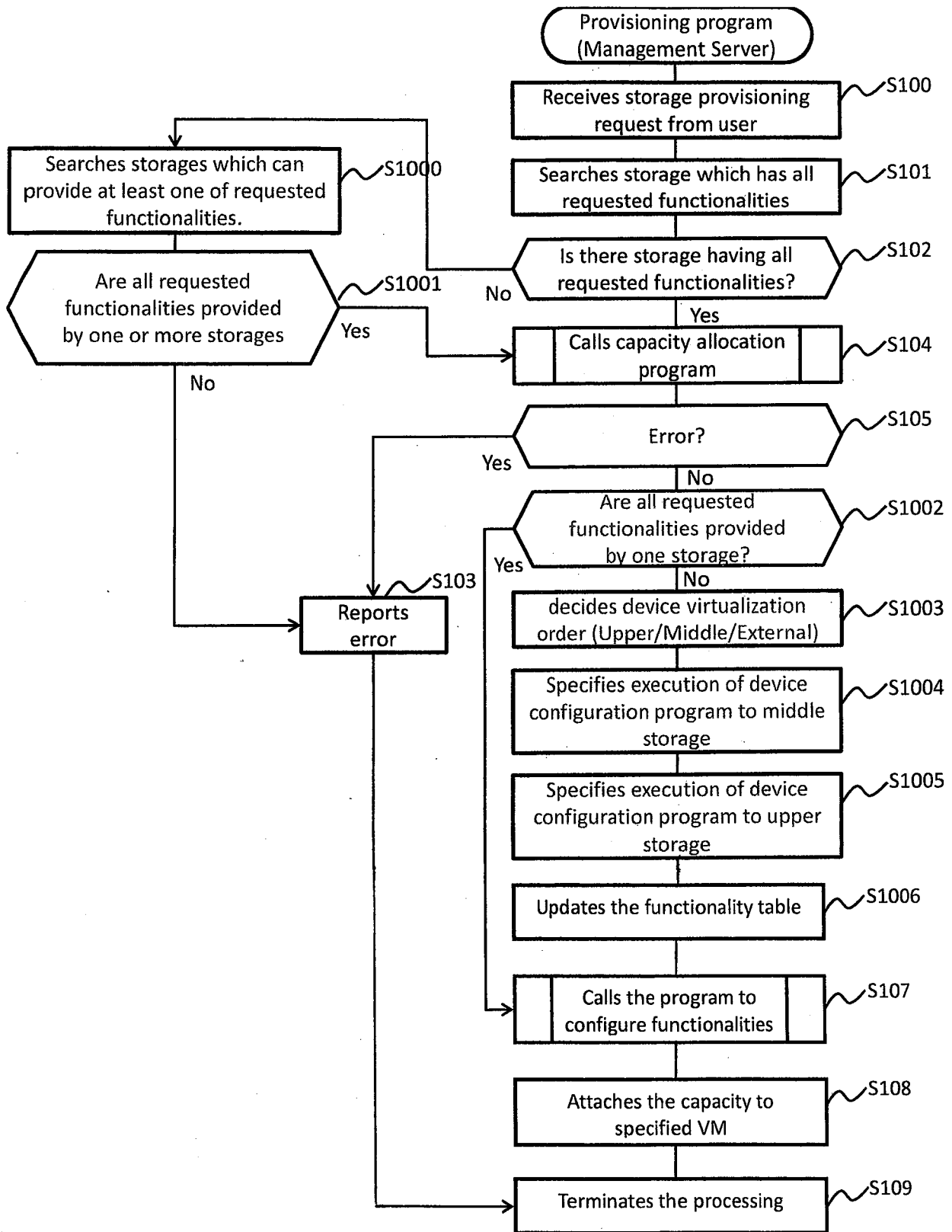


Fig. 20

Volume ID	Functionalities	Storage
1	Remote copy	Node ID 1
	Replication	Node ID 2
2	NAS	Node ID 3
	Remote Copy	Node ID 4
3	Cache	Node ID 5
	Replication	Node ID 2
...	...	...

Fig.21

INTERNATIONAL SEARCH REPORT

14/09/2014
International application No.

PCT/US 14/36556

A. CLASSIFICATION OF SUBJECT MATTER

IPC(8) - G06F 15/16 (2014.01)

CPC - H04L 29/08072

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)

IPC: G06F 15/16 (2014.01)

CPC: H04L 29/08072

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

USPC: 709/217; 707/999.01; 707/812; 718/1; 709/245; 711/100; CPC: H04L 29/08072; H04L 29/06; H04L 29/0809; G06Q 30/02; H04L 29/08981 (keyword limited - see search terms below)

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

PatBase; Google; Google Scholar

Terms: storage, provision, allocate, virtual, node, drive, device, capacity, size, functionality, capability, attribute, mapping, request, heterogeneous, layer, tier, swap, migrate, file, object, system, store, interface, volume.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 7,953,929 B1 (Chatterjee et al.) 31 May 2011 (31.05.2011), entire document, especially abstract, Fig. 3, col. 7, ln 51-67, col. 8, ln 1-26, col. 8, ln 49-54, col. 10, ln 36-51, col. 11, ln 45-51, col. 11, ln 52 to col. 12, ln 15.	1-18
Y	US 8,200,935 B2 (Noguchi et al.) 12 June 2012 (12.06.2012), entire document, especially abstract, col. 1, ln 34-46, col. 3, ln 25-29, col. 3, ln 61 to col. 4, ln 3, col. 5, ln 51-59.	1-18
Y	US 6,839,746 B1 (Muthiyan et al.) 04 January 2005 (04.01.2005), entire document, especially abstract, col. 2, ln 21-23, col. 4, ln 4-18, col. 5, ln 21-29, col. 5, ln 44-58.	4, 13
A	US 7,433,948 B2 (Edsall et al.) 07 October 2008 (07.10.2008), entire document.	1-18
A	US 5,758,050 A (Brady et al.) 26 May 1998 (26.05.1998), entire document	1-18
A	US 20130166839 A1 (Burton et al.) 27 June 2013 (27.06.2013), entire document	1-18
A	US 20110022812 A1 (van der Linden et al.) 27 January 2011 (27.01.2011), entire document	1-18

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

* 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

01 September 2014 (01.09.2014)

Date of mailing of the international search report

25 SEP 2014

Name and mailing address of the ISA/US

Mail Stop PCT, Attn: ISA/US, Commissioner for Patents
P.O. Box 1450, Alexandria, Virginia 22313-1450
Facsimile No. 571-273-3201

Authorized officer:

Lee W. Young

PCT Helpdesk: 571-272-4300
PCT OSP: 571-272-7774