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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
01 March 2018 (01.03.2018)

WIPO | PCT

(10) International Publication Number
WO 2018/037293 A1

(51) International Patent Classification:
*G06F 3/06* (2006.01)

(21) International Application Number:
PCT/IB2017/053501

(22) International Filing Date:
13 June 2017 (13.06.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
15/243,692 22 August 2016 (22.08.2016) US

(71) Applicant: **AMPLIDATA N.V.** [BE/BE]; Ottergemsesteenweg-Zuid 808 bus B43, 9000 Gent (BE).

(72) Inventors: **DE WAEGENEER, Wim**; c/o Western Digital Technologies, Inc., 5601 Great Oaks Parkway, San Jose, California 95119 (US). **DEMULDER, Timothy**; c/o Western Digital Technologies, Inc., 5601 Great Oaks Parkway, San Jose, California 95119 (US).

(74) Agent: **PLAS, Axel** et al.; IP Hills NV, Hubert Frère-Orbanlaan 329, 9000 Gent (BE).

(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, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, 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))*

(54) Title: AUTOMATIC RAID PROVISIONING

(57) Abstract: The present disclosure relates to systems and methods for automatic provisioning of a redundant array of independent disks (RAID). In particular, the present disclosure relates to methods including automatically determining, using one or more processors, that a first partition associated with a first storage device and a second partition associated with a second storage device are associated with a common mount point; and automatically adding, using the one or more processors, the first partition and the second partition to a RAID.

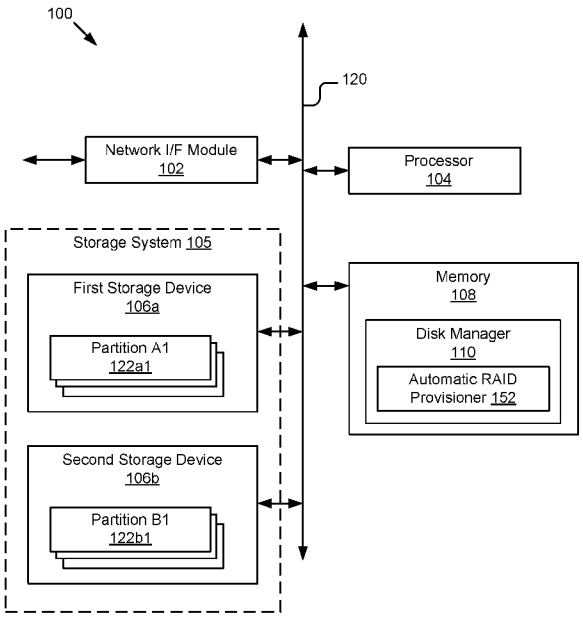


Figure 1

WO 2018/037293 A1

# Automatic RAID Provisioning

## FIELD OF THE INVENTION

[0001] The present disclosure relates to provisioning digital storage devices. In particular, the present disclosure relates to systems and methods of automatic redundant array of independent disks (RAID) provisioning.

## BACKGROUND

[0002] A redundant array of independent disks (RAID) combines multiple physical storage devices, or partitions thereof, into a single logical unit in order to provide redundancy, improved performance, improved availability, or a combination thereof. Provisioning of storage devices (e.g., disks) and creation or modification of a RAID using existing solutions is error-prone. Errors such as user errors inputting static information are not uncommon. Accordingly, a model-based approach that minimizes the amount of input and intervention a user needs to provide may be desirable.

[0003] Additionally, difficulties arise when a storage device that is part of a RAID is replaced, particularly when the replacement storage device is a different size and/or model type. Accordingly, a mechanism that reduces the difficulties associated with reassembling a RAID when a storage device is replaced may be desirable.

[0004] At least one system and method are disclosed herein that address one or more of the aforementioned issues.

## SUMMARY

[0005] Systems and methods for automatic redundant array of independent disks (RAID) provisioning are disclosed. For example, in some implementations, a system includes one or more processors and a memory storing instructions that cause the system to: automatically determine that a first partition associated with a first storage device and a second partition associated with a second storage device are associated with a common mount point; and automatically add the first partition and the second partition to a RAID.

[0006] These and other implementations may each optionally include one or more of the following features, such as: receiving a role of the first storage device, determining a first partition template based on the role of the first storage device, and applying the first partition template to the first storage device; receiving a role of a second storage device, determining a second partition template based on the role of the second storage device, and applying the second partition template to the second storage device; the first partition template defining one or more partitions and a mount point associated with each of the one or more partitions; automatically determining that a third partition associated with a third storage device is associated with the common mount point, and automatically adding the third partition to the RAID; the third storage device replacing one of the first storage device and the second storage device and the addition of the third partition to the RAID re-assembling the RAID; the third storage device being in addition to the first storage device and the second storage device.

[0007] Other implementations of one or more of these aspects include corresponding systems, apparatus, methods, and computer programs, configured to perform the actions of the methods, encoded on computer storage devices. It should be understood that the above is not all-inclusive and many additional steps, features and advantages are contemplated and fall within the scope of the present disclosure. Moreover, it should be understood that the language used in the present disclosure has been principally selected for readability and instructional purposes and not to limit the scope of the subject matter disclosed herein.

## BRIEF DESCRIPTION OF THE DRAWINGS

[0008] The techniques introduced herein are illustrated by way of example, and not by way of limitation, in the figures of the accompanying drawings in which like reference numerals are used to refer to similar elements.

[0009] Figure 1 is a block diagram illustrating an example system configured to automatically provision a redundant array of independent disks (RAID).

[0010] Figure 2 is a block diagram illustrating an example automatic RAID provisioner according to one implementation.

[0011] Figure 3 is a flowchart depicting an example method for automatic RAID provisioning according to the techniques described herein.

**[0012]** Figure 4 is a flowchart depicting an example method for automatic re-assembly of a RAID array according to the techniques described herein.

## DETAILED DESCRIPTION

**[0013]** Various systems and methods for automatic redundant array of independent disks (RAID) provisioning are disclosed. While the systems and methods of the present disclosure are described in the context of a particular system architecture, it should be understood that the systems and methods can be applied to other architectures and organizations of hardware.

**[0014]** Figure 1 is a block diagram illustrating an example system 100 configured to implement the techniques introduced herein. The system 100 may be any computing device including one or more memory and one or more processors (for example, a server computer, a laptop computer, a desktop computer, etc.).

**[0015]** The system 100 may include a network interface (I/F) module 102, a processor 104, a storage system 105, and a memory 108. The components of the system 100 are communicatively coupled to a bus or software communication mechanism 120 for communication with each other.

**[0016]** In some implementations, software communication mechanism 120 may be an object bus (e.g., Common Object Request Broker Architecture (CORBA)), direct socket communication (e.g., Transmission Control Protocol/Internet Protocol (TCP/IP) sockets) among software modules, remote procedure calls, User Datagram Protocol (UDP) broadcasts and receipts, Hypertext Transfer Protocol (HTTP) connections, function or procedure calls, etc. Further, any or all of the communication could be secure (Secure Shell (SSH), HTTP Secure (HTTPS), etc.). The software communication mechanism 120 can be implemented on any underlying hardware (for example, a network, the Internet, a bus, a combination thereof, etc.).

**[0017]** The network interface (I/F) module 102 is configured to connect system 100 to a network and/or other system. For example, network interface module 102 may enable communication through one or more of the Internet, cable networks, and wired networks. The network interface module 102 links the processor 104 to a network that may in turn be coupled to other processing systems (e.g., a client device, a server device). The network interface module 102 also provides other conventional connections to the network for

distribution and/or retrieval of files and/or media objects using standard network protocols such as, but not limited to, TCP/IP, HTTP, HTTPS and Simple Mail Transfer Protocol (SMTP) as will be understood. In some implementations, the network interface module 102 includes a transceiver for sending and receiving signals using Wi-Fi, Bluetooth®, or cellular communications for wireless communication.

**[0018]** The network interface module 102 may allow a client and/or a server device to store data to and retrieve data from the storage system 105 (for example, across a network). For example, the network interface module 102 may allow one or more components of the system 100 to act as network accessed storage or as an array of storage devices.

**[0019]** The network can be one of a conventional type, wired or wireless, and may have numerous different configurations including a star configuration, token ring configuration, or other configurations. Furthermore, the network may include a local area network (LAN), a wide area network (WAN) (e.g., the Internet), and/or other interconnected data paths across which multiple devices may communicate. In some implementations, the network may be a peer-to-peer network. The network may also be coupled with or include portions of a telecommunications network for sending data using a variety of different communication protocols. In some implementations, the network may include Bluetooth (or Bluetooth low energy) communication networks or a cellular communications network for sending and receiving data including via short messaging service (SMS), multimedia messaging service (MMS), HTTP, direct data connection, wireless application protocol (WAP), email, etc.

**[0020]** The processor 104 may include an arithmetic logic unit, a microprocessor, a general purpose controller or some other processor array to perform computations and provide electronic display signals to a display device. In some implementations, the processor 104 is a hardware processor having one or more processing cores. The processor 104 is coupled to the bus 120 for communication with the other components of system 100. Processor 104 processes data signals and may include various computing architectures including a complex instruction set computer (CISC) architecture, a reduced instruction set computer (RISC) architecture, or an architecture implementing a combination of instruction sets. Although only a single processor is shown in the example of Figure 1, multiple processors and/or processing cores may be included. It should be understood that other processor configurations are possible.

**[0021]** The memory 108 stores instructions and/or data that may be executed by the processor 104. The memory 108 is coupled to the bus 120 for communication with the other components of the system 100. The instructions and/or data stored in the memory 108 may include code for performing any and/or all of the techniques described herein. The memory 108 may be, for example, volatile memory such as a dynamic random access memory (DRAM) device, a static random access memory (SRAM) device, or some other volatile memory devices. In some implementations, the memory 108 also includes a non-volatile memory or similar permanent storage device and media (for example, a hard disk drive, a floppy disk drive, a compact disc read only memory (CD-ROM) device, a digital versatile disc read only memory (DVD-ROM) device, a digital versatile disc random access memory (DVD-RAM) device, a digital versatile disc rewritable (DVD-RW) device, a flash memory device, or some other non-volatile storage device). In some implementations, the memory 108 stores a disk manager 110. In the illustrated implementation, the disk manager 110 includes an automatic RAID provisioner 152.

**[0022]** The storage system 105 is a data storage system that may store information on any type of attached array of writable storage media, occasionally referred to herein as "storage device" in the singular or "storage devices" in the plural. Examples of storage devices include, but are not limited to, magnetic disk or tape, optical disk (e.g., CD-ROM or DVD), flash memory, solid-state drive (SSD), electronic random access memory (RAM), micro-electro mechanical and/or any other similar media adapted to store information, including data and parity information. However, as illustratively described herein, the information is stored on disks of hard disk drives (HDDs). Figure 1 illustrates a first storage device 106a and a second storage device 106b. Reference to a storage device 106 or storage devices 106 (without the 'a' or 'b' identifier appended to 106 and/or without being preceded by "first" or "second") is referring to a general instance or instances of a storage device 106. The storage system 105 may include one or more enclosures housed in the same location or physically distributed to disparate locations across a storage area network. The storage devices 106 are partitioned into one or more partitions 122.

**[0023]** Figure 2 is a block diagram illustrating an example automatic RAID provisioner 152 according to one implementation. In the illustrated implementation, the automatic RAID provisioner 152 includes a model module 202, a template-based partitioner

204 and a RAID creator 206. However, it should be recognized that Figure 2 is merely one implementation. Other implementations may include different components or organizations.

**[0024]** The model module 202 manages one or more models. Management of a model may include one or more of creating a model, receiving a role and identifying a model associated with that role, and identifying, based on the model, a template associated with that role.

**[0025]** In one implementation, a model defines a template associated with a role of a storage device 106 in the system 100. A template, also referred to herein as a "partition template," defines one or more partitions to be provisioned to a storage device 106 when the template is applied. In some implementations, a model may be user-created and/or modifiable. For example, in one implementation, the model module 202 may receive user input (e.g., via a command line interface or graphical user interface (GUI)) creating a model that defines a template including 'grub,' 'root,' 'swap,' 'var' and 'meta' partitions and each partition's mount point and size associated with an "operating system disk with an additional partition for metadata" role. In some implementations, a model may be preset (e.g., hardcoded or as a default).

**[0026]** In some implementations, the template defines a size for one or more of the partitions to be provisioned. In some implementations, a size of a partition may be proportional (e.g., a percentage of the total size of the storage device 106 being provisioned). In some implementations, a size of a partition may be absolute (e.g., a given number of bytes). In some implementations, a size of a partition may be constrained (e.g., the partition may have a minimum size, a maximum size or both).

**[0027]** The model module 202 receives a role of a storage device 106 and determines, using a model, a predefined partition template associated with that role. In some implementations, the model module 202 receives a role of a storage device 106 from a user. For example, in some implementations, the model module 202 receives input (e.g., via a command line interface or GUI) identifying the first storage device 106a as having a role of an "operating system disk with an additional partition for metadata." In some implementations, the model module 202 receives a role of a storage device 106 by automatically determining (e.g., via heuristics or rules) the role of the storage device 106. For example, assume that the first storage device 106a is an empty HDD. In some implementations, the model module 202 determines that the role of the first storage device

106a is as an "operating system disk with an additional partition for metadata," which is associated with a template (e.g., the template with the partitions described above).

**[0028]** The template-based partitioner 204 applies a template to a storage device 106. For example, assume that the first storage device 106a and the second storage device 106b are two empty HDDs to be provisioned. Also, assume that the model module 202 determines that the role of the first storage device 106a is as an "operating system disk with an additional partition for metadata" and the role of the second storage device 106b is as an "operating system disk with an additional partition for metadata." In one implementation, the template-based partitioner 204 applies a template including 'grub,' 'root,' 'swap,' 'var' and 'meta' partitions and each partition's mount point and size. For example, the template-based partitioner 204 populates a disk partition layout for the first storage device 106a as:

**[0029]**
```
{'buslocation': 'sg1',
 'device': 'sda',
 'disktype': 'HDD',
 'serial': 'HARDDISK_VBedf551cb-d6d08cb2',
 'size': 122880,
 'partition': [
   {'label': 'grub', 'mountpoint': '', 'partbegin': 1,    'partend': 512,
'partnumber': 1}
   {'label': 'root', 'mountpoint': '/', 'partbegin': 512,  'partend': 25003,
'partnumber': 2},
   {'label': 'swap', 'mountpoint': 'swap', 'partbegin': 25003, 'partend': 27003,
'partnumber': 3},
   {'label': 'var', 'mountpoint': '/var', 'partbegin': 27003, 'partend': 77003,
'partnumber': 4},
   {'label': 'meta', 'mountpoint': '/mnt/meta1', 'partbegin': 77003, 'partend':
122879, 'partnumber': 5}]
```

**[0030]** And, the template-based partitioner 204 populates a disk partition layout for the second storage device 106b as:

**[0031]**
{'buslocation': 'sg2',
'device': 'sdb',
'disktype': 'HDD',
'serial': 'HARDDISK_VBd9d5a5eb-7ab69280',
'size': 122880,
'partition': [
{'label': 'grub', 'mountpoint': '', 'partbegin': 1, 'partend': 512, 'partnumber': 1}
{'label': 'root', 'mountpoint': '/', 'partbegin': 512, 'partend': 25003, 'partnumber': 2},
{'label': 'swap', 'mountpoint': 'swap', 'partbegin': 25003, 'partend': 27003, 'partnumber': 3},
{'label': 'var', 'mountpoint': '/var', 'partbegin': 27003, 'partend': 77003, 'partnumber': 4},
{'label': 'meta', 'mountpoint': '/mnt/meta2', 'partbegin': 77003, 'partend': 122879, 'partnumber': 5}]

**[0032]** The RAID creator 206 determines whether two partitions are associated with a common mount point and creates a RAID for these partitions rather than indicating an error as existing systems would. For example, the RAID creator 206 detects that multiple partitions in the example above have the same mount point, so the RAID creator 206 creates a level 1 RAID for those partitions and updates the disk layout to remove the duplicate mount points. For example, the RAID creator 206 creates a RAID 'md0' for mount point '/' on the second partition of both disks:

**[0033]**
{'device': 'md0',
'label': 'root',
'members': ['/dev/sdb2', '/dev/sda2'],

'mountpoint': '/',

'status': 'ONLINE',

'uuid': '653ec1d4-5e63-4ef8-8215-7857976a0dd2'}

**[0034]** The RAID creator 206 creates a RAID 'md1' created for mount point '/var' on the fourth partition of both disks:

**[0035]** {'device': 'md1',

'label': 'var',

'members': ['/dev/sdb4', '/dev/sda4'],

'mountpoint': '/var',

'status': 'ONLINE',

'uuid': '3542289a-9a43-4a73-908c-ba26a7bdc86d'}

**[0036]** The RAID creator 206 creates a RAID 'md2' for a mirror swap for mount point 'swap' on the third partition of both disks:

**[0037]** {'device': 'md2',

'label': 'swap',

'members': ['/dev/sdb3', '/dev/sda3'],

'mountpoint': 'swap',

'status': 'ONLINE',

'uuid': 'f455e7f0-b9a7-4791-9b4d-1122ee2cb666'}

**[0038]** Disk resources:

{'buslocation': 'sg1',

'device': 'sda',

'disktype': 'HDD',

'serial': 'HARDDISK_VBedf551cb-d6d08cb2',

'size': 122880,

'partition': [

{'label': 'grub', 'mountpoint': '', 'partbegin': 1, 'partend': 512, 'partnumber': 1}

{'label': 'root', 'mountpoint': '', 'partbegin': 512, 'partend': 25003, 'partnumber': 2},

{'label': 'swap', 'mountpoint': '', 'partbegin': 25003, 'partend': 27003, 'partnumber': 3},

{'label': 'var', 'mountpoint': '', 'partbegin': 27003, 'partend': 77003, 'partnumber': 4},

{'label': 'meta', 'mountpoint': '/mnt/meta1', 'partbegin': 77003, 'partend': 122879, 'partnumber': 5}]

{'buslocation': 'sg2',

'device': 'sdb',

'disktype': 'HDD',

'serial': 'HARDDISK_VBd9d5a5eb-7ab69280',

'size': 122880,

'partition': [

{'label': 'grub', 'mountpoint': '', 'partbegin': 1, 'partend': 512, 'partnumber': 1}

{'label': 'root', 'mountpoint': '', 'partbegin': 512, 'partend': 25003, 'partnumber': 2},

{'label': 'swap', 'mountpoint': '', 'partbegin': 25003, 'partend': 27003, 'partnumber': 3},

{'label': 'var', 'mountpoint': '', 'partbegin': 27003, 'partend': 77003, 'partnumber': 4},

{'label': 'meta', 'mountpoint': '/mnt/meta2', 'partbegin': 77003, 'partend': 122879, 'partnumber': 5}]

**[0039]** In one implementation, after the automatic modelling, the RAID user, via the RAID creator 206, may modify or remove the proposed RAIDs before provisioning the actual system. The provisioning first deploys the partition layout on the disks (using the template-based partitioner 204) and then creates (using the RAID creator 206) the modeled RAIDs. The example above is merely an example of automatically creating an array and is provided for clarity and convenience. It should be recognized that other examples exist and are within the scope of this disclosure.

**[0040]** In some implementations, the automatic RAID provisioner 152 may add additional disks, or partitions thereof, to an existing RAID. For example, continuing the example discussed above, assume a new storage device 106 (e.g., a third HDD) is added to the system 100 (not shown). Also assume the model module 202 determines that the new storage device may have a role of "operating system disk" and, based on the model, the partition template defines the following partition layout, which is implemented by the template-based partitioner 204:

**[0041]** {'buslocation': 'sg3',
'device': 'sdc',
'disktype': 'HDD',
'serial': 'HARDDISK_VBedf551cb-f8sdf99d',
'size': 200000,
'partition': [
{'label': 'grub', 'mountpoint': '', 'partbegin': 1, 'partend': 512, 'partnumber': 1}
{'label': 'root', 'mountpoint': '/', 'partbegin': 512, 'partend': 25003, 'partnumber': 2},
{'label': 'meta', 'mountpoint': '/mnt/meta3', 'partbegin': 77003, 'partend': 200000, 'partnumber': 3}]

**[0042]** In some implementations, the RAID creator 206 again detects a duplicate mount point (i.e. '/'), but now between an existing RAID (e.g., md0) and a storage device 106 not currently in the RAID (e.g., sdc) instead of between two storage devices 106 not in a RAID (e.g., prior to the creation of the RAID md0). The RAID creator 206 automatically adds the partition associated with the duplicate mount point to the existing RAID.

**[0043]** RAID resource:

{'device': 'md0',
'label': 'root',
'members': ['/dev/sdb2', '/dev/sda2', '/dev/sdc2'],
'mountpoint': '/',
'status': 'ONLINE',
'uuid': '653ec1d4-5e63-4ef8-8215-7857976a0dd2'},

**[0044]** Disk resource: mount point '/' is removed from the second partition of disk 'sdc'

{'buslocation': 'sg3',
'device': 'sdc',
'disktype': 'HDD',
'serial': 'HARDDISK_VBedf551cb-f8sdf99d',
'size': 200000,
'partition': [
{'label': 'grub', 'mountpoint': '', 'partbegin': 1, 'partend': 512, 'partnumber': 1}
{'label': 'root', 'mountpoint': '', 'partbegin': 512, 'partend': 25003, 'partnumber': 2},
{'label': 'meta', 'mountpoint': '/mnt/meta3', 'partbegin': 77003, 'partend': 200000, 'partnumber': 3}]

**[0045]** In some implementations, the automatic RAID provisioner 152 may re-assemble a RAID when a storage device 106 (e.g., a disk) is replaced even when the new storage device does not exactly match the original storage device 106 being replaced in one or more of size and model type. In some implementations, the re-assembly of a RAID is similar to the addition of an additional storage device 106 except that a storage device 106 that was already provisioned and included in the array is physically removed from the system and the one or more RAIDs to which the storage device 106 belonged.

**[0046]** Figures 3 and 4 are example methods that may be performed by the system of Figures 1 and 2 according to one implementation. It should be recognized, however, that the methods of Figures 3 and 4 are merely examples, and the disclosure herein applies to other methods which may include additional, fewer or different steps and may perform those steps in a different order.

**[0047]** Figure 3 is a flowchart of an example method 300 for automatic RAID provisioning according to one implementation. At 302, the model module 202 of the automatic RAID provisioner 152 receives a role of a first storage device 106a. At 304, the model module 202 receives a role of a second storage device 106b. At 306, the model module 202 determines a partition template based on the role of the first storage device 106a. At 308, the model module 202 determines a partition template based on the role of the second storage device 106b. At 310, the template-based partitioner 204 applies the partition template, which is determined based on the role of the first storage device 106a, to the first storage device 106a. At 312, the template-based partitioner 204 applies the partition template, which is determined based on the role of the second storage device 106b, to the second storage device 106b. At 314, the RAID creator 206 determines that a first partition associated with the first storage device (e.g., Partition A1 122a1) and a second partition associated with the second storage device (e.g., Partition B1 122b1) are associated with a common mount point. At 316, the RAID creator 206 automatically adds the first partition associated with the first storage device (e.g., Partition A1 122a1) and the second partition associated with the second storage device (e.g., Partition B1 122b1) associated with a common mount point to a RAID.

**[0048]** Figure 4 is a flowchart of an example method 400 for automatic re-assembly of a RAID according to one implementation. At 402, the model module 202 of the automatic RAID provisioner 152 receives a role associated with a first storage device 106a. At 404, the

model module 202 determines a partition template based on the role of the first storage device 106a. At 406, the template-based partitioner 204 applies the partition template, which is determined based on the role of the first storage device 106a, to the first storage device 106a. At 408, the RAID creator 206 automatically determines that a partition associated with the first storage device includes a mount point in common with a partition of a second storage device, the second storage device and a third storage device previously belonging to a RAID, the third storage device having been physically removed from the system. At 410, the RAID creator 206 automatically re-assembles the RAID, where the first storage device replaces the third storage device so the re-assembled RAID includes the partition of the first storage device and the partition of the second storage device.

**[0049]** In the above description, for purposes of explanation, numerous specific details were set forth. It will be apparent, however, that the disclosed technologies can be practiced without any given subset of these specific details. In other instances, structures and devices are shown in block diagram form. For example, the disclosed technologies are described in some implementations above with reference to user interfaces and particular hardware.

**[0050]** Reference in the specification to "one implementation" or "an implementation" means that a particular feature, structure, or characteristic described in connection with the implementation is included in at least one implementation of the disclosed technologies. The appearances of the phrase "in one implementation" in various places in the specification are not necessarily all referring to the same implementation.

**[0051]** Some portions of the detailed descriptions above may be presented in terms of processes and symbolic representations of operations on data bits within a computer memory. A process can generally be considered a self-consistent sequence of steps leading to a result. The steps may involve physical manipulations of physical quantities. These quantities take the form of electrical or magnetic signals capable of being stored, transferred, combined, compared, and otherwise manipulated. These signals may be referred to as being in the form of bits, values, elements, symbols, characters, terms, numbers or the like.

**[0052]** These and similar terms can be associated with the appropriate physical quantities and can be considered labels applied to these quantities. Unless specifically stated otherwise as apparent from the prior discussion, it is appreciated that throughout the description, discussions utilizing terms for example "processing" or "computing" or

"calculating" or "determining" or "displaying" or the like, may refer to the action and processes of a computer system, or similar electronic computing 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 memories or registers or other such information storage, transmission or display devices.

**[0053]** The disclosed technologies may also relate to an apparatus for performing the operations herein. This apparatus may be specially constructed for the required purposes, or it may include a general-purpose computer selectively activated or reconfigured by a computer program stored in the computer. Such a computer program may be stored in a computer readable storage medium, for example, but is not limited to, any type of disk including floppy disks, optical disks, CD-ROMs, and magnetic disks, read-only memories (ROMs), random access memories (RAMs), Erasable Programmable Read-Only Memories (EPROMs), Electrically Erasable Programmable Read-Only Memories (EEPROMs), magnetic or optical cards, flash memories including USB keys with non-volatile memory or any type of media suitable for storing electronic instructions, each coupled to a computer system bus.

**[0054]** The disclosed technologies can take the form of an entirely hardware implementation, an entirely software implementation or an implementation containing both hardware and software elements. In some implementations, the technology is implemented in software, which includes but is not limited to firmware, resident software, microcode, etc.

**[0055]** Furthermore, the disclosed technologies can take the form of a computer program product accessible from a non-transitory computer-usable or computer-readable medium providing program code for use by or in connection with a computer or any instruction execution system. For the purposes of this description, a computer-usable or computer-readable medium can be any apparatus that can contain, store, communicate, propagate, or transport the program for use by or in connection with the instruction execution system, apparatus, or device.

**[0056]** A computing system or data processing system suitable for storing and/or executing program code will include at least one processor (e.g., a hardware processor) coupled directly or indirectly to memory elements through a system bus. The memory elements can include local memory employed during actual execution of the program code,

bulk storage, and cache memories which provide temporary storage of at least some program code in order to reduce the number of times code must be retrieved from bulk storage during execution.

**[0057]** Input/output or I/O devices (including but not limited to keyboards, displays, pointing devices, etc.) can be coupled to the system either directly or through intervening I/O controllers.

**[0058]** Network adapters may also be coupled to the system to enable the data processing system to become coupled to other data processing systems or remote printers or storage devices through intervening private or public networks. Modems, cable modems and Ethernet cards are just a few of the currently available types of network adapters.

**[0059]** Finally, the processes and displays presented herein may not be inherently related to any particular computer or other apparatus. Various general-purpose systems may be used with programs in accordance with the teachings herein, or it may prove convenient to construct more specialized apparatus to perform the required method steps. The required structure for a variety of these systems will appear from the description above. In addition, the disclosed technologies were 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 technologies as described herein.

**[0060]** The foregoing description of the implementations of the present techniques and technologies has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the present techniques and technologies to the precise form disclosed. Many modifications and variations are possible in light of the above teaching. It is intended that the scope of the present techniques and technologies be limited not by this detailed description. The present techniques and technologies may be implemented in other specific forms without departing from the spirit or essential characteristics thereof. Likewise, the particular naming and division of the modules, routines, features, attributes, methodologies and other aspects are not mandatory or significant, and the mechanisms that implement the present techniques and technologies or its features may have different names, divisions and/or formats. Furthermore, the modules, routines, features, attributes, methodologies and other aspects of the present technology can be implemented as software, hardware, firmware or any combination of the three. Also, wherever a component, an example of which is a module, is implemented as software, the component can be

implemented as a standalone program, as part of a larger program, as a plurality of separate programs, as a statically or dynamically linked library, as a kernel loadable module, as a device driver, and/or in every and any other way known now or in the future in computer programming. Additionally, the present techniques and technologies are in no way limited to implementation in any specific programming language, or for any specific operating system or environment. Accordingly, the disclosure of the present techniques and technologies is intended to be illustrative, but not limiting.

**WHAT IS CLAIMED IS:**

1. A computer-implemented method comprising:
automatically determining, using one or more processors, that a first partition associated with a first storage device and a second partition associated with a second storage device are associated with a common mount point; and
automatically adding, using the one or more processors, the first partition and the second partition to a redundant array of independent disks (RAID).

2. The computer-implemented method of claim 1 further comprising:
receiving a role of the first storage device;
determining a first partition template based on the role of the first storage device; and
applying the first partition template to the first storage device.

3. The computer-implemented method of claim 2 further comprising:
receiving a role of a second storage device;
determining a second partition template based on the role of the second storage device; and
applying the second partition template to the second storage device.

4. The computer-implemented method of claim 2, wherein the partition template defines one or more partitions and corresponding mount points associated with the one or more partitions.

5. The computer-implemented method of claim 1 further comprising:
automatically determining that a third partition associated with a third storage device is associated with the common mount point; and
automatically adding the third partition to the RAID.

6. The computer-implemented method of claim 5, wherein the third storage device replaces one of the first storage device and the second storage device, and the addition of the third partition to the RAID re-assembles the RAID.

7. The computer-implemented method of claim 5, wherein the third storage device is in addition to the first storage device and the second storage device.

8. A system comprising:

a processor; and

a non-transitory computer usable medium storing instructions that, when executed by the processor, cause the system to:

automatically determine that a first partition associated with a first storage device and a second partition associated with a second storage device are associated with a common mount point; and

automatically add the first partition and the second partition to a redundant array of independent disks (RAID).

9. The system of claim 8 wherein the instructions, when executed by the processor, further cause the system to:

receive a role of the first storage device;

determine a first partition template based on the role of the first storage device; and

apply the first partition template to the first storage device.

10. The system of claim 9 wherein the instructions, when executed by the processor, further cause the system to:

receive a role of a second storage device;

determine a second partition template based on the role of the second storage device; and

apply the second partition template to the second storage device.

11. The system of claim 9, wherein the partition template defines one or more partitions and corresponding mount points associated with the one or more partitions.

12. The system of claim 8 wherein the instructions, when executed by the processor, further cause the system to:

automatically determine that a third partition associated with a third storage device is associated with the common mount point; and

automatically add the third partition to the RAID.

13. The system of claim 12, wherein the third storage device replaces one of the first storage device and the second storage device, and the addition of the third partition to the RAID re-assembles the RAID.

14. The system of claim 12, wherein the third storage device is in addition to the first storage device and the second storage device.

15. A computer-program product comprising a non-transitory computer usable medium including a computer readable program, wherein the computer readable program, when executed on a computer, causes the computer to perform operations comprising:

automatically determining that a first partition associated with a first storage device and a second partition associated with a second storage device are associated with a common mount point; and

automatically adding the first partition and the second partition to a redundant array of independent disks (RAID).

16. The computer-program product of claim 15, wherein the computer readable program, when executed on a computer, further causes the computer to perform operations comprising:

receiving a role of the first storage device;

determining a first partition template based on the role of the first storage device; and

applying the first partition template to the first storage device.

17. The computer-program product of claim 16, wherein the computer readable program, when executed on a computer, further causes the computer to perform operations comprising:

receiving a role of a second storage device;

determining a second partition template based on the role of the second storage device; and

applying the second partition template to the second storage device.

18. The computer-program product of claim 16, wherein the partition template defines one or more partitions and corresponding mount points associated with the one or more partitions.

19. The computer-program product of claim 15, wherein the computer readable program, when executed on a computer, further causes the computer to perform operations comprising:

automatically determining that a third partition associated with a third storage device is associated with the common mount point; and

automatically adding the third partition to the RAID.

20. The computer-program product of claim 19, wherein the third storage device replaces one of the first storage device and the second storage device, and the addition of the third partition to the RAID re-assembles the RAID.

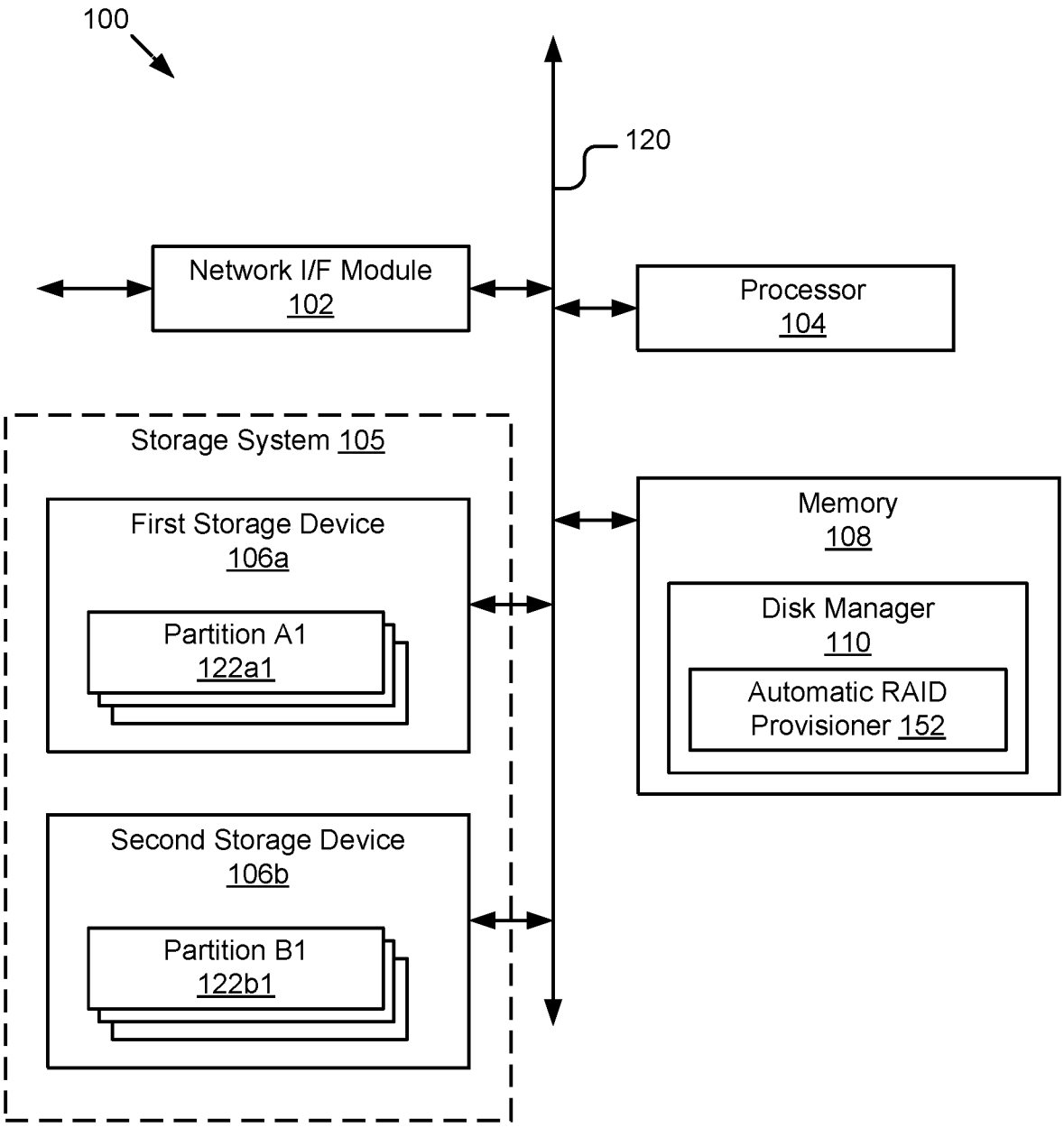
100
120
Network I/F Module
102
Processor
104
Storage System 105
First Storage Device
106a
Partition A1
122a1
Second Storage Device
106b
Partition B1
122b1
Memory
108
Disk Manager
110
Automatic RAID
Provisioner 152

Figure 1

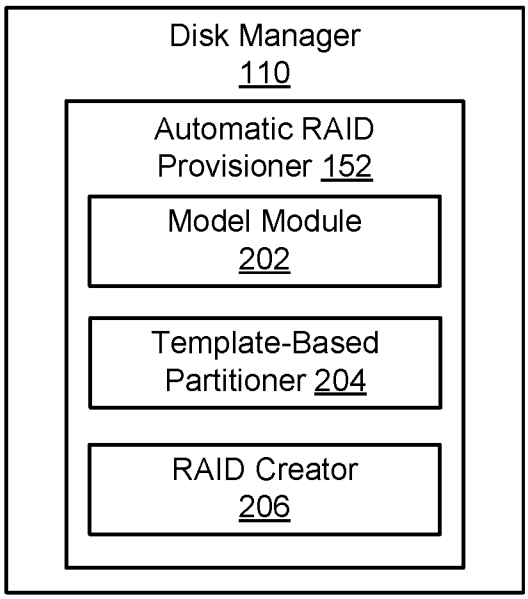
Disk Manager
110
Automatic RAID
Provisioner 152
Model Module
202
Template-Based
Partitioner 204
RAID Creator
206

Figure 2

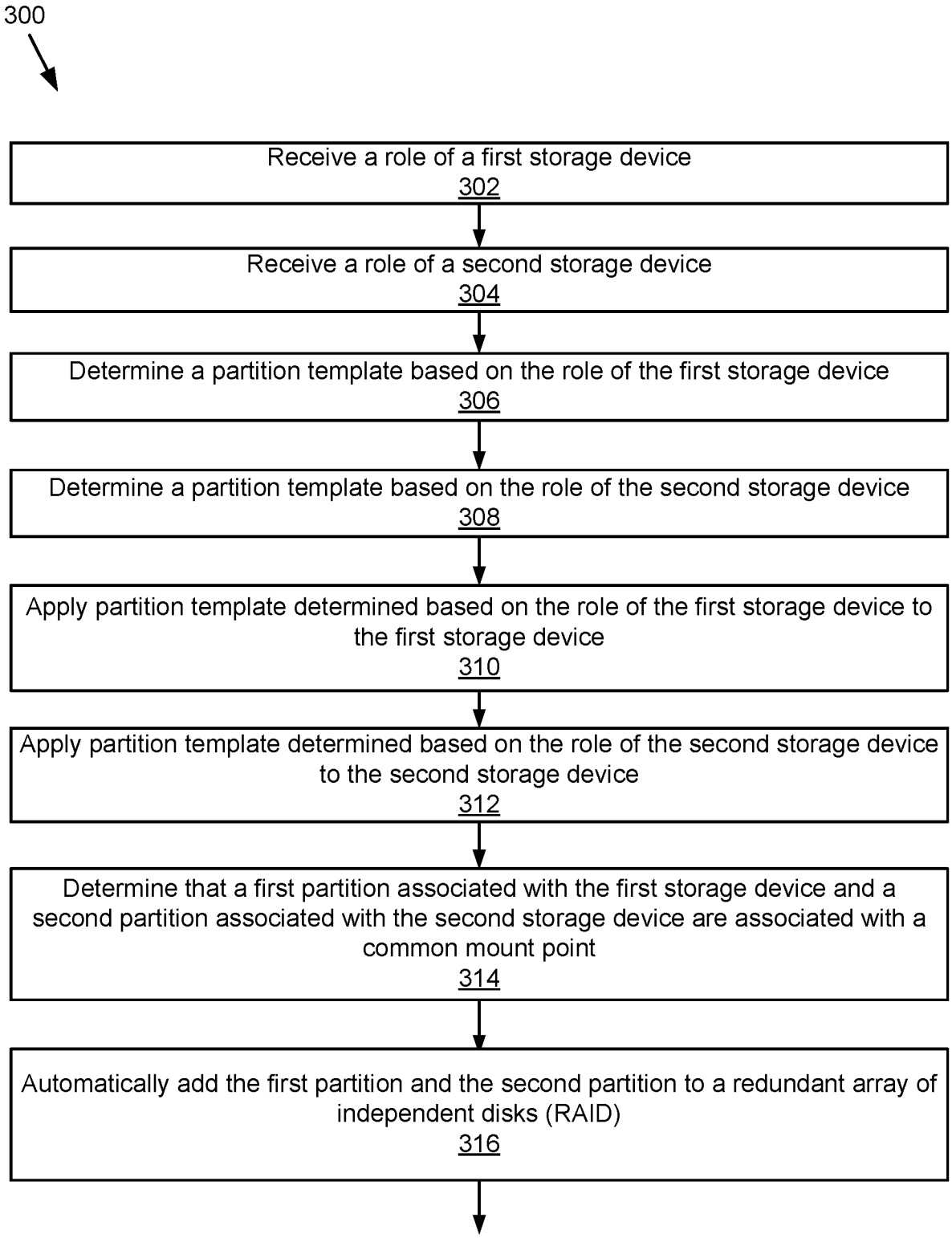
300
Receive a role of a first storage device
302
Receive a role of a second storage device
304
Determine a partition template based on the role of the first storage device
306
Determine a partition template based on the role of the second storage device
308
Apply partition template determined based on the role of the first storage device to the first storage device
310
Apply partition template determined based on the role of the second storage device to the second storage device
312
Determine that a first partition associated with the first storage device and a second partition associated with the second storage device are associated with a common mount point
314
Automatically add the first partition and the second partition to a redundant array of independent disks (RAID)
316

Figure 3

400

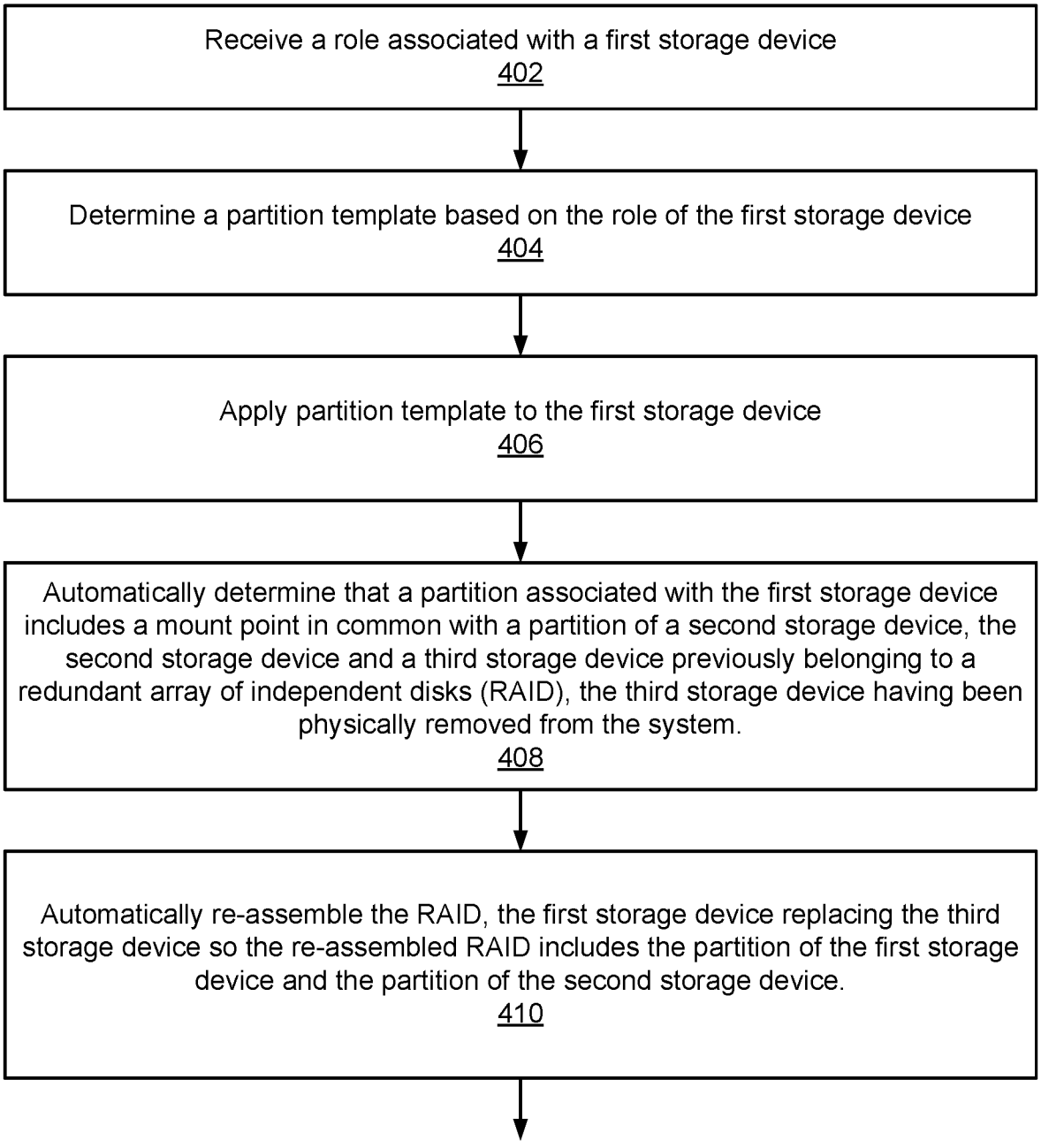
Receive a role associated with a first storage device
402
Determine a partition template based on the role of the first storage device
404
Apply partition template to the first storage device
406
Automatically determine that a partition associated with the first storage device includes a mount point in common with a partition of a second storage device, the second storage device and a third storage device previously belonging to a redundant array of independent disks (RAID), the third storage device having been physically removed from the system.
408
Automatically re-assemble the RAID, the first storage device replacing the third storage device so the re-assembled RAID includes the partition of the first storage device and the partition of the second storage device.
410

Figure 4

# INTERNATIONAL SEARCH REPORT

International application No
PCT/IB2017/053501

A. CLASSIFICATION OF SUBJECT MATTER
INV. G06F3/06
ADD.

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F

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

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

C. DOCUMENTS CONSIDERED TO BE RELEVANT

| Category* | Citation of document, with indication, where appropriate, of the relevant passages | Relevant to claim No. |
|---|---|---|
| X | US 7 454 566 B1 (OVERBY MARK A [US])<br>18 November 2008 (2008-11-18) | 1,5-8,<br>12-15,<br>19,20 |
| A | abstract<br>column 4, lines 6-55; figure 2B<br>column 5, line 13 - column 7, line 60;<br>figure 3A | 2-4,<br>9-11,<br>16-18 |
| | ----- | |
| A | US 6 119 131 A (CABRERA LUIS FELIPE [US]<br>ET AL) 12 September 2000 (2000-09-12)<br>abstract | 1-20 |
| | ----- | |

☐ 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 |
|---|---|
| 13 September 2017 | 25/09/2017 |

| Name and mailing address of the ISA/ | Authorized officer |
|---|---|
| European Patent Office, P.B. 5818 Patentlaan 2<br>NL - 2280 HV Rijswijk<br>Tel. (+31-70) 340-2040,<br>Fax: (+31-70) 340-3016 | De Ceulaer, Bart |

1

# INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/IB2017/053501

| Patent document cited in search report | Publication date | Patent family member(s) | Publication date |
|---|---|---|---|
| US 7454566 B1 | 18-11-2008 | US 7454566 B1 | 18-11-2008 |
| | | US 7512736 B1 | 31-03-2009 |
| US 6119131 A | 12-09-2000 | US 6119131 A | 12-09-2000 |
| | | US 6421684 B1 | 16-07-2002 |