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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
30 November 2017 (30.11.2017)

WIPO | PCT

(10) International Publication Number
WO 2017/204991 A1

(51) International Patent Classification:
G06F 17/30 (2006.01)

(21) International Application Number:
PCT/US2017/029982

(22) International Filing Date:
27 April 2017 (27.04.2017)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
15/167,908 27 May 2016 (27.05.2016) US

(71) Applicant: **INTEL CORPORATION** [US/US]; 2200 Mission College Boulevard, Santa Clara, California 95054 (US).

(72) Inventors: **FREYENSEE, James P.**; 2111 NE 25th Ave, Hillsboro, Oregon 97124 (US). **MINTURN, Dave B.**; 17125 SW Chehalem Way, Hillsboro, Oregon 97123 (US). **CAYTON, Phil C.**; 34940 Millard Road, Warren, Oregon 97053 (US). **STERNBERG, Jay E.**; 14101 NW Shadybrook Rd, North Plains, Oregon 97133 (US). **KNAPP, Anthony J.**; 4870 SW Main Ave #2, Beaverton, Oregon 97005 (US).

(74) Agent: **VICTOR, David W.**; Konrad Raynes Davda & Victor LLP, 350 S. Beverly Dr., Suite 360, Beverly Hills, California 90212 (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, DJ, 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, 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,

(54) Title: A COMPUTER PRODUCT, METHOD, AND SYSTEM TO DYNAMICALLY MANAGE STORAGE DEVICES ACCESSED REMOTELY OVER A NETWORK

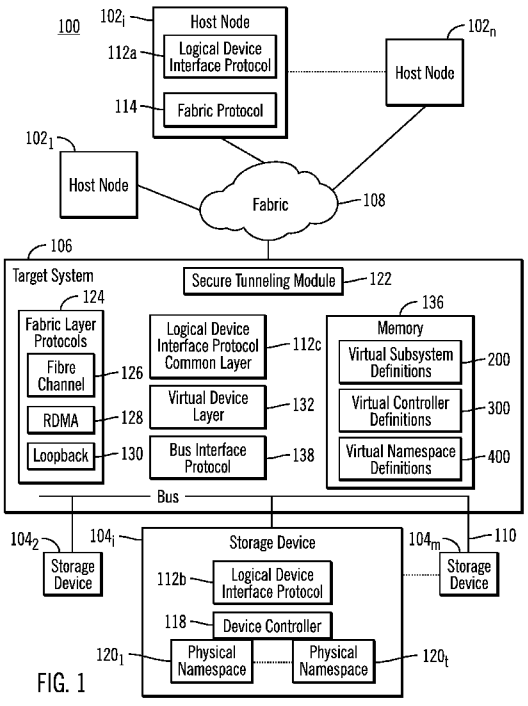


FIG. 1

(57) Abstract: Provided are a computer product, method, and system to generate a virtual subsystem to process read and write requests to storage devices. A virtual subsystem is configured in the memory to provide a representation of a subsystem presented to at least one host system. The at least one host system is to communicate with the virtual subsystem to access the at least one storage device. At least one virtual namespace is configured in the memory for the virtual subsystem. Each of the at least one virtual namespace maps to at least one physical namespace in at least one storage device. Each virtual namespace is assigned to one host system to use to direct read and write requests to the at least one physical namespace of the at least one storage device assigned to the virtual namespace.

[Continued on next page]

WO 2017/204991 A1

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))*

A COMPUTER PRODUCT, METHOD, AND SYSTEM TO DYNAMICALLY MANAGE STORAGE DEVICES ACCESSED REMOTELY OVER A NETWORK

TECHNICAL FIELD

Embodiments described herein generally relate to a computer product, method, and system to dynamically manage storage devices accessed remotely over a network.

BACKGROUND

Non-Volatile Memory Express (NVMe) is a logical device interface (http://www.nvmexpress.org) for accessing non-volatile storage media attached via a Peripheral Component Interconnect Express (PCIe) bus. The non-volatile storage media may comprise a flash memory and solid solid-state drives (SSDs). NVMe is designed for accessing low latency storage devices in computer systems, including personal and enterprise computer systems, and is also deployed in data centers requiring scaling of thousands of low latency storage devices. A computer system may communicate read/write requests over a network to a target system managing access to multiple attached storage devices, such as SSDs. The computer system may send communicate the read/write requests using an ANSI Standard Small Computer System Interface (SCSI) read/write command over the network. The receiving target system converts the SCSI read/write request to an NVMe request compatible with the SSD in the target system.

BRIEF DESCRIPTION OF THE DRAWINGS

Embodiments are described by way of example, with reference to the accompanying drawings, which are not drawn to scale, in which like reference numerals refer to similar elements.

FIG. 1 illustrates an embodiment of a storage environment

FIG. 2 illustrates an embodiment of a virtual subsystem definition to represent a subsystem to attached hosts. .

FIG. 3 illustrates an embodiment of a virtual controller definition within one of the virtual subsystems.

FIG. 4 illustrates a virtual namespace definition to define a virtual namespace that maps to a physical namespace/partition.

FIG. 5 illustrates an embodiment of a fabric packet.

FIGs. 6 and 7 illustrate embodiments of virtual subsystem configurations.

FIG. 8 illustrates an embodiment of operations to generate virtual subsystems, virtual controllers, and virtual namespaces.

FIGs. 9a and 9b illustrate an embodiment of operations to process a read/write request directed to a virtual namespace.

FIG. 10 illustrates an embodiment of a computer node architecture in which components may be implemented

DESCRIPTION OF EMBODIMENTS

A target system includes an NVMe subsystem with one or more controllers to manage read/write requests to namespace identifiers (NSID) defining ranges of addresses in the connected storage devices. The hosts may communicate to the NVMe subsystem over a fabric or network or a PCIe bus and port. Each network or PCIe port may connect to one NVMe subsystem, and the NVMe subsystem may communicate with the attached storage devices using a PCIe bus.

Described embodiments provide techniques for virtualizing subsystems, controllers, and namespaces to present to host systems to perform Input/Output (I/O) operations with respect to physical namespaces in storage devices assigned to virtual subsystems and virtual controllers. Host systems are assigned to virtual subsystems and to virtual namespaces that map to physical namespaces in the storage devices. The host transmits a read/write request using a logical device interface protocol, such as NVMe, to a target system in which the virtual subsystems and virtual namespaces are implemented. The target system translates the virtual namespace in the read/write request in the logical device interface protocol to the corresponding physical namespace and sends the

read/write request in the same logical device interface protocol, e.g., NVMe, to the storage device to process. For instance, the described embodiments may be deployed in NVMe-over-Fabric environments, where host systems communicate with an NVMe subsystem over a network interface (such as Ethernet, Remote Direct Memory Access (RDMA)-based networks, Fibre Channel based networks). A network port may connect to one NVMe subsystem, and the NVMe subsystem may communicate with the attached storages devices using a Peripheral Component Interconnect Express (PCIe) bus.

Described embodiments improve latency, because a read/write request from a host system that is directed to a storage device is included or encapsulated in a logical device interface protocol that can be processed directly at the storage device without having to be translated or converted from one protocol to another. In environments where the host communicates with the virtual subsystems over a fabric, such as an NVMe-over-Fabrics environment, the host may encapsulate the read/write request in the logical device interface protocol used at the storage device in a fabric packet for transmission through a fabric network. The target system receiving the fabric packet may then extract the read/write request encapsulated in the logical device interface protocol, which is further encapsulated in the fabric packet, and substitute the physical namespace to which the target virtual namespace in the read/write request maps before forwarding to the storage device over a bus, e.g., a PCIe bus. Described embodiments reduce latency in the transmission of the requests to the storage device by avoiding having to convert the read/write request from one protocol to another through the use of a virtual subsystem and virtual namespaces realized in the target system.

In the following description, numerous specific details such as logic implementations, opcodes, means to specify operands, resource partitioning/sharing/duplication implementations, types and interrelationships of system components, and logic partitioning/integration choices are set forth in order to provide a more thorough understanding of the present invention. It will be appreciated, however, by one skilled in the art that the invention may be practiced without such specific details. In other instances, control structures, gate

level circuits and full software instruction sequences have not been shown in detail in order not to obscure the invention. Those of ordinary skill in the art, with the included descriptions, will be able to implement appropriate functionality without undue experimentation.

References in the specification to "one embodiment," "an embodiment," "an example embodiment," etc., indicate that the embodiment described may include a particular feature, structure, or characteristic, but every embodiment may not necessarily include the particular feature, structure, or characteristic. Moreover, such phrases are not necessarily referring to the same embodiment.

Certain embodiments relate to storage device electronic assemblies. Embodiments include both devices and methods for forming electronic assemblies.

FIG. 1 illustrates an embodiment of a storage environment 100 having a plurality of host nodes $102_1 \ldots 102_n$ that communicate with multiple storage devices $104_1 \ldots 104_m$ via a target system 106. The host nodes $102_1 \ldots 102_n$ communicate with the target system 106 over a fabric 108, and the target system 106 communicates with the storage devices $104_1 \ldots 104_m$ over a bus interface 110. Each of the host nodes $102_1 \ldots 102_n$, includes, as shown with respect to host node 102_i, a logical device interface protocol 112a to perform read/write operations with respect to the storage devices $104_1 \ldots 104_m$ and a fabric protocol 114 to communicate over the fabric 108. Each of the storage devices $104_1 \ldots 104_m$ may include, as shown with respect to storage device 104_i, a logical device interface protocol 112b, the same as the logical device interface protocol 112a, a device controller 118 to perform storage device 104_i operations, and one or more physical namespaces $120_1 \ldots 120_t$. A physical namespace comprises a quantity of non-volatile memory that may be formatted into logical blocks. When formatted, a namespace of size n is a collection of logical blocks with logical block addresses from 0 to (n-1). The namespaces may further be divided into partitions or ranges of addresses. The physical namespaces $120_1 \ldots 120_t$ are identified by a namespace identifier (NSID) used by the device controller 118 to provide access to the namespace $120_1 \ldots 120_t$.

The target system 106 may include a secure tunneling mode module 122 for secure communications from the host nodes $102_1 \ldots 102_n$ and one or more fabric layer protocols 124, compatible with the fabric protocol 114 used by the host nodes $102_1 \ldots 102_n$. The target system 106 may support multiple fabric layer protocols, such as Fibre Channel 126, Remote Direct Memory Access (RDMA) 128, loopback interface 130, etc. The target system 106 includes an instance of the logical device interface protocol common layer 112c to process the logical device interface protocol communications communicated from the host nodes $102_1 \ldots 102_n$. The virtual device layer 132 generates and manages virtualization definitions in a target system memory 136 that provide representations of target hardware and physical namespaces to the host nodes $102_1 \ldots 102n$, including virtual subsystem definitions 200, virtual controller definitions 300, and virtualization namespace definitions 400. The target system 106 further includes a bus interface protocol 138 to transmit logical device interface protocol read/write requests to the storage devices $104_1 \ldots 104_m$.

In one embodiment, the logical device interface protocol 112a, 112b, 112c may comprise the Non-Volatile Memory Express (NVMe) protocol and the bus 110 and bus interface protocol 138 may comprise a Peripheral Component Interconnect Express (PCIe) bus. With described embodiments, a same NVMe read/write request may be transmitted from the host nodes $102_1 \ldots 102_n$ to the storage devices $104_1 \ldots 104_m$ without the need for conversion from one protocol to another. This use of the same logical device protocol request, e.g., NVMe, reduces latency in transmissions between the host nodes $102_1 \ldots 102_n$ and storage devices $104_1 \ldots 104_m$.

The address space of each of the namespaces in the storage devices $104_1 \ldots 104_m$ may be divided into one or more namespaces partitions, where each partition comprises a subset of addresses in a physical namespace $120_1 \ldots 120_t$. The host nodes $102_1 \ldots 102_n$ may further comprise any type of compute node capable of accessing storage partitions and performing compute operations.

The program components of the target system, such as 122, 124, 114c, 132, 138 may be implemented in a software program executed by a processor of the

target system 106, firmware, a hardware device, or in application specific integrated circuit (ASIC) devices, or some combination thereof.

The storage devices 104_1, 104_2....104_m may comprise electrically erasable and non-volatile memory cells, such as flash storage devices. For instance, the storage devices 104_1, 104_2....104_m may comprise NAND dies of flash memory cells. In one embodiment, the NAND dies may comprise a multilevel cell (MLC) NAND flash memory that in each cell records two bit values, a lower bit value and an upper bit value. Alternatively, the NAND dies may comprise single level cell (SLC) memories or three bit per cell (TLC) memories. The storage devices 104_1, 104_2....104_m may also comprise, but not limited to, ferroelectric random-access memory (FeTRAM), nanowire-based non-volatile memory, three-dimensional (3D) cross-point memory, phase change memory (PCM), memory that incorporates memristor technology, Magnetoresistive random-access memory (MRAM), Spin Transfer Torque (STT)-MRAM, a single level cell (SLC) Flash memory and other electrically erasable programmable read only memory (EEPROM) type devices. The storage devices 104_1, 104_2....104_m may also comprise a magnetic storage media, such as a hard disk drive etc. .

The target system memory 136 may comprise a non-volatile or volatile memory type of device known in the art, such as a Flash Memory, a non-volatile dual in-line memory module (NVDIMM), Dynamic Random Access Memories (DRAMs), etc., or may also comprise, but not limited to, ferroelectric random-access memory (FeTRAM), nanowire-based non-volatile memory, three-dimensional (3D) cross-point memory, phase change memory (PCM), memory that incorporates memristor technology, Magnetoresistive random-access memory (MRAM), Spin Transfer Torque (STT)-MRAM, a single level cell (SLC) Flash memory and other electrically erasable programmable read only memory (EEPROM) type devices, such as magnetic storage media, such as a hard disk drive etc. In certain embodiments, the target system memory 136 comprises a persistent, non-volatile storage of the virtual subsystem, virtual controller, and virtual namespace definitions to provide persistent storage over power cycle events.

FIG. 2 illustrates an embodiment of a virtual subsystem definition 200_i defining one of the virtual subsystems 200 configured in the target system 106, and includes: a virtual subsystem identifier (VSSID) 202; assigned hosts 204 comprising zero or more hosts assigned to use that virtual subsystem 202 to access the below storage devices 104_1 ...104_m; and one or more virtual controllers 300_1....300_t configured in the virtual subsystem 202.

FIG. 3 illustrates an embodiment of a virtual controller definition 300_i to define one of the virtual controllers 300 configured in the target system 106, and includes: a virtual controller (VC) identifier 302 and one or more virtual namespaces 400_1...400_p, identified by a VNSID, assigned to that virtual controller 302. A virtual controller 300i may also have no assigned virtual namespaces.

FIG. 4 illustrates an embodiment of a virtual namespace definition 400i, and includes: a virtual namespace identifier (VNSID) 402; an assigned host 404 assigned to use the VNSID 402 to access a physical namespace; one or more assigned physical namespace/partitions 406_1...406_p and the storage device 408_1...408_p including the physical namespace/partition 406_1...406_p. In certain embodiments, indication may be made in the assigned host 404 field or other part of the definition 400_i to allow any host, multiple hosts or no host to use the VNSID to access a physical namespace.

FIG. 5 illustrates an embodiment of a fabric packet 500 for transmission across the fabric 108 having a fabric layer 502, including fabric information such as a header, error correction codes, source and destination addresses, and other information required for transmission across the fabric 108, and an encapsulated logic device interface protocol request 504, including a request type command 506, e.g., read or write; a target namespace 508, which may indicate a virtual namespace ID (VNSID) or physical namespace ID (NSID) to which the request 506 is directed; and specific target addresses 510 subject to the read/write request, which may comprise one or more logical block addresses which are subject to the requested read/write operation. The logic device interface protocol request 504 may include additional fields and information to process the request.

The virtual device layer 132 may configure the virtual subsystems 200, virtual controllers 300, and virtual namespaces 400 in the target memory 136 to represent to the attached host nodes $102_1 \ldots 102_n$.

FIG. 6 illustrates an embodiment of a representation of a configuration of virtual subsystems $200_1 \ldots 200_n$ in the target system 106, where each virtual subsystem $200_1 \ldots 200_n$ may include, as shown with respect to virtual subsystem 200_1, one or more virtual controllers $300_1 \ldots 300_m$. Each virtual controller $300_1 \ldots 300_m$, as shown with respect to virtual controller 300_1, can include one or more assigned virtual namespace identifiers (VNSID) $400_1 \ldots 400_p$. Each virtual namespace identifier $400_1 \ldots 400_p$, maps to one or more physical the namespaces $120_1 \ldots 120_t$ in the storage devices $104_1 \ldots 104_m$, including a partition (range of addresses in the namespace) or the entire namespace. Each of the host nodes $102_1 \ldots 102_n$ are assigned to one or more virtual subsystems $200_1 \ldots 200_n$, and further to one or more virtual namespace IDs $400_1 \ldots 400_p$ in the virtual controllers $300_1 \ldots 300_m$ of the virtual subsystems $200_1 \ldots 200_n$ to which the host node 102_i is assigned. The host nodes $102_1 \ldots 102_n$ may access the physical namespace $120_1 \ldots 120_t$ partitions that map to the virtual namespace IDs $400_1 \ldots 400_p$ assigned to the hosts, where the host nodes $102_1 \ldots 102_n$ access the virtual namespace through the virtual controller 300_i to which the VNSID is assigned and virtual subsystem 200_i to which the host node is assigned.

FIG. 7 illustrates a further embodiment of a representation of a configuration of virtual subsystems 200_1 having multiple virtual controllers 300_1, 300_2, each having assigned VNSIDs 400_1, 400_2 and 400_3, respectively. In the configuration of FIG. 7, each of the VNSIDs 400_1 and 400_2 in the virtual controller 300i maps to different partitions 150_1, 150_2 of a same physical namespace 120_1 in storage device 104_1, and the single VNSID 400_3 in virtual controller 300_2 maps to two different physical namespaces 120_2 and 120_3 in storage device 104_2. In this way, a write to the VNSID 400_3 in the second virtual controller 300_2 writes to two separate physical namespaces 120_2, 120_3.

Additional configurations are possible. For instance, the same defined virtual namespace identifier that maps to one physical namespace may be included

in two separate virtual controllers to allow for the sharing of a virtual namespace and the mapped physical namespace. Further, one virtual namespace can map to different physical namespaces or different partitions within a namespace in the same or different storage devices. A virtual namespace mapping to a physical namespace/partition may be included in multiple virtual controllers 300_i of one virtual subsystem to allow sharing of the virtual namespace by multiple hosts.

The host nodes $102_1 \ldots 102_n$ may address a virtual namespace, by including the virtual subsystem (VSS) name, the virtual controller (VC), and the virtual namespace identifier (VNSID) in a combined address, such as VSSname.VCname.VNSID. In this way, virtual namespace IDs in different virtual controllers may have the same number identifier but point to different physical namespaces/partitions. Alternatively, the same virtual namespace IDs in different virtual controllers may point to the same shared physical namespace/partition.

FIG. 8 illustrates an embodiment of operations performed by the virtual device layer 132 to generate the virtual subsystem 200, virtual controller 300, and virtual namespace 400 definitions. The virtual device layer 132 may generate the virtual definitions 200, 300, 400 in response to commands via a script program or a user interface to create these resources. In further embodiments, the virtual device layer 132 may generate the virtual definitions 200, 300, 400 in response to bits being set in a hardware register address space in the target system 106 or in hardware components implementing the virtual device layer 132 or in settings configured in the memory 136. Upon initiating (at block 800) an operation to configure virtual subsystems $200_1 \ldots 200_n$, the virtual device layer 132 configures (at block 802) in the memory device 136 a virtual subsystem 200_i providing a representation of a subsystem presented to at least one host node $102_1 \ldots 102_n$ indicated in assigned hosts 204 as assigned to the virtual subsystem 200_i. The virtual device layer 132 discovers (at block 804) physical namespaces $120_1 \ldots 120_t$ in the storage devices $104_1 \ldots 104_m$ and their address spaces. At least one virtual controller 300_i is configured (at block 806) in the memory 136 and indicated in one virtual subsystem definition 200_i. The virtual device layer 132 configures (at

block 808) at least one virtual namespace 400_i for a physical namespace $120_1....120_t$ and optionally a partition within the discovered physical namespace $120_1....120_t$ or the entire physical namespace. At least one of the virtual namespaces is indicated as assigned (at block 810) to at least one virtual controller 300_i by indicating the virtual namespace identifier (VNSID) 400_i in the virtual controller definition 300_i. The virtual device layer 132 further indicates (at block 812) at least one host node $102_1...102_n$ as assigned, such as in field, 204 of the virtual subsystem definition 200_i, to one of the virtual subsystem 200_i. For a host node $102_1...102_n$ assigned to the virtual subsystem 200_i, the virtual device layer 132 assigns (at block 814) a host 102_j to one or more virtual namespaces 400_i assigned to one or more virtual controllers 300_i in a virtual subsystem 200_i, such as by indicating the host 102_j in the assigned host 404 field of the virtual namespace definition 400_i to which the host 102_j is assigned. Once assigned (at block 814), the host 102_j may submit logical device interface protocol requests 504 to the virtual namespace identifier to access the assigned one or more physical namespace/partitions 406_i in the storage device 408_i.

With the operations of FIG. 8, the virtual device layer 132 constructs virtual representations in the memory 136 of the target system 106 of subsystems, controllers, and namespaces that are presented to the host nodes $102_1...102_n$ to access physical namespaces $120_1....120_t$ in the storage devices $104_1 ...104_m$.

FIGs. 9a and 9b illustrate an embodiment of operations performed by the virtual device layer 132 and other components of the target system 106, such as the fabric layer protocols 124 and bus interface protocol 138, to process a fabric packet 500 having a logic device interface protocol request 504, e.g., NVMe request, directed to a virtual namespace identifier (VNSID) 508. Upon receiving (at block 900) from a host node 102_i a fabric packet 500 including a read/write request 506 directed to a virtual namespace, e.g., VSSname.VCname.VNSID, a specific fabric protocol 126, 128, 130 in the fabric layer protocols 124 extracts (at block 902) the logic device interface protocol request 504, e.g., NVMe packet, from the fabric layer 502 of the fabric packet 500. If (at block 904) the host 102_i initiating the request, which may be indicated in the fabric layer 502 or request

504, is authorized to access the virtual subsystem 200_i specified in the target namespace 508 of the request, such as the host 102_i is indicated in the assigned hosts 204 of the virtual subsystem definition 200_i identified in the request, such as in the path of the target namespace 508, then a determination is made (at block 906) as to whether the requesting host 102_i is assigned to the requested virtual namespace (VNSID), such as indicated in host assigned field 404 of the requested virtual namespace 400_i in the virtual controller 300_i, which may be indicated in the target namespace request path 508. If (at block 904) the host 102_i does not have access to the requested virtual subsystem 200_i or does not have access to the requested virtual namespace 400_i in the target namespace path 508, then the request is denied and an access not authorized message is returned (at block 908) to the requesting host 102_i.

If (at block 904 and 906) the requesting host 102_i is authorized to access the virtual subsystem 200_i and the virtual namespace 400_i in the target namespace path 508, respectively, then the virtual device layer 132 determines (at block 910) the one or more physical namespaces $120_1 \ldots 120_t$/partitions 406_i to which the requested virtual namespace 400_i maps, as indicated in the virtual namespace definition 400_i. Control then proceeds to FIG. 9b where the virtual device layer 132 determines (at block 912) whether the target virtual namespace 508 maps to multiple different physical namespaces/partitions $406_1 \ldots 406_p$. If so, then the virtual device layer 132 performs the loop of operations at blocks 914-920 for each physical namespace/partition i of the multiple physical namespace/partitions $406_i \ldots 406_p$ to which the target virtual namespace maps The virtual device layer 132 generates (at block 916) a logic device interface protocol request 504 (NVMe request) including the determined physical namespace/partition 406_i as the target namespace 508, and the target addresses 510 from received the request 504 subject to the write request. The virtual device layer 132 uses (at block 918) a bus interface protocol 138 (e.g., PCIe) to transmit the logical device interface protocol request 504 to the storage device 408_i having the target physical namespace/partition 406_i.

If (at block 912) the target virtual namespace 508 maps to only one target physical namespace/partition 406_1, then the virtual device layer 132 modifies (at block 922) the logic device interface protocol request 504 (e.g., NVMe request) to include the determined physical namespace/partition 406_i as the target namespace 508 to replace the target virtual namespace inserted by the host node to produce a modified logical device interface protocol request 504, which is then transmitted (at block 924) using the bus interface protocol 138 to the storage device 104_i indicated in field 408_1 of the virtual namespace definition 400_i for the target VNSID.

With the described embodiments, a single logical device interface protocol request is transferred from the host system, over a fabric, to a target system, where the target virtual namespace in the request is replaced with the physical namespace to which the virtual namespace maps. This modified logical device interface protocol request is then transmitted to the storage device having the target namespace in the request using a bus protocol. With the described embodiments, the request does not need to be converted from one format to another, as the same request logical device protocol is used in the transmitted request. This is accomplished by generating virtual subsystems, virtual controllers, and virtual namespaces to present to the host nodes to communicate with the connected storage devices.

The described operations of the processing components, such as components 112a, 112b, 112c, 114, 124, 132, 138, and other components, may be implemented as a method, apparatus, device, computer product comprising a computer readable storage medium using standard programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof. The described operations may be implemented as code or logic maintained in a "computer readable storage medium". The term "code" as used herein refers to software program code, hardware logic, firmware, microcode, etc. The computer readable storage medium, as that term is used herein, includes a tangible element, including at least one of electronic circuitry, storage materials, inorganic materials, organic materials, biological materials, a casing, a housing, a

coating, and hardware. A computer readable storage medium may comprise, but is not limited to, a magnetic storage medium (e.g., hard disk drives, floppy disks, tape, etc.), optical storage (CD-ROMs, DVDs, optical disks, etc.), volatile and non-volatile memory devices (e.g., EEPROMs, ROMs, PROMs, RAMs, DRAMs, SRAMs, Flash Memory, firmware, programmable logic, etc.), Solid State Devices (SSD), computer encoded and readable punch cards, etc. The computer readable storage medium may further comprise a hardware device implementing firmware, microcode, etc., such as in an integrated circuit chip, a programmable logic device, a Programmable Gate Array (PGA), field-programmable gate array (FPGA), Application Specific Integrated Circuit (ASIC), etc. Still further, the code implementing the described operations may be implemented in "transmission signals", where transmission signals may propagate through space or through a transmission media, such as an optical fiber, copper wire, etc. The transmission signals in which the code or logic is encoded may further comprise a wireless signal, satellite transmission, radio waves, infrared signals, Bluetooth, etc. The program code embedded on a computer readable storage medium may be transmitted as transmission signals from a transmitting station or computer to a receiving station or computer. A computer readable storage medium is not comprised solely of transmission signals, but includes physical and tangible components. Those skilled in the art will recognize that many modifications may be made to this configuration without departing from the scope of the present invention, and that the article of manufacture may comprise suitable information bearing medium known in the art.

FIG. 10 illustrates an embodiment of a computer node architecture 1000, such as the components included in the host nodes 102_1, 102_2....102_n and the target system 106, including a processor 1002 that communicates over a bus 1004 with a volatile memory device 1006 in which programs, operands and parameters being executed are cached, and a non-volatile storage device 1004, such as target system memory 136. The bus 1004 may comprise multiple buses. Further, the bus 1004 may comprise a multi-agent bus or not be a multi-agent bus, and instead provide point-to-point connections according to PCIe architecture. The processor

1002 may also communicate with Input/output (I/O) devices 1012a, 1012b, which may comprise input devices, display devices, graphics cards, ports, network interfaces, etc. For the host nodes 102_1, 102_2....102_n, the virtual subsystems may also appear on the bus 1004 as bus components.

In certain embodiments, the computer node architecture 1000 may comprise a personal computer, server, mobile device or embedded compute device. In a silicon-on-chip (SOC) implementation, the architecture 1000 may be implemented in an integrated circuit die. In certain implementations, the architecture 1000 may not include a PCIe bus to connect to NVMe storage devices, and instead include a network adaptor to connect to a fabric or network and send communications using the NVMe interface to communicate with the target system 106 to access underlying storage devices 104_1 ...104_m.

The reference characters used herein, such as *i, j, m, n, p,* and *t* are used to denote a variable number of instances of an element, which may represent the same or different values, and may represent the same or different value when used with different or the same elements in different described instances.

The terms "an embodiment", "embodiment", "embodiments", "the embodiment", "the embodiments", "one or more embodiments", "some embodiments", and "one embodiment" mean "one or more (but not all) embodiments of the present invention(s)" unless expressly specified otherwise.

The terms "including", "comprising", "having" and variations thereof mean "including but not limited to", unless expressly specified otherwise.

The enumerated listing of items does not imply that any or all of the items are mutually exclusive, unless expressly specified otherwise.

The terms "a", "an" and "the" mean "one or more", unless expressly specified otherwise.

Devices that are in communication with each other need not be in continuous communication with each other, unless expressly specified otherwise. In addition, devices that are in communication with each other may communicate directly or indirectly through one or more intermediaries.

A description of an embodiment with several components in communication with each other does not imply that all such components are required. On the contrary a variety of optional components are described to illustrate the wide variety of possible embodiments of the present invention.

When a single device or article is described herein, it will be readily apparent that more than one device/article (whether or not they cooperate) may be used in place of a single device/article. Similarly, where more than one device or article is described herein (whether or not they cooperate), it will be readily apparent that a single device/article may be used in place of the more than one device or article or a different number of devices/articles may be used instead of the shown number of devices or programs. The functionality and/or the features of a device may be alternatively embodied by one or more other devices which are not explicitly described as having such functionality/features. Thus, other embodiments of the present invention need not include the device itself.

The foregoing description of various embodiments of the invention has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the invention 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 invention be limited not by this detailed description, but rather by the claims appended hereto. The above specification, examples and data provide a complete description of the manufacture and use of the composition of the invention. Since many embodiments of the invention can be made without departing from the spirit and scope of the invention, the invention resides in the claims herein after appended.

EXAMPLES

Example 1 is a computer product including a computer readable storage media deployed in a target system, wherein the target system includes a memory device and communicates with at least one host system and at least one storage device, wherein the computer readable storage media includes code that is executed to: configure, in the memory device, a virtual subsystem to provide a

representation of a subsystem presented to the at least one host system, wherein the at least one host system is to communicate with the virtual subsystem to access the at least one storage device; and configure, in the memory device, at least one virtual namespace for the virtual subsystem, wherein each of the at least one virtual namespace maps to at least one physical namespace in at least one storage device, wherein each virtual namespace is assigned to one host system to use to direct read and write requests to the at least one physical namespace of the at least one storage device assigned to the virtual namespace.

In Example 2, the subject matter of Examples 1 and 3-11 can optionally include that one virtual namespace maps to at least two physical namespaces in at least one of the storage devices.

In Example 3, the subject matter of Examples 1, 2 and 4-11 can optionally include that at least two virtual namespaces map to different ranges of addresses within one physical namespace in one of the at least one storage device.

In Example 4, the subject matter of Examples 1-3 and 5-11 can optionally include that two virtual namespaces in the at least one storage device are assigned to different host systems.

In Example 5, the subject matter of Examples 1-4 and 6-11 can optionally include that the code is further executed to: configure, in the memory device, at least one virtual controller in the virtual subsystem, wherein each of the at least one virtual controller is assigned at least virtual namespace.

In Example 6, the subject matter of Examples 1-5 and 7-11 can optionally include that one virtual controller is assigned a first virtual namespace assigned to a first host system and a second virtual namespace assigned to a second host system, wherein the first and second virtual namespaces map to first and second ranges of addresses, respectively, in one of the storage devices.

In Example 7, the subject matter of Examples 1-6 and 8-11 can optionally include that at least two of the virtual controllers are assigned a same virtual namespace assigned to one physical namespace to provide shared access to the physical namespace assigned to the virtual namespace.

In Example 8, the subject matter of Examples 1-7 and 9-11 can optionally include that the code is further executed to: receive from the host system a read or write request directed to a target virtual namespace; determine the physical namespace and a storage device including the physical namespace to which the target virtual namespace maps; and send to the determined storage device the received read or write request directed to the determined physical namespace.

In Example 9, the subject matter of Examples 1-8 and 10-11 can optionally include that the code is further executed to: receive from the host system a read or write request directed to a target virtual namespace that maps to a plurality of physical namespaces; and send the received read or write request directed to at least one storage device having the plurality of physical namespaces to which the target virtual namespace maps.

In Example 10, the subject matter of Examples 1-9 and 11 can optionally include that the target system, the at least one host system, and the at least one storage device implement a logical device interface protocol to communicate read and write requests between the at least one host system and the at least one storage device, wherein the target system communicates with the at least one host system using a fabric protocol and wherein the targets system and the at least one storage device communicate using a bus interface protocol, wherein the code is further executed to: receive from the host system a packet in the fabric protocol encapsulating a read or write request in the logical device interface protocol directed to a target virtual namespace; determine the physical namespace and the storage device including the physical namespace to which the target virtual namespace maps; generate a modified read or write request in the logical device interface protocol directed to the determined physical namespace to perform the received read or write request; and send to the determined storage device, using the bus interface protocol, the modified read or write request directed to the determined physical namespace.

In Example 11, the subject matter of Examples 1-10 can optionally include that the logical device interface protocol comprises a Non-volatile

Memory Express (NVMe) protocol and wherein the bus interface protocol comprises a Peripheral Component Interconnect Express (PCIe) protocol.

Example 12 is a system for dynamically managing at least one storage device accessed by at least one host system over a network, comprising: a processor to process received read and write requests directed to the at least one storage device; a memory device; a computer readable storage medium including code that is executed by the processor to: configure, in the memory device, a virtual subsystem to provide a representation of a subsystem presented to the at least one host system, wherein the at least one host system is to communicate with the virtual subsystem to access the at least one storage device; and configure, in the memory device, at least one virtual namespace for the virtual subsystem, wherein each of the at least one virtual namespace maps to at least one physical namespace in at least one storage device, wherein each virtual namespace is assigned to one host system to use to direct read and write requests to the at least one physical namespace of the at least one storage device assigned to the virtual namespace.

In Example 13, the subject matter of Examples 12 and 14-18 can optionally include that one virtual namespace maps to at least two physical namespaces in at least one of the storage devices.

In Example 14, the subject matter of Examples 12, 13 and 15-18 can optionally include that the code is further executed to: configure, in the memory device, at least one virtual controller in the virtual subsystem, wherein each of the at least one virtual controller is assigned at least virtual namespace.

In Example 15, the subject matter of Examples 12-14 and 16-18 can optionally include that one virtual controller is assigned a first virtual namespace assigned to a first host system and a second virtual namespace assigned to a second host system, wherein the first and second virtual namespaces map to first and second ranges of addresses, respectively, in one of the storage devices.

In Example 16, the subject matter of Examples 12-15, 17, and 18 can optionally include that the code is further executed to: receive from the host system a read or write request directed to a target virtual namespace; determine

the physical namespace and a storage device including the physical namespace to which the target virtual namespace maps; and send to the determined storage device the received read or write request directed to the determined physical namespace.

In Example 17, the subject matter of Examples 12-16 and 18 can optionally include that the code is further executed to: receive from the host system a read or write request directed to a target virtual namespace that maps to a plurality of physical namespaces; and send the received read or write request directed to at least one storage device having the plurality of physical namespaces to which the target virtual namespace maps.

In Example 18, the subject matter of Examples 12-17 can optionally include that the target system, the at least one host system, and the at least one storage device implement a logical device interface protocol to communicate read and write requests between the at least one host system and the at least one storage device, wherein the target system communicates with the at least one host system using a fabric protocol and wherein the targets system and the at least one storage device communicate using a bus interface protocol, wherein the code is further executed to: receive from the host system a packet in the fabric protocol encapsulating a read or write request in the logical device interface protocol directed to a target virtual namespace; determine the physical namespace and the storage device including the physical namespace to which the target virtual namespace maps; generate a modified read or write request in the logical device interface protocol directed to the determined physical namespace to perform the received read or write request; and send to the determined storage device, using the bus interface protocol, the modified read or write request directed to the determined physical namespace.

Example 19 is a method for dynamically managing at least one storage device accessed by at least one host system over a network, comprising: configuring, in a memory device, a virtual subsystem to provide a representation of a subsystem presented to at least one host system, wherein the at least one host system is to communicate with the virtual subsystem to access at least one storage

device; and configuring, in the memory device, at least one virtual namespace for the virtual subsystem, wherein each of the at least one virtual namespace maps to at least one physical namespace in at least one storage device, wherein each virtual namespace is assigned to one host system to use to direct read and write requests to the at least one physical namespace of the at least one storage device assigned to the virtual namespace.

In Example 20, the subject matter of claims 19 and 21-25 can optionally include that one virtual namespace maps to at least two physical namespaces in at least one of the storage devices.

In Example 21, the subject matter of claims 19, 20 and 22-25 can optionally include configuring, in the memory device, at least one virtual controller in the virtual subsystem, wherein each of the at least one virtual controller is assigned at least virtual namespace.

In Example 22, the subject matter of claims 19-21 and 23-25 can optionally include that one virtual controller is assigned a first virtual namespace assigned to a first host system and a second virtual namespace assigned to a second host system, wherein the first and second virtual namespaces map to first and second ranges of addresses, respectively, in one of the storage devices.

In Example 23, the subject matter of claims 19-22 and 24-25 can optionally include receiving from the host system a read or write request directed to a target virtual namespace; determining the physical namespace and a storage device including the physical namespace to which the target virtual namespace maps; and sending to the determined storage device the received read or write request directed to the determined physical namespace.

In Example 24, the subject matter of claims 19-23 and 25 can optionally include receiving from the host system a read or write request directed to a target virtual namespace that maps to a plurality of physical namespaces; and sending the received read or write request directed to at least one storage device having the plurality of physical namespaces to which the target virtual namespace maps.

In Example 25, the subject matter of claims 19-24 can optionally include that the target system, the at least one host system, and the at least one storage

device implement a logical device interface protocol to communicate read and write requests between the at least one host system and the at least one storage device, wherein the target system communicates with the at least one host system using a fabric protocol and wherein the targets system and the at least one storage device communicate using a bus interface protocol, further comprising: receiving from the host system a packet in the fabric protocol encapsulating a read or write request in the logical device interface protocol directed to a target virtual namespace; determining the physical namespace and the storage device including the physical namespace to which the target virtual namespace maps; generating a modified read or write request in the logical device interface protocol directed to the determined physical namespace to perform the received read or write request; and sending to the determined storage device, using the bus interface protocol, the modified read or write request directed to the determined physical namespace.

In Example 26, the subject matter of claims 19 can optionally include the at least any one of:

(1) wherein one virtual namespace maps to at least two physical namespaces in at least one of the storage devices; and/or

(2) wherein at least two virtual namespaces map to different ranges of addresses within one physical namespace in one of the at least one storage device; and/or

(3) wherein two virtual namespaces in the at least one storage device are assigned to different host systems; and/or

(4) configuring, in the memory device, at least one virtual controller in the virtual subsystem, wherein each of the at least one virtual controller is assigned at least virtual namespace; and/or

(5) wherein one virtual controller is assigned a first virtual namespace assigned to a first host system and a second virtual namespace assigned to a second host system, wherein the first and second virtual namespaces map to first and second ranges of addresses, respectively, in one of the storage devices; and/or.

(6) wherein at least two of the virtual controllers are assigned a same virtual namespace assigned to one physical namespace to provide shared access to the physical namespace assigned to the virtual namespace; and/or

(7) receiving from the host system a read or write request directed to a target virtual namespace; determining the physical namespace and a storage device including the physical namespace to which the target virtual namespace maps; and sending to the determined storage device the received read or write request directed to the determined physical namespace; and/or

(8) receiving from the host system a read or write request directed to a target virtual namespace that maps to a plurality of physical namespaces; and sending the received read or write request directed to at least one storage device having the plurality of physical namespaces to which the target virtual namespace maps; and/or

(9) wherein the target system, the at least one host system, and the at least one storage device implement a logical device interface protocol to communicate read and write requests between the at least one host system and the at least one storage device, wherein the target system communicates with the at least one host system using a fabric protocol and wherein the targets system and the at least one storage device communicate using a bus interface protocol, further comprising: receiving from the host system a packet in the fabric protocol encapsulating a read or write request in the logical device interface protocol directed to a target virtual namespace; determining the physical namespace and the storage device including the physical namespace to which the target virtual namespace maps; generating a modified read or write request in the logical device interface protocol directed to the determined physical namespace to perform the received read or write request; and sending to the determined storage device, using the bus interface protocol, the modified read or write request directed to the determined physical namespace; and/or

(10) wherein the logical device interface protocol comprises a Non-volatile Memory Express (NVMe) protocol and wherein the bus interface protocol comprises a Peripheral Component Interconnect Express (PCIe) protocol.

Example 27 is an apparatus for dynamically managing at least one storage device accessed by at least one host system over a network, having computer program code executed to: configure, in the memory device, a virtual subsystem to provide a representation of a subsystem presented to the at least one host system, wherein the at least one host system is to communicate with the virtual subsystem to access the at least one storage device; and configure, in the memory device, at least one virtual namespace for the virtual subsystem, wherein each of the at least one virtual namespace maps to at least one physical namespace in at least one storage device, wherein each virtual namespace is assigned to one host system to use to direct read and write requests to the at least one physical namespace of the at least one storage device assigned to the virtual namespace.

In Example 28, the subject matter of claims 27 and 29-37 can optionally include that one virtual namespace maps to at least two physical namespaces in at least one of the storage devices.

In Example 29, the subject matter of claims 27, 28 and 30-37 can optionally include that at least two virtual namespaces map to different ranges of addresses within one physical namespace in one of the at least one storage device.

In Example 30, the subject matter of claims 27-29 and 31-37 can optionally include that two virtual namespaces in the at least one storage device are assigned to different host systems.

In Example 31, the subject matter of claims 27-30 and 32-37 can optionally include configure, in the memory device, at least one virtual controller in the virtual subsystem, wherein each of the at least one virtual controller is assigned at least virtual namespace.

In Example 32, the subject matter of claims 27-31 and 33-37 can optionally include that one virtual controller is assigned a first virtual namespace assigned to a first host system and a second virtual namespace assigned to a second host system, wherein the first and second virtual namespaces map to first and second ranges of addresses, respectively, in one of the storage devices.

In Example 33, the subject matter of claims 27-32 and 34-36 can optionally include that at least two of the virtual controllers are assigned a same

virtual namespace assigned to one physical namespace to provide shared access to the physical namespace assigned to the virtual namespace.

In Example 34, the subject matter of claims 27-34 and 35-36 can optionally receive from the host system a read or write request directed to a target virtual namespace; determine the physical namespace and a storage device including the physical namespace to which the target virtual namespace maps; and send to the determined storage device the received read or write request directed to the determined physical namespace.

In Example 35, the subject matter of claims 27-34 and 36-37 can optionally include receive from the host system a read or write request directed to a target virtual namespace that maps to a plurality of physical namespaces; and send the received read or write request directed to at least one storage device having the plurality of physical namespaces to which the target virtual namespace maps.

In Example 36, the subject matter of claims 27-35 and 37 can optionally include that the target system, the at least one host system, and the at least one storage device implement a logical device interface protocol to communicate read and write requests between the at least one host system and the at least one storage device, wherein the target system communicates with the at least one host system using a fabric protocol and wherein the targets system and the at least one storage device communicate using a bus interface protocol, wherein the code is further executed to: receive from the host system a packet in the fabric protocol encapsulating a read or write request in the logical device interface protocol directed to a target virtual namespace; determine the physical namespace and the storage device including the physical namespace to which the target virtual namespace maps; generate a modified read or write request in the logical device interface protocol directed to the determined physical namespace to perform the received read or write request; and send to the determined storage device, using the bus interface protocol, the modified read or write request directed to the determined physical namespace.

In Example 37, the subject matter of claims 27-35 can optionally include that the logical device interface protocol comprises a Non-volatile Memory Express (NVMe) protocol and wherein the bus interface protocol comprises a Peripheral Component Interconnect Express (PCIe) protocol.

Example 38 is an apparatus for dynamically managing at least one storage device accessed by at least one host system over a network, comprising: means for configuring, in a memory device, a virtual subsystem to provide a representation of a subsystem presented to at least one host system, wherein the at least one host system is to communicate with the virtual subsystem to access at least one storage device; and means for configuring, in the memory device, at least one virtual namespace for the virtual subsystem, wherein each of the at least one virtual namespace maps to at least one physical namespace in at least one storage device, wherein each virtual namespace is assigned to one host system to use to direct read and write requests to the at least one physical namespace of the at least one storage device assigned to the virtual namespace.

Example 39 is an apparatus comprising means to perform a method as claimed in any preceding claim.

Example 40 is a machine-readable storage including machine-readable instructions, when executed, to implement a method or realize an apparatus as claimed in any preceding claim.

WHAT IS CLAIMED

1. A computer product including a computer readable storage media deployed in a target system, wherein the target system includes a memory device and communicates with at least one host system and at least one storage device, wherein the computer readable storage media includes code that is executed to:

configure, in the memory device, a virtual subsystem to provide a representation of a subsystem presented to the at least one host system, wherein the at least one host system is to communicate with the virtual subsystem to access the at least one storage device; and

configure, in the memory device, at least one virtual namespace for the virtual subsystem, wherein each of the at least one virtual namespace maps to at least one physical namespace in at least one storage device, wherein each virtual namespace is assigned to one host system to use to direct read and write requests to the at least one physical namespace of the at least one storage device assigned to the virtual namespace.

2. The computer product of claim 1, wherein one virtual namespace maps to at least two physical namespaces in at least one of the storage devices.

3. The computer product of claim 1, wherein at least two virtual namespaces map to different ranges of addresses within one physical namespace in one of the at least one storage device.

4. The computer product of claim 1, wherein two virtual namespaces in the at least one storage device are assigned to different host systems.

5. The computer product of claim 1, wherein the code is further executed to:

configure, in the memory device, at least one virtual controller in the virtual subsystem, wherein each of the at least one virtual controller is assigned at least virtual namespace.

6. The computer product of claim 5, wherein one virtual controller is assigned a first virtual namespace assigned to a first host system and a second virtual namespace assigned to a second host system, wherein the first and second virtual namespaces map to first and second ranges of addresses, respectively, in one of the storage devices.

7. The computer product of claim 5, wherein at least two of the virtual controllers are assigned a same virtual namespace assigned to one physical namespace to provide shared access to the physical namespace assigned to the virtual namespace.

8. The computer product of any one of claims 1-7, wherein the code is further executed to:

receive from the host system a read or write request directed to a target virtual namespace;

determine the physical namespace and a storage device including the physical namespace to which the target virtual namespace maps; and

send to the determined storage device the received read or write request directed to the determined physical namespace.

9. The computer product of any one of claims 1-7, wherein the code is further executed to:

receive from the host system a read or write request directed to a target virtual namespace that maps to a plurality of physical namespaces; and

send the received read or write request directed to at least one storage device having the plurality of physical namespaces to which the target virtual namespace maps.

10. The computer product of any one of claims 1-7, wherein the target system, the at least one host system, and the at least one storage device implement a logical device interface protocol to communicate read and write requests

between the at least one host system and the at least one storage device, wherein the target system communicates with the at least one host system using a fabric protocol and wherein the targets system and the at least one storage device communicate using a bus interface protocol, wherein the code is further executed to:

receive from the host system a packet in the fabric protocol encapsulating a read or write request in the logical device interface protocol directed to a target virtual namespace;

determine the physical namespace and the storage device including the physical namespace to which the target virtual namespace maps;

generate a modified read or write request in the logical device interface protocol directed to the determined physical namespace to perform the received read or write request; and

send to the determined storage device, using the bus interface protocol, the modified read or write request directed to the determined physical namespace.

11. The computer product of claim 10, wherein the logical device interface protocol comprises a Non-volatile Memory Express (NVMe) protocol and wherein the bus interface protocol comprises a Peripheral Component Interconnect Express (PCIe) protocol.

12. A system for dynamically managing at least one storage device accessed by at least one host system over a network, comprising:

a processor to process received read and write requests directed to the at least one storage device;

a memory device;

a computer readable storage medium including code that is executed by the processor to:

configure, in the memory device, a virtual subsystem to provide a representation of a subsystem presented to the at least one host system, wherein

the at least one host system is to communicate with the virtual subsystem to access the at least one storage device; and

configure, in the memory device, at least one virtual namespace for the virtual subsystem, wherein each of the at least one virtual namespace maps to at least one physical namespace in at least one storage device, wherein each virtual namespace is assigned to one host system to use to direct read and write requests to the at least one physical namespace of the at least one storage device assigned to the virtual namespace.

13. The system of claim 12, wherein one virtual namespace maps to at least two physical namespaces in at least one of the storage devices.

14. The system of claim 12, wherein the code is further executed to:

configure, in the memory device, at least one virtual controller in the virtual subsystem, wherein each of the at least one virtual controller is assigned at least virtual namespace.

15. The system of claim 14, wherein one virtual controller is assigned a first virtual namespace assigned to a first host system and a second virtual namespace assigned to a second host system, wherein the first and second virtual namespaces map to first and second ranges of addresses, respectively, in one of the storage devices.

16. The system of any one of claims 12-15, wherein the code is further executed to:

receive from the host system a read or write request directed to a target virtual namespace;

determine the physical namespace and a storage device including the physical namespace to which the target virtual namespace maps; and

send to the determined storage device the received read or write request directed to the determined physical namespace.

17. The system of any one of claims 12-15, wherein the code is further executed to:

receive from the host system a read or write request directed to a target virtual namespace that maps to a plurality of physical namespaces; and

send the received read or write request directed to at least one storage device having the plurality of physical namespaces to which the target virtual namespace maps.

18. The system of any one of claims 12-15, wherein the target system, the at least one host system, and the at least one storage device implement a logical device interface protocol to communicate read and write requests between the at least one host system and the at least one storage device, wherein the target system communicates with the at least one host system using a fabric protocol and wherein the targets system and the at least one storage device communicate using a bus interface protocol, wherein the code is further executed to:

receive from the host system a packet in the fabric protocol encapsulating a read or write request in the logical device interface protocol directed to a target virtual namespace;

determine the physical namespace and the storage device including the physical namespace to which the target virtual namespace maps;

generate a modified read or write request in the logical device interface protocol directed to the determined physical namespace to perform the received read or write request; and

send to the determined storage device, using the bus interface protocol, the modified read or write request directed to the determined physical namespace.

19. A method for dynamically managing at least one storage device accessed by at least one host system over a network, comprising:

configuring, in a memory device, a virtual subsystem to provide a representation of a subsystem presented to at least one host system, wherein the at

least one host system is to communicate with the virtual subsystem to access at least one storage device; and

configuring, in the memory device, at least one virtual namespace for the virtual subsystem, wherein each of the at least one virtual namespace maps to at least one physical namespace in at least one storage device, wherein each virtual namespace is assigned to one host system to use to direct read and write requests to the at least one physical namespace of the at least one storage device assigned to the virtual namespace.

20. The method of claim 19, wherein one virtual namespace maps to at least two physical namespaces in at least one of the storage devices.

21. The method of claim 19, further comprising:

configuring, in the memory device, at least one virtual controller in the virtual subsystem, wherein each of the at least one virtual controller is assigned at least virtual namespace.

22. The method of claim 21, wherein one virtual controller is assigned a first virtual namespace assigned to a first host system and a second virtual namespace assigned to a second host system, wherein the first and second virtual namespaces map to first and second ranges of addresses, respectively, in one of the storage devices.

23. The method of any one of claims 19-23, further comprising:

receiving from the host system a read or write request directed to a target virtual namespace;

determining the physical namespace and a storage device including the physical namespace to which the target virtual namespace maps; and

sending to the determined storage device the received read or write request directed to the determined physical namespace.

24. The method any one of claims 19-23, further comprising:

receiving from the host system a read or write request directed to a target virtual namespace that maps to a plurality of physical namespaces; and

sending the received read or write request directed to at least one storage device having the plurality of physical namespaces to which the target virtual namespace maps.

25. An apparatus for dynamically managing at least one storage device accessed by at least one host system over a network, comprising:

means for configuring, in a memory device, a virtual subsystem to provide a representation of a subsystem presented to at least one host system, wherein the at least one host system is to communicate with the virtual subsystem to access at least one storage device; and

means for configuring, in the memory device, at least one virtual namespace for the virtual subsystem, wherein each of the at least one virtual namespace maps to at least one physical namespace in at least one storage device, wherein each virtual namespace is assigned to one host system to use to direct read and write requests to the at least one physical namespace of the at least one storage device assigned to the virtual namespace.

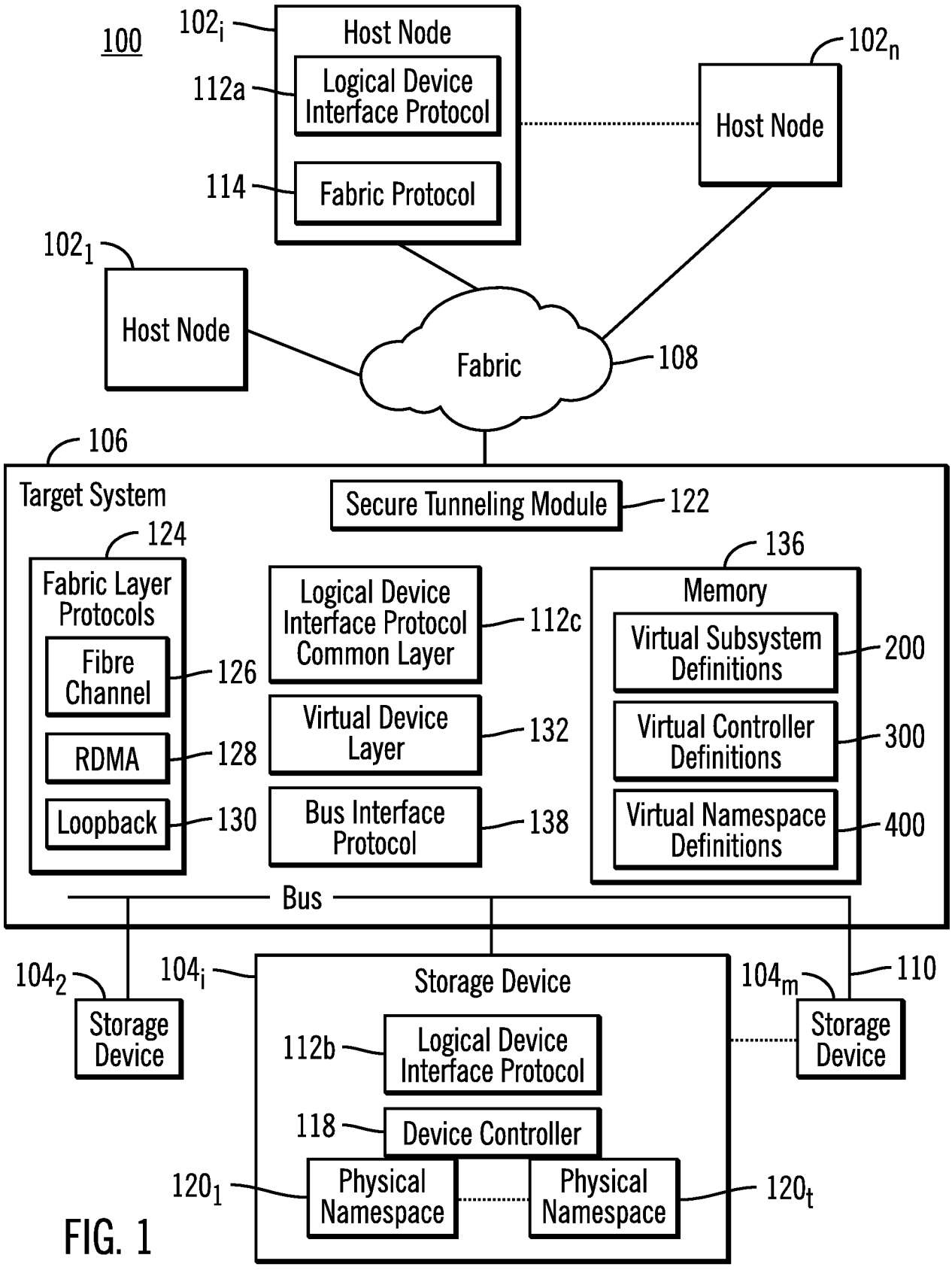

100
102i
Host Node
112a
Logical Device Interface Protocol
114
Fabric Protocol
102n
Host Node
102₁
Host Node
Fabric
108
106
Target System
Secure Tunneling Module
122
124
Fabric Layer Protocols
Fibre Channel
126
RDMA
128
Loopback
130
Logical Device Interface Protocol Common Layer
112c
Virtual Device Layer
132
Bus Interface Protocol
138
136
Memory
Virtual Subsystem Definitions
200
Virtual Controller Definitions
300
Virtual Namespace Definitions
400
Bus
104₂
Storage Device
104i
Storage Device
112b
Logical Device Interface Protocol
118
Device Controller
120₁
Physical Namespace
Physical Namespace
120t
104m
Storage Device
110

FIG. 1

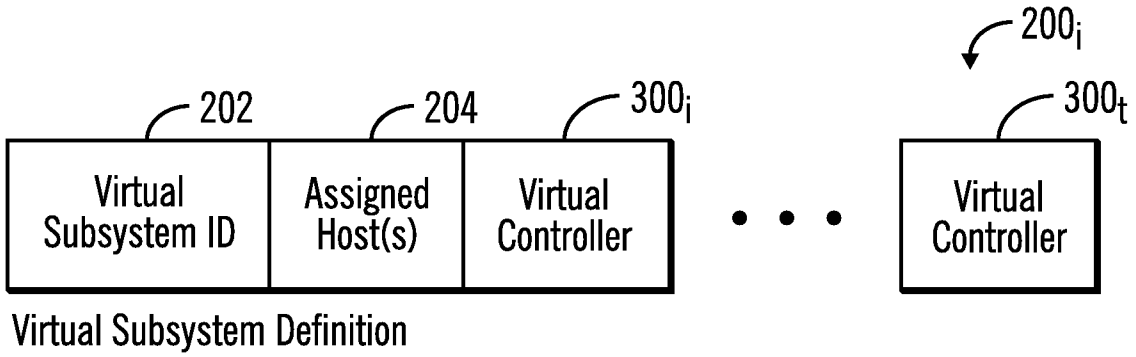


FIG. 2

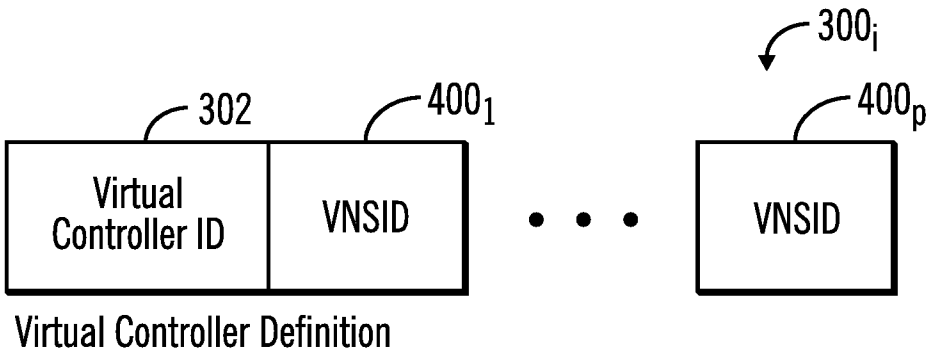


FIG. 3

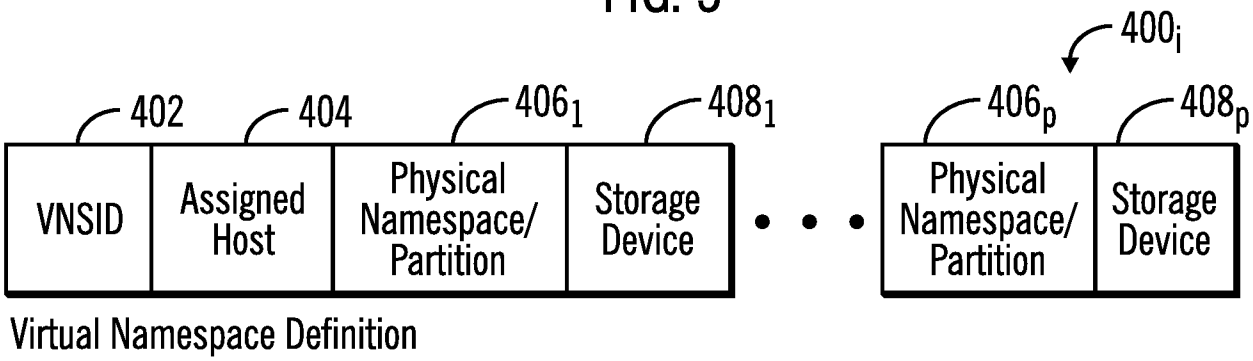


FIG. 4

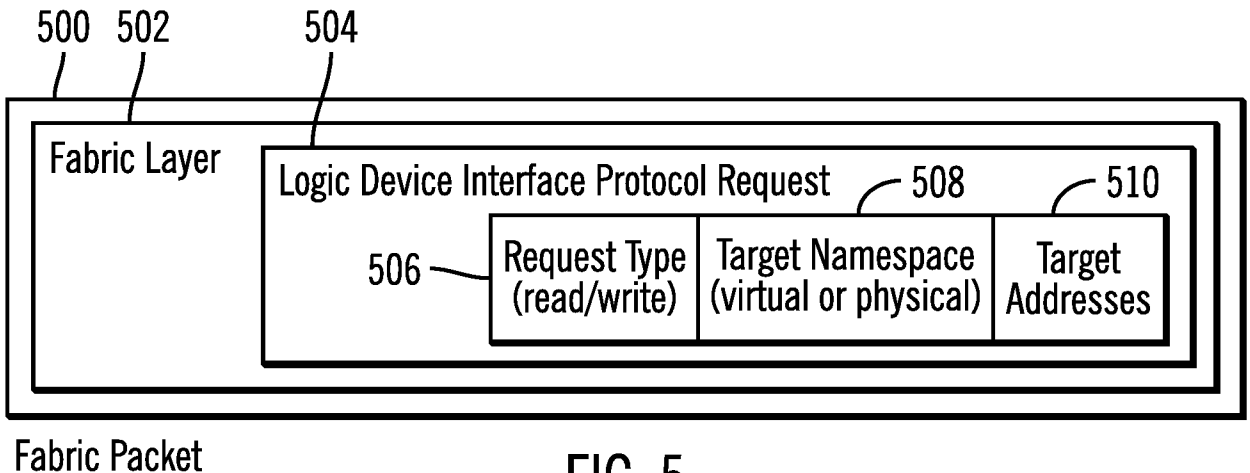


FIG. 5

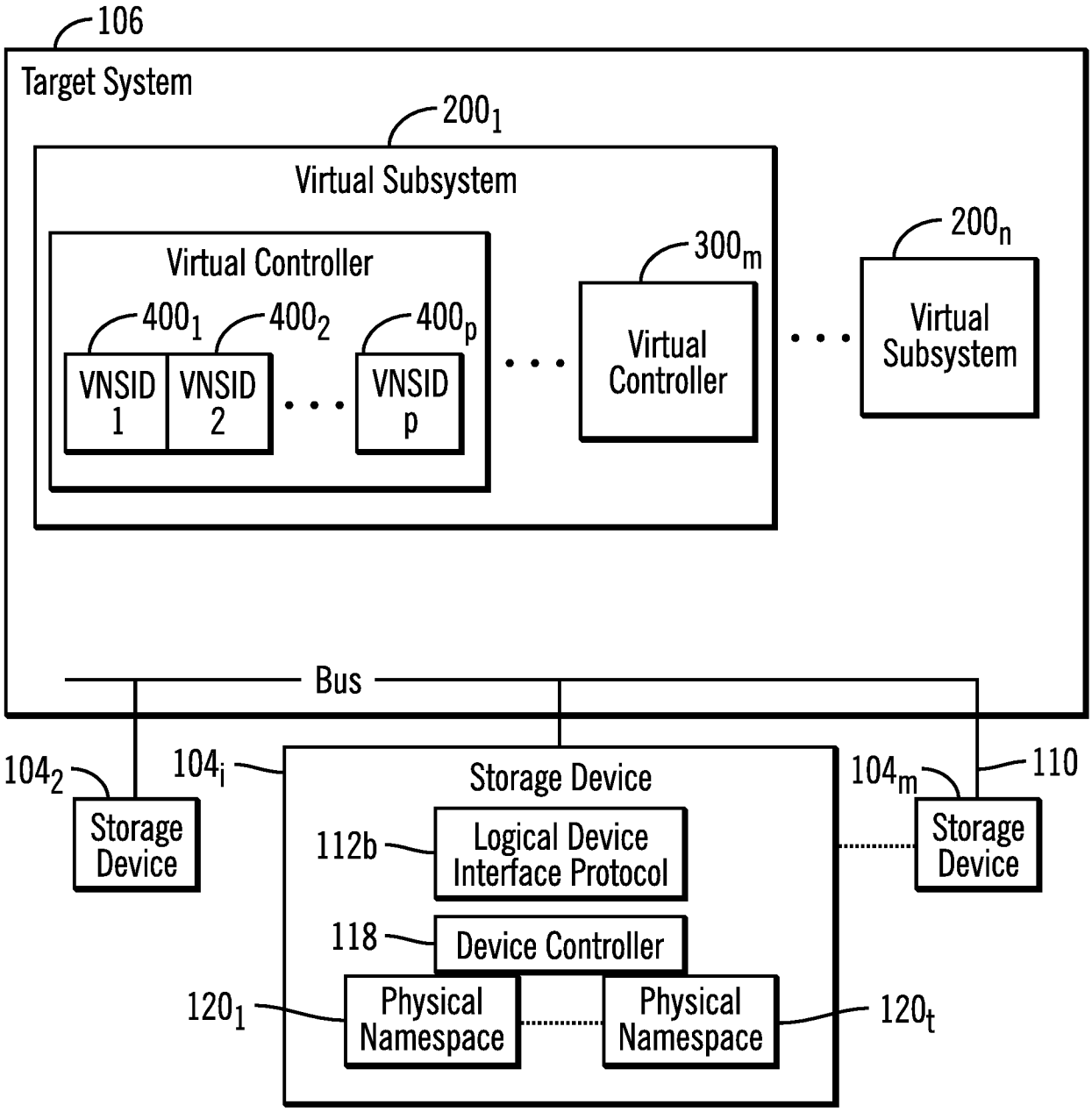

106
Target System
200$_1$
Virtual Subsystem
Virtual Controller
400$_1$
400$_2$
400$_p$
VNSID 1
VNSID 2
VNSID p
300$_m$
Virtual Controller
200$_n$
Virtual Subsystem
Bus
104$_2$
Storage Device
104$_i$
Storage Device
112b
Logical Device Interface Protocol
118
Device Controller
120$_1$
Physical Namespace
Physical Namespace
120$_t$
104$_m$
Storage Device
110

FIG. 6

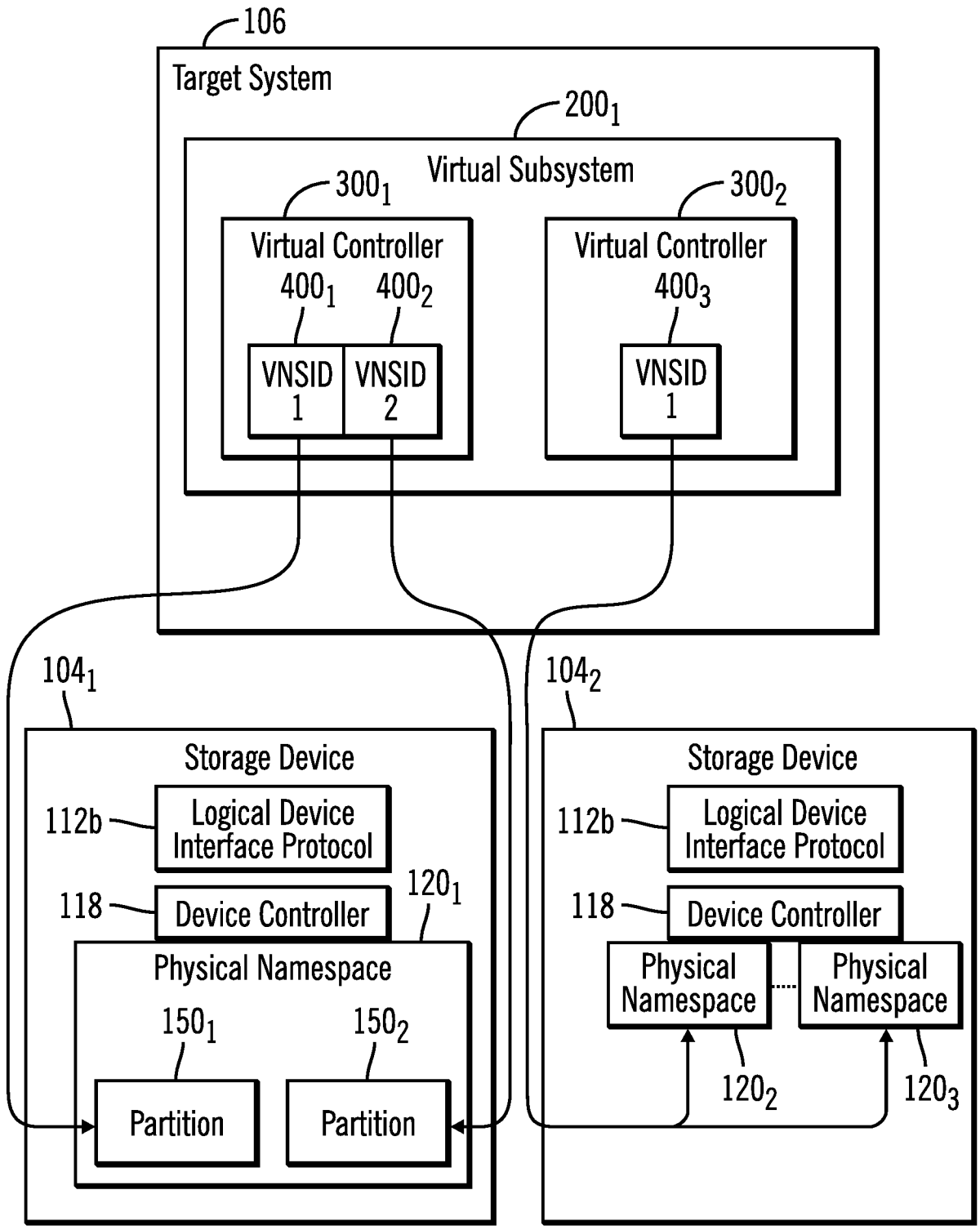

106
Target System
200_1
Virtual Subsystem
300_1
300_2
Virtual Controller
400_1
400_2
VNSID 1
VNSID 2
Virtual Controller
400_3
VNSID 1
104_1
104_2
Storage Device
112b
Logical Device Interface Protocol
118
Device Controller
120_1
Physical Namespace
150_1
150_2
Partition
Partition
Storage Device
112b
Logical Device Interface Protocol
118
Device Controller
Physical Namespace
Physical Namespace
120_2
120_3

FIG. 7

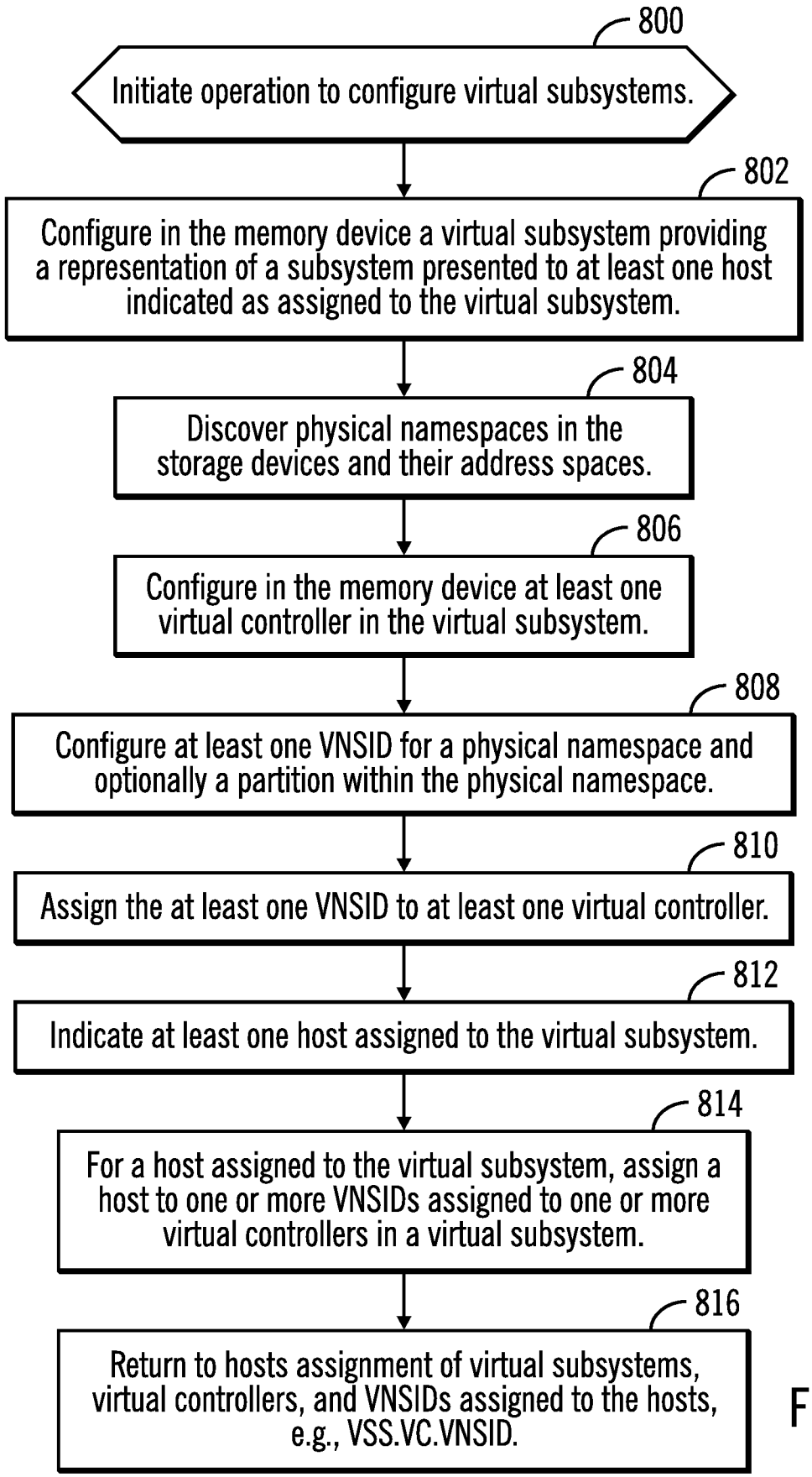

800
Initiate operation to configure virtual subsystems.
802
Configure in the memory device a virtual subsystem providing a representation of a subsystem presented to at least one host indicated as assigned to the virtual subsystem.
804
Discover physical namespaces in the storage devices and their address spaces.
806
Configure in the memory device at least one virtual controller in the virtual subsystem.
808
Configure at least one VNSID for a physical namespace and optionally a partition within the physical namespace.
810
Assign the at least one VNSID to at least one virtual controller.
812
Indicate at least one host assigned to the virtual subsystem.
814
For a host assigned to the virtual subsystem, assign a host to one or more VNSIDs assigned to one or more virtual controllers in a virtual subsystem.
816
Return to hosts assignment of virtual subsystems, virtual controllers, and VNSIDs assigned to the hosts, e.g., VSS.VC.VNSID.

FIG. 8

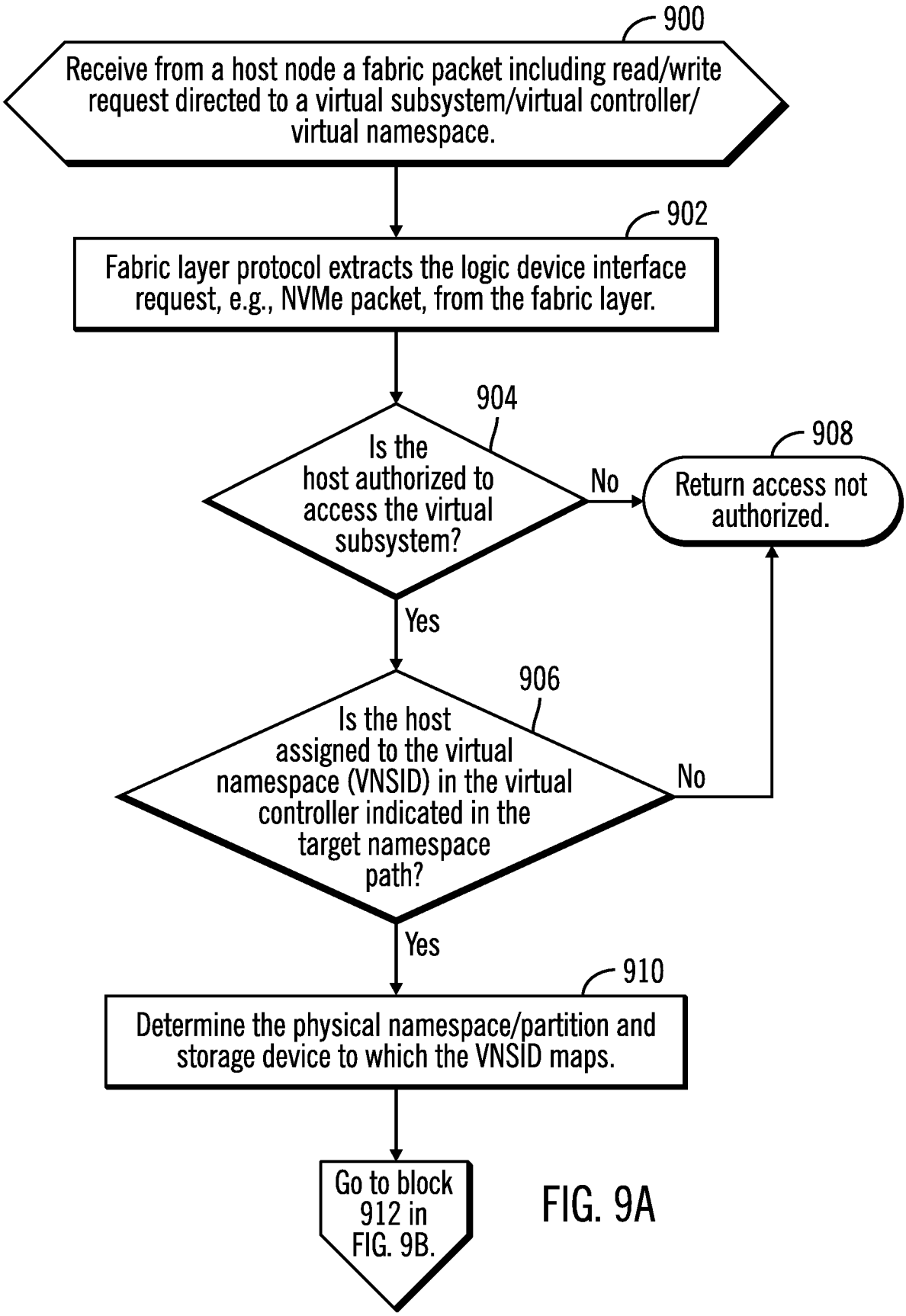

900
Receive from a host node a fabric packet including read/write request directed to a virtual subsystem/virtual controller/ virtual namespace.
902
Fabric layer protocol extracts the logic device interface request, e.g., NVMe packet, from the fabric layer.
904
Is the host authorized to access the virtual subsystem?
No
908
Return access not authorized.
Yes
906
Is the host assigned to the virtual namespace (VNSID) in the virtual controller indicated in the target namespace path?
No
Yes
910
Determine the physical namespace/partition and storage device to which the VNSID maps.
Go to block 912 in FIG. 9B.

FIG. 9A

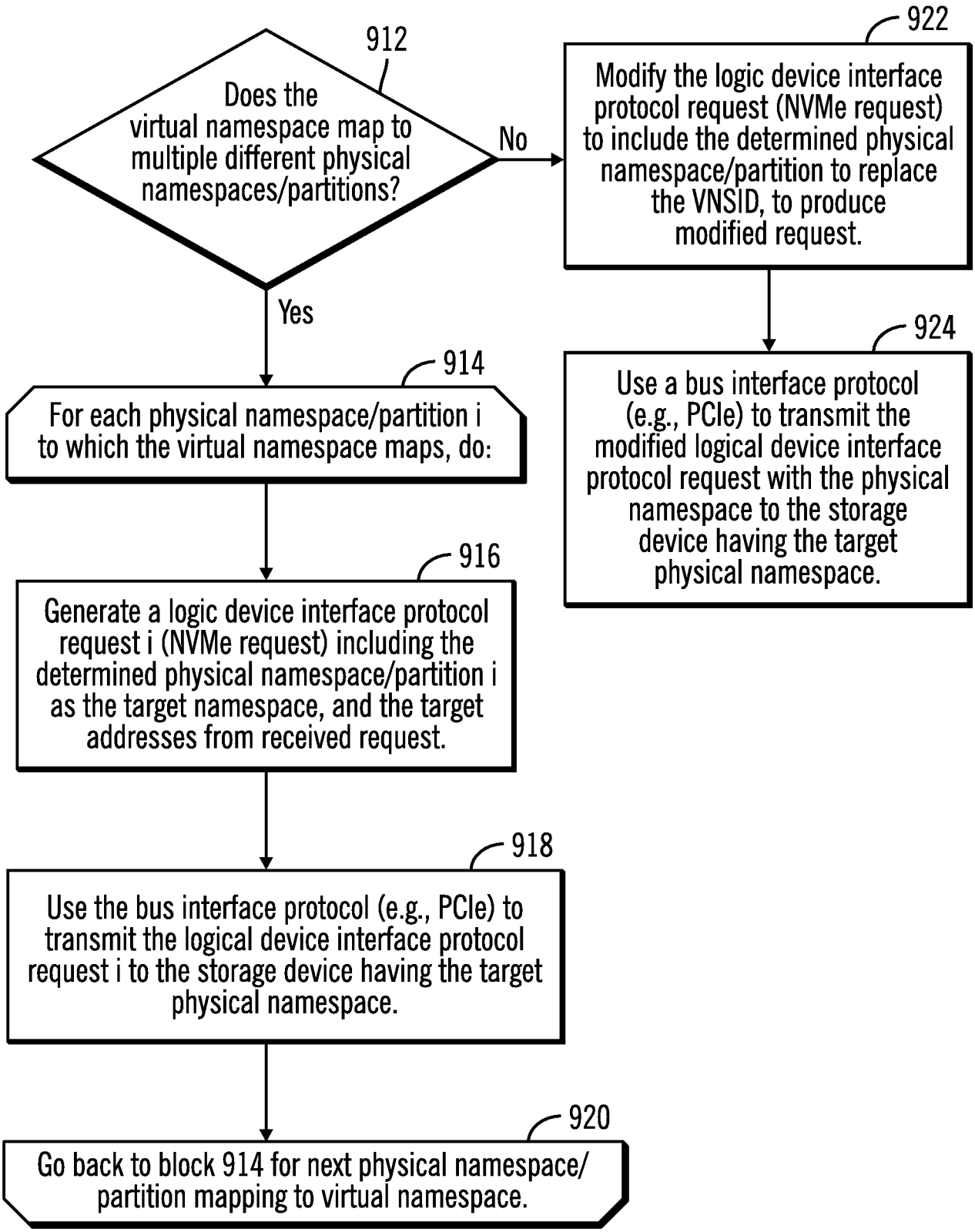

912
Does the virtual namespace map to multiple different physical namespaces/partitions?
No
Yes
922
Modify the logic device interface protocol request (NVMe request) to include the determined physical namespace/partition to replace the VNSID, to produce modified request.
924
Use a bus interface protocol (e.g., PCIe) to transmit the modified logical device interface protocol request with the physical namespace to the storage device having the target physical namespace.
914
For each physical namespace/partition i to which the virtual namespace maps, do:
916
Generate a logic device interface protocol request i (NVMe request) including the determined physical namespace/partition i as the target namespace, and the target addresses from received request.
918
Use the bus interface protocol (e.g., PCIe) to transmit the logical device interface protocol request i to the storage device having the target physical namespace.
920
Go back to block 914 for next physical namespace/ partition mapping to virtual namespace.

FIG. 9B

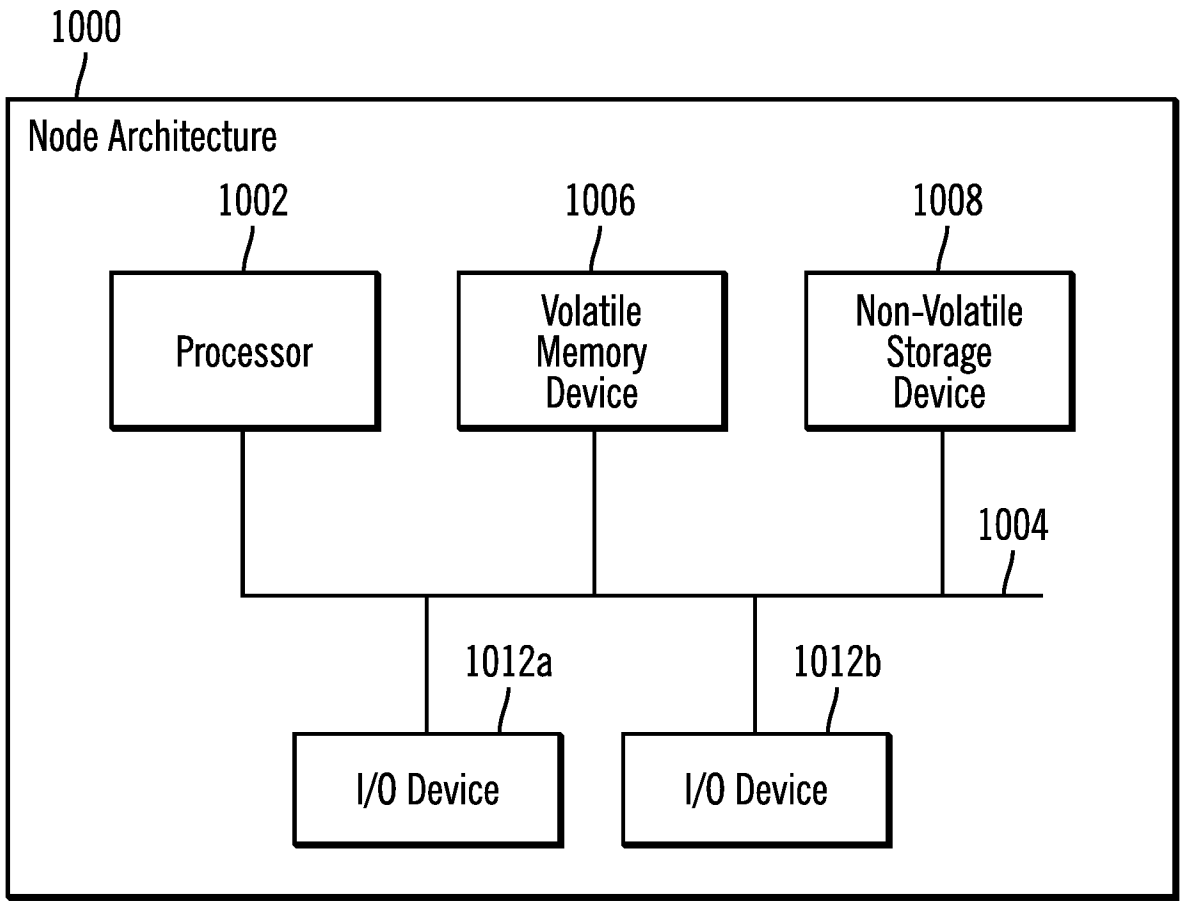

1000
Node Architecture
1002
1006
1008
Processor
Volatile Memory Device
Non-Volatile Storage Device
1004
1012a
1012b
I/O Device
I/O Device

FIG. 10

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US2017/029982

A. CLASSIFICATION OF SUBJECT MATTER

G06F 17/30(2006.01)i

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)
G06F 17/30; G06F 3/06; H04L 29/08; G06F 21/62; G06F 12/00; G06F 9/455; G06F 15/167

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
Korean utility models and applications for utility models
Japanese utility models and applications for utility models

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)
eKOMPASS(KIPO internal) & Keywords: host, target, over network, storage device, access, virtual namespace, physical namespace, mapping, NVMe, and similar terms.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2015-0319237 A1 (CAVIUM, INC.) 05 November 2015 See paragraphs [0017]-[0020], [0025]-[0028], [0032]-[0033], and [0035]; claims 1 and 14; and figures 1-3 and 5.	1-9,12-17,19-25
Y		10-11,18
Y	US 2016-0085718 A1 (CNEX LABS, INC.) 24 March 2016 See paragraphs [0047], [0050], [0052]-[0055], [0057], and [0074]; claims 1-2; and figures 1A-2 and 5.	10-11,18
A	US 2015-0143134 A1 (KABUSHIKI KAISHA TOSHIBA) 21 May 2015 See paragraphs [0021]-[0027] and figure 2.	1-25
A	US 2010-0114889 A1 (FARAMARZ RABII et al.) 06 May 2010 See paragraphs [0046]-[0051] and figure 2.	1-25
A	US 2014-0281040 A1 (FUTUREWEI TECHNOLOGIES, INC.) 18 September 2014 See paragraphs [0027]-[0042] and [0045]; and figures 1-3.	1-25

☐ 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
28 July 2017 (28.07.2017)	**28 July 2017 (28.07.2017)**

Name and mailing address of the ISA/KR
International Application Division
Korean Intellectual Property Office
189 Cheongsa-ro, Seo-gu, Daejeon, 35208, Republic of Korea
Facsimile No. +82-42-481-8578

Authorized officer
NHO, Ji Myong
Telephone No. +82-42-481-8528

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/US2017/029982

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2015-0319237 A1	05/11/2015	TW 201543225 A	16/11/2015
		TW 201543226 A	16/11/2015
		TW 201543366 A	16/11/2015
		TW 201543843 A	16/11/2015
		TW 201546717 A	16/12/2015
		TW 201617918 A	16/05/2016
		US 2015-0317088 A1	05/11/2015
		US 2015-0317091 A1	05/11/2015
		US 2015-0317176 A1	05/11/2015
		US 2015-0317177 A1	05/11/2015
		US 2015-0319243 A1	05/11/2015
		US 2016-0162438 A1	09/06/2016
		US 9294567 B2	22/03/2016
		US 9430268 B2	30/08/2016
		US 9501245 B2	22/11/2016
		US 9529773 B2	27/12/2016
US 2016-0085718 A1	24/03/2016	CN 105556930 A	04/05/2016
		TW I571087 B	11/02/2017
		US 2015-0006663 A1	01/01/2015
		US 2015-0378605 A1	31/12/2015
		US 2015-0378606 A1	31/12/2015
		US 2015-0378640 A1	31/12/2015
		US 2016-0357698 A1	08/12/2016
		US 9430412 B2	30/08/2016
		WO 2014-209764 A1	31/12/2014
		WO 2017-040706 A1	09/03/2017
US 2015-0143134 A1	21/05/2015	US 2016-0110295 A1	21/04/2016
		US 9245140 B2	26/01/2016
		US 9529735 B2	27/12/2016
US 2010-0114889 A1	06/05/2010	US 8078622 B2	13/12/2011
US 2014-0281040 A1	18/09/2014	US 9003071 B2	07/04/2015