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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
27 February 2014 (27.02.2014)

WIPO | PCT

(10) International Publication Number
WO 2014/031430 A1

(51) International Patent Classification:
H04L 12/933 (2013.01)

(21) International Application Number:
PCT/US2013/055086

(22) International Filing Date:
15 August 2013 (15.08.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
13/590,700 21 August 2012 (21.08.2012) US

(71) Applicant: **ADVANCED MICRO DEVICES, INC.** [US/US]; 1 AMD Place, Sunnyvale, California 94085 (US).

(72) Inventors: **CHERNOFF, Anton**; 451 King Street, #1521, Littleton, Massachusetts 01460 (US). **KRISHNAN, Venkata**; 26 Albert Ray Drive, Ashland, Massachusetts 01721 (US). **HUMMEL, Mark**; 68 Stewart Street, Franklin, Massachusetts 02038 (US). **MAYHEW, David**; 159 Pleasant Street, Northborough, Massachusetts 01532 (US). **OSBORN, Michael**; 50 Black Oak Drive, Hollis, New Hampshire 03049 (US).

(74) Agent: **COLLINS, Timothy P.**; Guerin & Rodriguez, LLP, 5 Mount Royal Avenue Mount Royal Office Park, Marlborough, Massachusetts 01752 (US).

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

(84) Designated States *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

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

[Continued on next page]

(54) Title: SYSTEMS AND METHODS FOR SHARING DEVICES IN A VIRTUALIZATION ENVIRONMENT

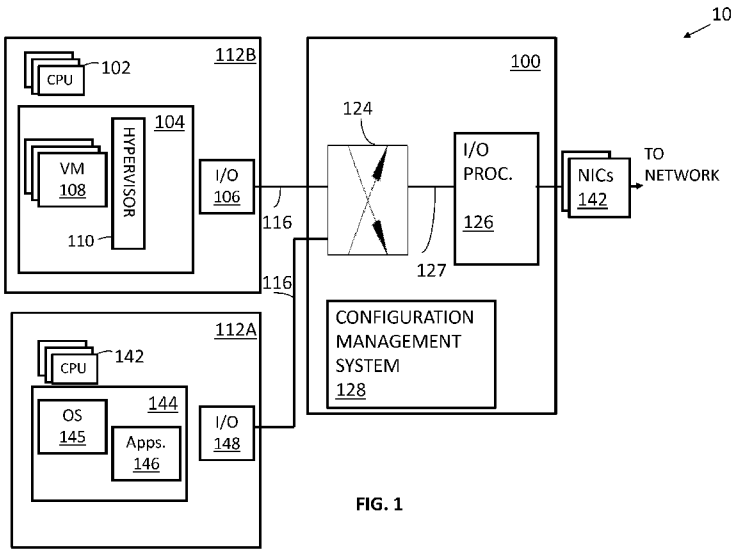

FIG. 1

(57) Abstract: Described are systems and methods for communication between a plurality of electronic devices and an aggregation device. An aggregation device processes instructions related to a configuration of an electronic device in communication with the aggregation device. One or more virtual devices are generated in response to processing the instructions. The electronic device enumerates a configuration space to determine devices for use by the electronic device. The aggregation device detects an access of the configuration space by the electronic device. The one or more virtual devices are presented from the aggregation device to the electronic device in accordance with the instructions.

WO 2014/031430 A1

— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

SYSTEMS AND METHODS FOR SHARING DEVICES IN A VIRTUALIZATION ENVIRONMENT

FIELD OF THE INVENTION

[0001] The present invention relates generally to virtualized networks, and more specifically, to systems and methods for input/output (I/O) virtualization.

BACKGROUND

[0002] Data centers are generally centralized facilities that provide Internet and intranet services supporting businesses and organizations. A typical data center can house various types of electronic equipment, such as computers, domain name system (DNS) servers, network switches, routers, data storage devices, and so on. A given data center can have hundreds or thousands of interconnected server nodes communicating with each other and external devices via a switching architecture comprising the switches and routers. Conventional data centers can be configured for virtualization, permitting the server nodes to share network interface cards (NICs), hard disk drives, or other hardware.

BRIEF SUMMARY OF EMBODIMENTS

[0003] In accordance with embodiments of the inventive concepts, provided is an apparatus and method that includes an aggregation device that makes available to a server node an unbounded number of virtual devices of different types, e.g., vNICs, vHBAs, and so on, which are generated independently of, and unconstrained by, a particular physical device, e.g., NIC, HBA, and so on. For virtualized nodes, the aggregation device can create virtual devices for each VM and allow the VMs to use the network, for example, through what appears to the node to be one or more dedicated NICs. To provision a new device for a node, the aggregation device can monitor the communication channels between the aggregation

device and any or all interconnected electronic devices, and determine when the node generates a request on its PCIe space, for example, a configuration space or a data space. The aggregation device provides a response, permitting the node to "discover" the requested virtual devices when it accesses its configuration space. The server node can process the virtual devices as though the virtual devices are physical devices for use by the node.

[0004] In one aspect, provided is a method for communication between a plurality of electronic devices and an aggregation device. At the aggregation device, instructions are provided that relate to a configuration of an electronic device in communication with the aggregation device. One or more virtual devices generate in response to processing the instructions. The electronic device enumerates a configuration space to determine devices for use by the electronic device. The aggregation device detects an access of the configuration space by the electronic device. The aggregation device presents to the electronic device, the one or more virtual devices in accordance with the instructions.

[0005] In another aspect, provided is a method for sharing physical devices in a virtualization environment. An electronic communication is established between an electronic device and an aggregation server. The aggregation server has a physical device; receiving a request at the aggregation server to provide the electronic device with access to a new device. A new range of addresses is detected at the electronic device. A virtual device is dynamically constructed at the aggregation server in response to the new range of addresses and arranging for the virtual device to at least one of transmit and receive data. An interrupt is sent to a processor at the electronic device. The processor discovers at the electronic device the virtual device as the new device.

[0006] In another aspect, provided is an apparatus. A virtual device generator generates one or more virtual devices for presentation to an electronic device of a plurality of electronic devices in communication with a configuration management system of the apparatus. A bus

interface module monitors a communication channel between the electronic device and the apparatus for addresses intended for transmitting or receiving data between the electronic device and the configuration management system. A data processor establishes a communication with the generated one or more virtual devices for processing the data between the electronic device and the configuration management system.

[0007] In another aspect, provided is a computer program product comprising a computer readable storage medium having computer readable program code embodied therewith. The computer readable program code comprises computer readable program code configured to process, at an aggregation device, instructions related to a configuration of an electronic device in communication with the aggregation device; computer readable program code configured to generate one or more virtual devices in response to processing the instructions; computer readable program code configured to enumerate, by the electronic device, a configuration space to determine devices for use by the electronic device; computer readable program code configured to detect, by the aggregation device, an access of the configuration space by the electronic device; and computer readable program code configured to present, from the aggregation device to the electronic device, the one or more virtual devices in accordance with the instructions.

BRIEF DESCRIPTION OF THE SEVERAL VIEWS OF THE DRAWINGS

[0008] The above and further advantages of this invention may be better understood by referring to the following description in conjunction with the accompanying drawings, in which like numerals indicate like structural elements and features in various figures. The drawings are not necessarily to scale, emphasis instead being placed upon illustrating the principles of the invention.

[0009] FIG. 1 is a diagram of a computing infrastructure, in which embodiments of the

present inventive concepts can be practiced;

[0010] FIG. 2 is a block diagram of the configuration management system of FIG. 1, in accordance with an embodiment;

[0011] FIG. 3 is a flow diagram illustrating a method for generating an unbounded number of virtual devices for a set of interconnected server nodes, in accordance with an embodiment;

[0012] FIG. 4 is a flow diagram illustrating a method for performing a data transfer operation, in accordance with an embodiment;

[0013] FIG. 5 is an illustration of data flow paths between various elements of a computing infrastructure during a data transfer operation, in accordance with another embodiment;

[0014] FIG. 6 is a method for dynamically generating a virtual device for a server node, in accordance with an embodiment; and

[0015] FIG. 7 is a method for modifying a number of virtual devices available to a server node in an operation, in accordance with an embodiment.

DETAILED DESCRIPTION OF EMBODIMENTS

[0016] In the following description, specific details are set forth although it should be appreciated by one of ordinary skill that the systems and methods can be practiced without at least some of the details. In some instances, known features or processes are not described in detail so as not to obscure the present invention.

[0017] Conventional I/O virtualization solutions can provide a finite number of virtualized devices, for example, virtual network interface cards (vNICs), virtual host bus adapters (vHBAs), or virtual basic input/output systems (vBIOSs). For example, a single-root input/output virtualization (SR-IOV) device or a multi-root input/output virtualization

(MR-IOV) device allows a physical device such as a NIC to appear as multiple NICs to a host machine. Accordingly, multiple guests at a host machine can share a single PCI Express® (PCIe) device. This can be achieved by the host hypervisor mapping one or more virtual functions to a guest, where each virtual function can appear as a single port NIC to the host operating system. However, SR-IOV and MR-IOV are hardware-dependent, in that the number of virtualized instances, e.g., virtual functions, of a given hardware type, e.g., NIC, HBA, BIOS, is hardware-dependent, and therefore constrained by the implementation of corresponding hardware. For example, a NIC is limited by hardware constraints with respect to the number of virtual functions that it can support, for example, limited to 64 virtual functions per device.

[0018] The present inventive concepts provide an I/O aggregation device coupled to a plurality of host machines, for example, server nodes or related processor devices such as PCIe processor nodes, and makes available to each node an unbounded number of virtual devices of different types, e.g., vNICs, vHBAs, and so on, which are generated independently of, and unconstrained by, a particular physical device, e.g., NIC, HBA, and so on. The virtual devices can be created at boot time. Alternatively, the number and type of virtual devices accessible by the node can be dynamically provided at run-time to accommodate node resource or processing requirements. For example, a number of virtual NICs can be increased to address bottlenecks at a node. New virtual devices can be provisioned for a node under the direction of a management system connected to the aggregation device, or by a request from the node's CPU, hypervisor, or other program involved with processing data requiring peripheral devices and the like, such as NICs, BIOS, drivers, storage devices, terminals, or other PCIe-compliant devices.

[0019] To provision a new device for a node, the aggregation device can monitor the communication channels between the aggregation device and any or all interconnected

electronic devices, and determine when the node generates a request on its PCIe space, for example, a configuration space or a data space. The aggregation device provides a response, permitting the node to "discover" the requested virtual devices when it accesses its configuration space. The node's CPU can generate data structures related to the virtual devices, and process the virtual devices as though the node interprets that the virtual devices are physical devices for use by the node. Due to the soft nature of the methods embodying certain aspects of the present inventive concepts, an unbounded number of virtual devices can be created at the aggregation device for any of the nodes connected to the aggregation device, preferably via PCIe communication channels.

[0020] In another embodiment, for virtualized processor nodes, i.e., nodes that are constructed and arranged to include a plurality of virtual machines (VM) having multiple guest operating systems, the aggregation device can create virtual devices for each VM and allow the VMs to use the network, for example, through what appears to the node to be one or more dedicated NICs. A hypervisor at the node is not required to intercept the guests' use of the virtualized devices, and the devices need not coordinate resources themselves, since the coordination is the responsibility of the virtual device software in the aggregation device.

[0021] During a data transfer operation, a node can process data using the virtual devices discovered during enumeration. The aggregation device can listen at a range of addresses spanning any or all server nodes having PCIe links with the aggregation device. After detecting a reference on a PCIe link to an address intended for the transmission of data, the aggregation device can emulate the physical device, for example, a network interface, intended to receive the data, and perform functions that the physical device would otherwise perform. For example, when a server node uses a virtual device to perform a data transfer or other I/O operation, the aggregation device can coordinate the handoff of the data from the server node to a physical device that can output the data to a network via a physical NIC or

other device. To achieve this, read and write operations can be performed by the aggregation device to and from the node's local memory.

[0022] FIG. 1 is a diagram of a computing infrastructure 10, in which embodiments of the present inventive concepts can be practiced. In the computing infrastructure 10, a virtualized server node 112B and a non-virtualized server node 112A (generally, 112) are in communication with an aggregation system 100. The aggregation system 100 can communicate with the server nodes 112 via a plurality of ports. Each port can be an I/O port and can provide access to a node-centric PCIe configuration space.

[0023] The server nodes 112 can be microservers, single socket servers, PCIe processor nodes, or other low-power electronic devices. The server nodes 112 can be constructed and arranged as a processor cluster or other well-known arrangement.

[0024] The virtualized server node 112B includes a processor 102, which can include one or more microprocessors, central processing units (CPUs), graphics processing units (GPUs), digital signal processors (DSPs), application-specific integrated circuits (ASICs), memory controllers, multi-core processors, or other types of data processing devices, or portions and combinations of these and other devices.

[0025] The virtualized server node 112B also includes a memory 104 and an I/O logic 106. The memory 104 can be a non-volatile or volatile memory, for example, DRAM or static RAM (SRAM). Stored at the memory 104 can include an operating system, one or more applications, and/or other program code. The I/O logic 106 can be configured to include a southbridge or related I/O controller for managing data transfers between the server node 112 and the aggregation system 100, and for performing other computer I/O functions. The I/O logic 106 can include an Ethernet PCIe port or related network connector and software, for example, Ethernet device drivers, for establishing a communication channel 116 with the aggregation system 100. A communication channel 116 can include at least one PCIe link.

Each PCIe link can comprise a plurality of PCIe lanes that form a channel with a server node 112.

[0026] In an embodiment, the memory 104 of the virtualized server node 112B includes one or more virtual machines 108, which share the one or more processors 102. Also stored in the memory 104 can include program code of an operating system, one or more applications, or other software programs executed by a processor 102.

[0027] The virtualized server node 112B can include a hypervisor 110. The hypervisor 110 can be configured to allow PCIe device functions and the like to be assigned to different VMs 108. In an embodiment, multiple virtualized server nodes 112B can share hardware, such as a NIC 142, or BIOS, HBA, or other hardware device via the aggregation system 100, and can therefore be constructed and arranged to include inexpensive, low-power components, achieved at least in part by eliminating expensive chips associated with NICs, BIOS, and the like.

[0028] The non-virtualized server node 112A includes a processor 142, a memory 144, and an I/O logic 148 that can be the same as or similar to counterpart elements of the virtualized server node 112B. Stored at the memory 144 can include an operating system 145, one or more applications 146, or other program code related to the performance of operations described herein.

[0029] The aggregation system 100 can include a switch fabric 124, an input/output (I/O) processor 126, and a configuration management system 128.

[0030] The switch fabric 124 provides a data plane interconnection between the server nodes 112, and includes a plurality of I/O ports (not shown) for exchanging data between the server nodes 112 and/or one or more remote electronic devices in communication with the aggregation system 100 via one or more NICs 142. The switch fabric 124 can include a switching configuration, for example, a crossbar, for moving data between the I/O ports. In

another embodiment, the switch fabric 124 can include memory switch, for example, described at U.S. Patent Application No. 13/526,973 filed June 19, 2012 entitled "Devices and Methods for Interconnecting Server Nodes" and U.S. Patent Application No. 13/529,452 filed June 21, 2012 entitled "Memory Switch for Interconnecting Server Nodes," the contents of each of which is incorporated by reference herein in its entirety. The data can be provided in variable or fixed length data packets, cells, and the like, for exchanging between the input and output ports, and for facilitating communication between the server nodes 112 and/or shared physical devices such as a NIC 142, and/or external electronic devices in communication with the system 100 via the NIC 142.

[0031] The switch fabric 124 and the I/O processor 126 can be interconnected via an I/O communication channel 127, which provides bandwidth for exchanging data between output ports at the switch fabric 124 and the I/O processor 126. The I/O processor 126 processes data transferred between the aggregation system 100 and the server nodes 112 and/or remote computing devices. The I/O processor 126 can examine incoming data packets and the like directed to a server node 112, and/or external computing devices via the NICs 142, and can route the data packets and the like to their destination, or output data packets to a remote device, for example, via the NIC 142, based on destination address information or other identification fields. The I/O processor 126 can include a packet processor that examines data packets to determine whether they should be filtered or forwarded.

[0032] The configuration management system 128 can receive and process instructions related to a configuration of one or more server nodes 112, for example, instructions on a type and number of virtual devices a server node can receive for enumeration, and generate the virtual devices in accordance with the instructions. Virtual devices generated at the configuration management system 128 can be used by the server nodes 112 for performing data transfer operations or related operations.

[0033] FIG. 2 is a block diagram of the configuration management system 128 of FIG. 1, in accordance with an embodiment.

[0034] The configuration management system 128 can include a management controller 202, a PCIe bus interface module 204, a virtual device generator 206, a data processor 212, and a node interrupt module 218.

[0035] The management controller 202 can receive instructions related to a node configuration from a source such as a network management system, a human operator, a data repository, or related computer having instruction data. A physical console or other configuration data source (not shown) can be coupled to the configuration management system 128 via an Ethernet connection (not shown) and the like for providing configuration information to the management controller 202. For example, a management console can be used to indicate how many NICs or other devices to provision for one or more electronic devices. In another embodiment, the management controller 202 includes one or more default configurations, whereby instructions and/or other configuration information are provided that define aspects of the configurations. Default or remotely provided configuration data can establish a maximum, minimum, and/or predetermined number of virtual devices for enumeration and processing by a server node 112. For example, the management controller 202 can receive an instruction from a controller or from the server node 112 that a predetermined number of NIC devices, for example, two NICs, are required for a given server node processor or virtual machine. In this example, the management controller 202 can process data including a requirement establishing that a server node processor, in accessing its configuration space related to an enumeration operation, can discover the two requested NICs, which in fact are two virtual devices provided by the configuration management system 128 that can emulate the desired NICs. Alternatively, instructions and/or configuration information can be provided to the management controller

202 via a virtual machine 108, more specifically, output by a hypervisor 110 communication with the management controller 202 via a PCIe connection. The management controller 202 can communicate with the virtual device generator 206, the data processor 212, and/or other elements of the aggregation system 100 to process received instructions.

[0036] The PCIe bus interface module 204 listens for PCIe configuration space accesses made by the server nodes 112 related to an enumeration operation. Accordingly, the PCIe bus interface module 204 can detect a reference to an address corresponding to the device intended for a transmission or receipt of data. Since the aggregation system 100 can be coupled to numerous server nodes 112 via PCIe connections, the PCIe bus interface module 204 can monitor all communication channels 116 for a reference to accesses related to particular addresses among a vast range of PCIe addresses, to identify server nodes 112 performing PCIe accesses to a configuration space. The configuration management system 128 can include a PCI bus interface module 204 for each PCIe port, or for multiple PCIe ports.

[0037] The virtual device generator 206 can generate a plurality of virtual devices that can be processed by the non-virtualized server node 112A and/or the non-virtualized server node 112B during an enumeration operation. The virtual device information presented for enumeration can be provided accordance with instructions processed by the management controller 202 or a default configuration, for example, an instruction that provides a maximum number of "devices" provided to a server node 112 for enumeration.

[0038] The virtual device generator 206 includes a lookup table 208 that can be configured to include a plurality of entries corresponding to the generated virtual devices. Each entry can include a bus address, for example, a PCIe address, and data for mapping the address and a generated virtual device that the server node 112 can discover during enumeration, and dynamically allocate resources to. For example, each PCIe address or

range of addresses can correspond to a virtual NIC that can be used by the server node 112 after the server node 112 performs a runtime search of its PCIe configuration space and generates the appropriate data structures after detecting the virtual NICs, subject to constraints provided by the management controller 202 and/or the server node 112. The number of virtual devices generated for one or more server nodes 112 is not dependent on the physical device emulated by the virtual devices, since the "devices" presented to the server node 112 for enumeration are entries at the lookup table 208.

[0039] The configuration management system 128 can populate the table 208 with entries in accordance with received instructions. Alternatively, the table 208 can include entries that include default configuration information. Each entry can provide data describing a requested "device" to be emulated. These data can be presented to the processor or virtual entity when it enumerates devices. Thus, when the processor or virtual entity accesses its configuration space, it will see during enumeration the number of requested devices according to the table entries.

[0040] The data processor 212 processes data provided by a server node 112 via the virtual devices provided to the server nodes 112. When a server node 112 outputs data using an assigned virtual device, the data processor 212 can access the table 208 to determine a destination for the data, for example, a physical device, e.g., NIC 142. This can be achieved by the data processor 212 mapping a PCIe address corresponding to an outbound virtual device used by the sending server node 112 for outputting the data. When data is received by the aggregation system 100 for a destination server node 112, the data processor 212 can determine from the table 208 the inbound virtual device at the destination server node 112 for receiving the data. The data processor 212 can determine from the PCIe bus interface module 204 that the data is being written to or read from a PCIe address corresponding to a generated virtual device that is identified as an entry at the table 208.

[0041] The node interrupt module 218 can interrupt a server node 112 to inform the server node 112 of a new "device", which in fact is a virtual device in the table 208. The server node 112 can "discover" the new device during an enumeration operation performed at the server node's PCIe configuration space.

[0042] FIG. 3 is a flow diagram illustrating a method 300 for generating a plurality of virtual devices for availability to a connected system of nodes, in accordance with an embodiment. In describing the method 300, reference is made to FIGs. 1 and 2. Some or all of the method 300 can be performed at the configuration management system 128, one or more server nodes 112, or a combination thereof.

[0043] At block 302, node configuration instructions are provided to the configuration management system 128. A network or other user can enter the instructions to a computer connected to the management controller 202. Alternatively, the instructions can be provided by the server node 112, for example, a hypervisor 110 at the server node 112. Accordingly, the configuration management system 128 can receive instructions regarding a number and/or type of virtual devices for the server node 112 to process during an enumeration operation. The generated virtual devices can appear as PCIe devices to a server node 112, for example, a physical NIC. Alternatively, the generated virtual devices can appear as vNICs and the like to a server node 112 configured to include virtual machines 108, a hypervisor 110, a virtual switch (not shown), and the like. Here, virtual devices can be generated by the configuration management system 128 for each virtual machine 108. A guest OS running on a virtual machine 108 in executing an enumeration operation can be intercepted by the hypervisor 110, which provides a plausible fiction of what is available for devices, i.e., virtual devices by the configuration management system 128 for the guest OS to use.

[0044] At block 304, the configuration management system 128 can create a set of virtual devices in response to the node configuration instructions, or according to preconfigured

rules provided by the rules engine 210, or based on other configuration instructions. The table 208 at the virtual device generator 206 can be populated with data related to the virtual devices, for example, the type of virtual device. The table 208 can include PCIe addresses for mapping to the virtual devices.

[0045] At block 306, the server node 112 executes an enumeration operation, for example, to enumerate existing devices. The enumeration operation can include a runtime search through the configuration space to discover the devices to be processed by the server node 112.

[0046] At block 308, the PCIe bus interface module 204 listens to all PCIe accesses made by server nodes 112 coupled to the aggregation system 100. Accordingly, the PCIe bus interface module 204 can detect a request from the server node 112 during an enumeration operation, when accessing its configuration space.

[0047] At block 310, the data processor 212 can provide a response to the enumeration operation by presenting information related to the created virtual devices to the server node 112. The information can be provided in accordance with the instructions and/or default configuration information, for example, establishing that when the requesting server node 112 looks at its configuration space, it will "discover" a predetermined number and type of devices. The information can be presented to the requesting server node 112 in a manner that appears as physical devices and/or virtual devices, depending on the requirements and capabilities of the server node 112. The information can include addresses where the control and/or status registers related to the created virtual devices can be accessed, permitting the server node 112 to communicate with the virtual devices using these addresses on the PCIe bus 116 as though they are "real" devices.

[0048] At block 312, the operating system of the server node 112 can be configured to communicate with the virtual devices, for example, by creating data structures using the

information provided at block 310. Accordingly, at block 314, data is exchanged between the server node 112 and the aggregation system 100, which the server node 112 is configured to believe is a PCIe device, when in fact the aggregation system 100 is emulating a PCIe device. The data can include control-related information, and or response data. The data can include payload data. The virtualized server node 112B can run a hypervisor that provides the virtual devices to one or more guests. A non-virtualized server node 112A can use the virtual devices as though they are physical devices, for example, Ethernet ports.

[0049] FIG. 4 is a flow diagram illustrating a method 400 for performing a data transfer operation, in accordance with an embodiment. In describing the method 400, reference is made to FIGs. 1-3. The server node 112 can be constructed and arranged to exchange data with the virtual devices presented to a server node 112 in accordance with the method 400.

[0050] At block 402, a request is made at the server node 112 over a bus 116, for example, a PCIe bus, for the virtual device to process the data. For example, a CPU, virtual machine, or other processing element of the server node 112 can send a request, for example, an Ethernet request, on the PCIe bus 116 using control and status registers provided during the enumeration operation. Control information and/or other data related to a PCIe address of a range of addresses corresponding to the virtual device can be output on the PCIe bus 116 to the aggregation system 200.

[0051] The PCIe bus interface module 204 can monitor the PCIe bus 116 for a range of addresses corresponding to the control and status registers of the virtual devices assigned to the server node 112. At block 404, the PCIe bus interface module 204 can detect reference to an address of the PCIe address range corresponding to the device that the server node 112 intends to send data, or receive data.

[0052] At block 406, the data processor 212 can access the lookup table 208 to determine, for example, at the data processor 212, that the data is being written to a PCIe address

corresponding to a virtual device assigned to the requesting server node 112. The data processor 212 can access the lookup table 208 to determine the manner in which to process the data received from the server node 112. s described above, the actual responder to the request is not a physical device, but is instead a software module designated by the entry at the lookup table 208 representing the "device" presented to the server node 112 during enumeration.

[0053] At block 408, the data processor 212 can copy data from the server node 112 to a buffer (not shown) at the aggregation system 100. This can be achieved by outputting descriptors, interrupts, or other related data during the data transfer. The aggregation system 100 can subsequently process the data via the virtual device, which can perform the functions of an actual NIC, such as exchanging descriptors with buffers at the server node 112, or performing routing-related functions such as extracting header information from the data, transmitting the data over the switch fabric 124 to another server node 112, or an external device via the I/O processor 126, and so on.

[0054] FIG. 5 is an illustration of data flow paths between various elements of a computing infrastructure 20 during a data transfer operation, in accordance with another embodiment. Prior to the data transfer operation, the method 300 described at FIG. 3 can be executed. In performing an enumeration step of method 300, a virtualized server node 112B can perform a data transfer operation at one or more virtual devices 302-1, 302-2, 302-3 (generally, 302) at an aggregation system 500. In FIG. 5, the server node 112B after performing an enumeration operation can output data via one or more of the three virtual devices 302 presented by the aggregation system 500. In other embodiments, any number of virtual devices can be generated. In other embodiments, a non-virtualized server node can be connected to the aggregation system 500 and can process virtual devices presented by the aggregation system 500 during enumeration. In other embodiments, data can be exchanged

between the server node 112B and remote devices in the communication with the aggregation system 500 via a NIC 142. In other embodiments, the NIC 142 is not required for communications between server node 112B and other devices in direct or indirect communication with the aggregation system 500, for example, another server node 112.

[0055] The server node 112B generates a request 502 that is output to the aggregation system 500 for the designated virtual device 302-1 to process data designated for a destination device, for example, another server node connected to the aggregation system 500, or a remote device via a NIC 142 connected to the aggregation system 500. The CPU 102 of the server node 112B can build control and/or data information, then output a request out on the PCIe bus 116. In an embodiment, the request is an Ethernet request.

[0056] The PCIe bus interface module 204 detects a communication 504 at a range of PCIe addresses assigned to the server node 112B, more specifically, a PCIe address among the range of PCIe addresses that corresponds to the designated virtual device 302-1 generated at the virtual device generator 206, and intended for transmitting the server node data to a destination.

[0057] The data processor 212 can access 506 the lookup table 208 for mapping information related to the detected PCIe address and an outbound virtual device that the server node 112B believes is outputting the data sent from the server node 112B to the aggregation system 300. For example, a first entry in the table 208 can include data D1 related to a virtual device 302-1 and a corresponding PCIe address A1. A second entry can include data D2 related to a virtual device 302-2 and a corresponding PCIe address A2. A third entry can include data D3 related to a virtual device 302-2 and a corresponding PCIe address A3. A comparison can be made between the PCIe address and a corresponding entry at the lookup table 208 to determine if the virtual device for which the data received from the server node 112B is processed.

[0058] A request is sent to the aggregation system 500, which then proceeds to directly access the memory on the server node 112B to transfer the data. The request can contain the type of transfer, remote node address, and address/size of the data within the server node 112B virtual memory. The CPU 102 of the server node 112B can communicate with the assigned virtual device, for example, virtual device 302-1, which is believed by the server node 112B to be a PCIe device. The virtual device 302-1 can exchange descriptors with the server node 112B or perform related operations related to the exchange of data. The data processor 212 can directly access 508 the data for transmission from the server node 112B. The data is accessed after the descriptors are exchanged. Accordingly, the virtual device 302-1, which is generated from the data A1, D1 at the lookup table, receives 510 the data from the server node 112B, and outputs 512 the data to its destination, in this example, a device via a NIC 142. In another example, the data can be output to a destination server node (not shown) coupled to the aggregation system 500. Here, the virtual device 302-1 can transfer the data to the receiving server node, send an interrupt, and so on.

[0059] FIG. 6 is a method 600 for generating an unbounded number of virtualized devices for a server node, in accordance with an embodiment. In describing the method 600, reference is made to FIGs. 1-5.

[0060] At block 602, a new virtual machine guest is created at a virtualized server node 112B. The new guest can be created according to virtualization techniques known to those of ordinary skill in the art.

[0061] At block 604, the server node 112B generates a request for a new device. For example, the hypervisor 110 may send a request to the aggregation system 100 that a virtual NIC is to be provided for the new guest.

[0062] At block 606, the configuration management system 128 can dynamically configure a virtual device for the new guest. An entry can be created at the lookup table 208

that identifies the new virtual device.

[0063] At block 608, the configuration management system 128 can assign a PCIe address range at the PCIe space of the server node 132. The configuration management system 128 can also provide mapping information between a PCIE address and the virtual device, a pointer to a physical device, or/or information related to the generated virtual device.

[0064] At block 610, the guest is informed to access the node configuration space for the requested virtual device. Accordingly, the method 600 permits virtual devices to be created for each virtual machine at one or more server nodes, and to allow the virtual machines to use a network, for example, through a plurality of virtual devices that appear to the virtual machines as one or more dedicated NICs or other hardware devices. The hypervisor 110 is not required to receive the virtual devices, and need not intercept the guests' use of their virtual devices. Instead, the aggregation system 100 coordinates the various resources required to process data to and from the virtual devices.

[0065] FIG. 7 is a method 700 for modifying a number of virtual devices available to a server node in an operation, in accordance with an embodiment. In describing the method 700, reference can be made to elements of FIGs. 1-6.

[0066] In accordance with an embodiment, the configuration management system 128 can dynamically modify a pool of available resources for a server node 112 according to changing requirements of the server node 112. For example, an operator such as a network manager and/or a network management system can determine that a server node 112 is experiencing congestion at its I/O ports, and that additional physical or virtual devices are required to alleviate the congestion. A determination can be made that that the congestion can be reduced or eliminated by adding a device that is recognized by the server node 112 as a NIC or other device, but is actually a virtual device generated at the aggregation system

100.

[0067] At block 702, the management controller 202 can generate a request for a new NIC for the server node 112, more specifically, a virtual device that emulates a NIC, or other device such as a DMA device, terminals, storage device, or other PCIe device. The management controller 202 can generate the request in response to a request or instructions provided by the server node 112, or by the network management system.

[0068] At block 704, the virtual device generator 206 creates the structures for responding to the subrange of PCIe addresses corresponding to the server node 112. In doing so, an entry is generated for the lookup table 208 to include data for mapping the PCIe address to the new virtual device.

[0069] At block 706, the node interrupt module 718 can generate an interrupt that is output to the server node 112 to inform the server node 112 of the new virtual device according to the table entry contents. At block 708, the server node 112 can discover the new device during an enumeration operation performed at the server node's PCIe configuration space.

[0070] As will be appreciated by one skilled in the art, aspects of the present invention may be embodied as a system, method or computer program product. Accordingly, aspects of the present invention may take the form of an entirely hardware embodiment, an entirely software embodiment (including firmware, resident software, micro-code, etc.) or an embodiment combining software and hardware aspects that may all generally be referred to herein as a "circuit," "module" or "system." Furthermore, aspects of the present invention may take the form of a computer program product embodied in one or more computer readable medium(s) having computer readable program code embodied thereon.

[0071] Any combination of one or more computer readable medium(s) may be utilized. The computer readable medium may be a computer readable signal medium or a computer

readable storage medium. A computer readable storage medium may be, for example, but not limited to, an electronic, magnetic, optical, electromagnetic, infrared, or semiconductor system, apparatus, or device, or any suitable combination of the foregoing. More specific examples (a non-exhaustive list) of the computer readable storage medium would include the following: an electrical connection having one or more wires, a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), an optical fiber, a portable compact disc read-only memory (CD-ROM), an optical storage device, a magnetic storage device, or any suitable combination of the foregoing. In the context of this document, a computer readable storage medium may be any tangible medium that can contain, or store a program for use by or in connection with an instruction execution system, apparatus, or device.

[0072] A computer readable signal medium may include a propagated data signal with computer readable program code embodied therein, for example, in baseband or as part of a carrier wave. Such a propagated signal may take any of a variety of forms, including, but not limited to, electro-magnetic, optical, or any suitable combination thereof. A computer readable signal medium may be any computer readable medium that is not a computer readable storage medium and that can communicate, propagate, or transport a program for use by or in connection with an instruction execution system, apparatus, or device. Program code embodied on a computer readable medium may be transmitted using any appropriate medium, including but not limited to wireless, wireline, optical fiber cable, RF, etc., or any suitable combination of the foregoing.

[0073] Computer program code for carrying out operations for aspects of the present invention may be written in any combination of one or more programming languages, including an object oriented programming language such as Java, Smalltalk, C++ or the like

and conventional procedural programming languages, such as the "C" programming language or similar programming languages. The program code may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider).

[0074] Aspects of the present invention are described herein with reference to flowchart illustrations and/or block diagrams of methods, apparatus (systems) and computer program products according to embodiments of the invention. It will be understood that each block of the flowchart illustrations and/or block diagrams, and combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer program instructions. These computer program instructions may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing apparatus, create means for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks.

[0075] These computer program instructions may also be stored in a computer readable medium that can direct a computer, other programmable data processing apparatus, or other devices to function in a particular manner, such that the instructions stored in the computer readable medium produce an article of manufacture including instructions which implement the function/act specified in the flowchart and/or block diagram block or blocks. The computer program instructions may also be loaded onto a computer, other programmable data processing apparatus, or other devices to cause a series of operational steps to be performed

on the computer, other programmable apparatus or other devices to produce a computer implemented process such that the instructions which execute on the computer or other programmable apparatus provide processes for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks.

[0076] The flowchart and block diagrams in the figures illustrate the architecture, functionality, and operation of possible implementations of systems, methods and computer program products according to various embodiments of the present invention. In this regard, each block in the flowchart or block diagrams may represent a module, segment, or portion of code, which comprises one or more executable instructions for implementing the specified logical function(s). It should also be noted that, in some alternative implementations, the functions noted in the block may occur out of the order noted in the figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending upon the functionality involved. It will also be noted that each block of the block diagrams and/or flowchart illustration, and combinations of blocks in the block diagrams and/or flowchart illustration, can be implemented by special purpose hardware-based systems that perform the specified functions or acts, or combinations of special purpose hardware and computer instructions.

[0077] While the invention has been shown and described with reference to specific embodiments, it should be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention.

What is claimed is:

1. A method for communication between a plurality of electronic devices and an aggregation device, the method comprising:

processing, at the aggregation device, instructions related to a configuration of an electronic device in communication with the aggregation device;

generating one or more virtual devices in response to processing the instructions;

enumerating, by the electronic device, a configuration space to determine devices for use by the electronic device;

detecting, by the aggregation device, an access of the configuration space by the electronic device; and

presenting, from the aggregation device to the electronic device, the one or more virtual devices in accordance with the instructions.

2. The method of claim 1, further comprising:

performing a data transfer operation between the electronic device and the one or more virtual devices at the aggregation device.

3. The method of claim 2, wherein the data transfer operation includes the exchange of command and response information between the electronic device and the aggregation device.

4. The method of claim 2, wherein performing the data transfer operation includes:

detecting, by the aggregation device, a reference to an address, the address corresponding to a generated virtual device of the one or more virtual devices;

determining that the data in the data transfer operation is a command intended related to a transmission of data or a response related to a receipt of data by the virtual device;

determining, in response to detecting the reference to the address, that the data is written to or read from the address; and

processing an operation related to the data written to or read from the address at the virtual device.

5. The method of claim 1, further comprising:

generating, at the electronic device, a request for a virtual device of the one or more virtual devices to provide data to a destination via the aggregation device;

intercepting the request at the aggregation device;

copying the data from the electronic device to the aggregation device; and

processing the data by the virtual device at the aggregation device.

6. The method of claim 1, wherein the one or more virtual devices include a virtual network interface card (NIC), a virtual host bus adapter (vHBA), a virtual basic input/output system (vBIOS), or a combination therof.

7. The method of claim 1, further comprising:

monitoring, by the aggregation device, all communication channels with the plurality of electronic devices for a plurality of ranges of addresses assigned to the plurality of electronic devices.

8. The method of claim 7, wherein the range of addresses includes a subset of addresses corresponding to the electronic device of the plurality of electronic devices.

9. The method of claim 7, wherein performing the data transfer operation comprises:

detecting, by the aggregation device at a bus between the electronic device and the aggregation device, a request at an address of the subset of addresses, the request directed at a virtual device of the one or more virtual devices intended for transmitting command data to the electronic device.

10. The method of claim 1, further comprising listening for PCIe accesses to a configuration space at the electronic device to determine whether a request is generated for new devices.

11. The method of claim 1, wherein sending the request for available devices from the

electronic device to the aggregation device comprises generating the request at a hypervisor at the electronic device.

12. The method of claim 1, further comprising:

providing, by a configuration data source in communication with the aggregation device, the instructions.

13. The method of claim 1, further comprising:

providing, by a communication from a management console to the aggregation device, instructions on a number and type of virtual devices that are to be created for use by the electronic device of the plurality of electronic devices.

14. The method of claim 1, further comprising:

providing, by a default configuration within the aggregation device, instructions on a number and type of virtual devices that are to be created for use by the electronic device of plurality of electronic devices.

15. A method for sharing physical devices in a virtualization environment, comprising:

establishing an electronic communication between an electronic device and an aggregation server, the aggregation server having a physical device;

receiving a request at the aggregation server to provide the electronic device with

access to a new device;

detecting a new range of addresses at the electronic device;

dynamically constructing a virtual device at the aggregation server in response to the new range of addresses and arranging for the virtual device to at least one of transmit and receive data;

sending an interrupt to a processor at the electronic device; and

discovering, by the processor at the electronic device, the virtual device as the new device.

16. The method of claim 15, wherein the data is transmitted or received between the

electronic device and another electronic device in communication with the aggregation device.

17. The method of claim 16, wherein the electronic device and the other electronic device are server nodes.

18. The method of claim 15, wherein the electronic device is a virtualized node or a non-virtualized node.

19. The method of claim 15, further comprising:

enumerating, by the electronic device, a configuration space to determine the new device for use by the electronic device;

20. The method of claim 15, further comprising:

generating, at the electronic device, a request for the virtual device to provide data to a destination via the aggregation device;

intercepting the request at the aggregation device;

copying the data from the electronic device to the aggregation device; and

processing the data by the virtual device at the aggregation device.

21. An apparatus, comprising:

a virtual device generator that generates one or more virtual devices for presentation to an electronic device of a plurality of electronic devices in communication with a configuration management system of the apparatus;

a bus interface module that monitors a communication channel between the electronic device and the apparatus for addresses intended for transmitting or receiving data between the electronic device and the configuration management system; and

a data processor that establishes a communication with the generated one or more virtual devices for processing the data between the electronic device and the configuration management system.

22. The apparatus of claim 21, further comprising a rules engine that processes

instructions that establish configuration parameters regarding the use of the virtual devices by the electronic device.

23. The apparatus of claim 22, wherein the rules engine generates a rule from the instructions that establishes that the electronic device identifies a virtual device during an enumeration-related operation.

24. The apparatus of claim 22, further comprising a management controller that provides the instructions to the rules engine.

25. The apparatus of claim 24, wherein the management controller receives the instructions from a table or other source regarding a number of devices for enumeration and/or other configuration information.

26. The apparatus of claim 24, wherein the management controller includes one or more default configurations, whereby the instructions are provided from the default configurations.

27. The apparatus of claim 24, further comprising a console in communication with the management controller for providing the instructions to the management controller.

28. The apparatus of claim 22, further comprising a configuration data source that provides the instructions to the rules engine.

29. The apparatus of claim 21, wherein the bus interface module listens for PCIe accesses to a configuration space at the electronic device to determine whether a request is generated for new devices.

30. The apparatus of claim 21, wherein the virtual device generator includes a lookup table that includes a plurality of entries.

31. The apparatus of claim 30, wherein each entry includes a PCIe address and data

used by the electronic device to believe a real device is available to it during enumeration.

32. The apparatus of claim 21, further comprising a management console that provides instructions on a number and type of virtual devices that are to be created for use by the electronic device.

33. The apparatus of claim 21, further comprising:

a default configuration that includes instructions on a number and type of virtual devices that are to be created for use by the electronic device.

34. A computer program product, comprising:

a computer readable storage medium having computer readable program code embodied therewith, the computer readable program code comprising:

computer readable program code configured to process, at an aggregation device, instructions related to a configuration of an electronic device in communication with the aggregation device;

computer readable program code configured to generate one or more virtual devices in response to processing the instructions;

computer readable program code configured to enumerate, by the electronic device, a configuration space to determine devices for use by the electronic device;

computer readable program code configured to detect, by the aggregation device, an access of the configuration space by the electronic device; and

computer readable program code configured to present, from the aggregation device to the electronic device, the one or more virtual devices in accordance with the instructions.

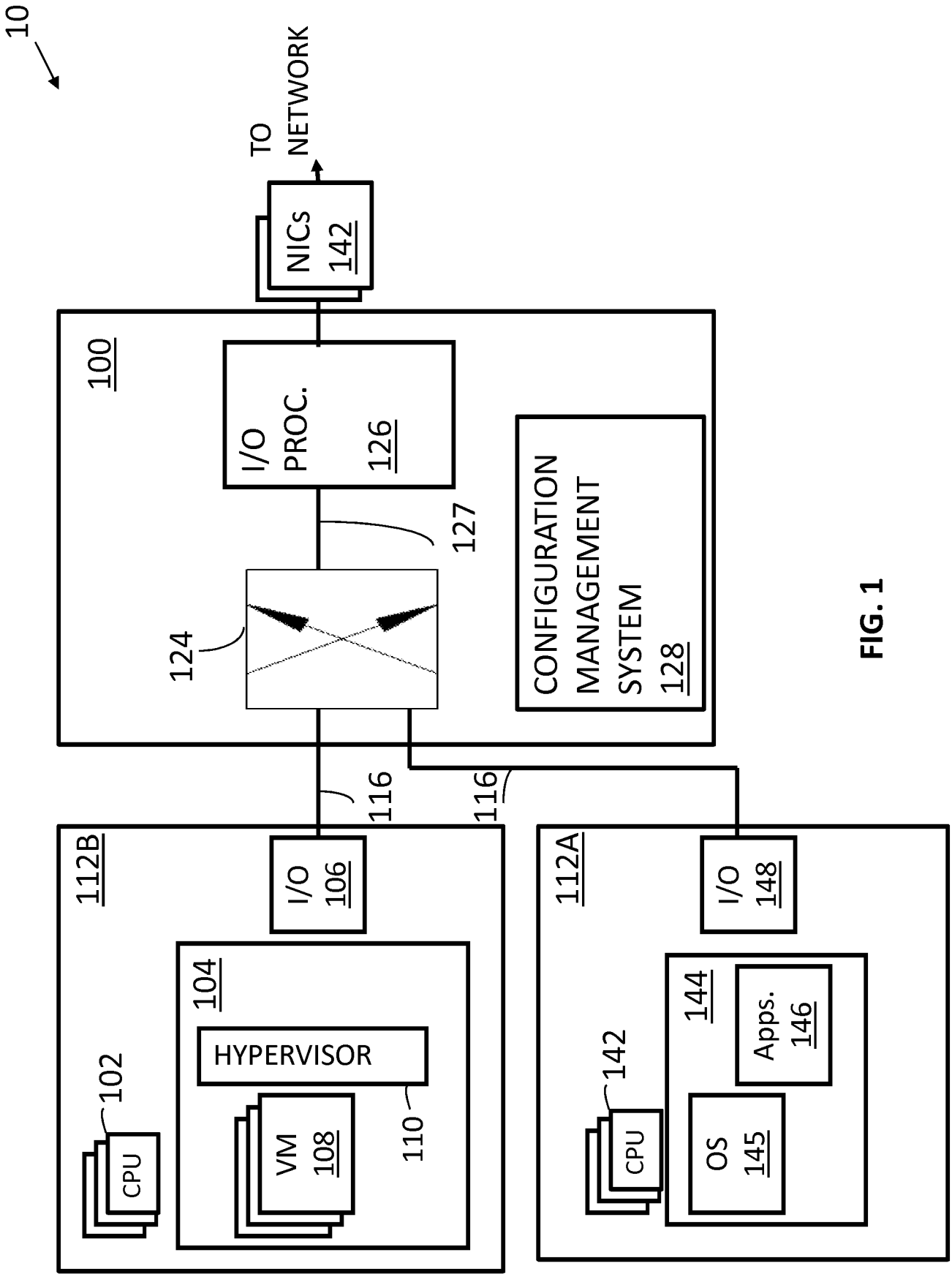
10
TO NETWORK
NICs 142
100
I/O PROC. 126
127
124
CONFIGURATION MANAGEMENT SYSTEM 128
116
116
112B
I/O 106
104
HYPERVISOR
VM 108
110
102
CPU
112A
I/O 148
144
Apps. 146
OS 145
142
CPU

FIG. 1

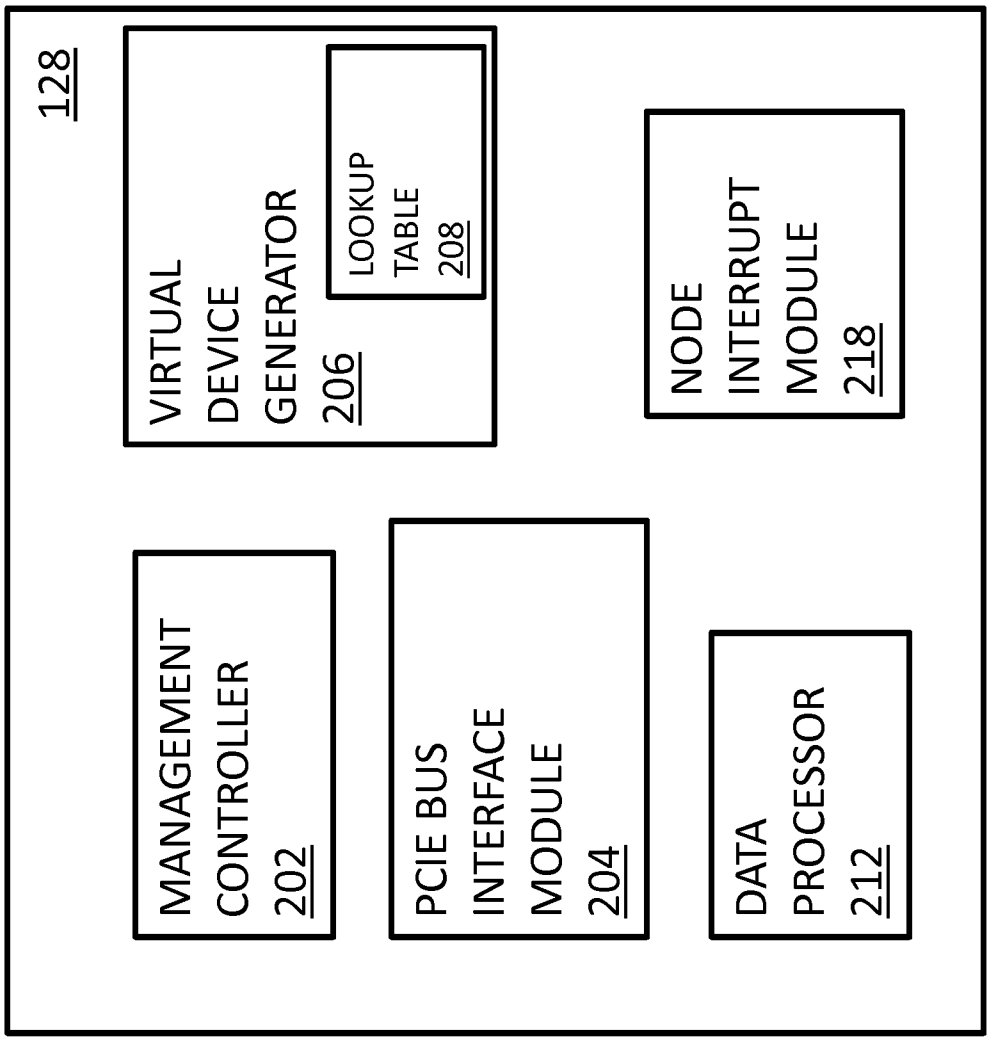
128
VIRTUAL DEVICE GENERATOR 206
LOOKUP TABLE 208
NODE INTERRUPT MODULE 218
MANAGEMENT CONTROLLER 202
PCIE BUS INTERFACE MODULE 204
DATA PROCESSOR 212

FIG. 2

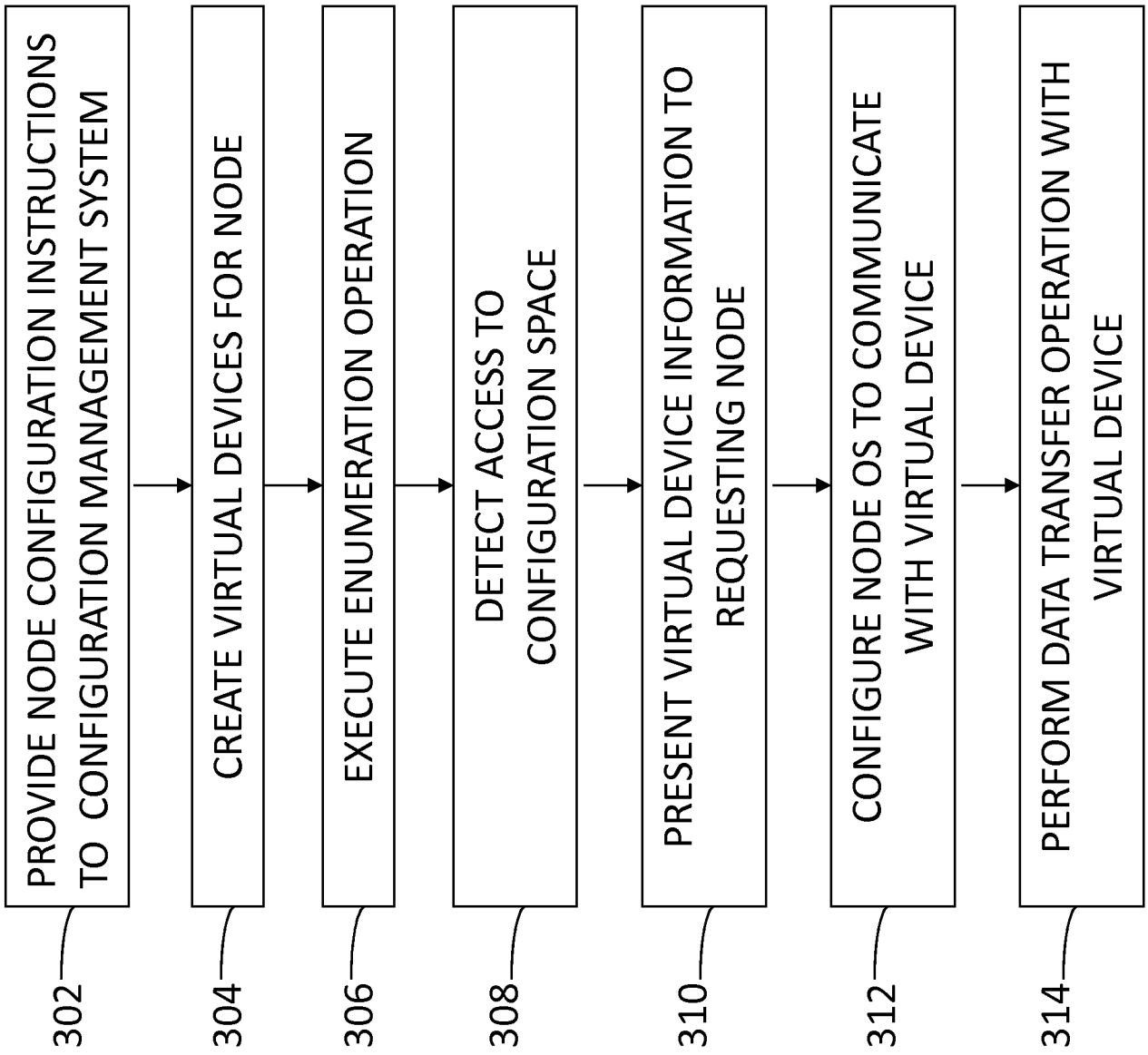
300
PROVIDE NODE CONFIGURATION INSTRUCTIONS TO CONFIGURATION MANAGEMENT SYSTEM
CREATE VIRTUAL DEVICES FOR NODE
EXECUTE ENUMERATION OPERATION
DETECT ACCESS TO CONFIGURATION SPACE
PRESENT VIRTUAL DEVICE INFORMATION TO REQUESTING NODE
CONFIGURE NODE OS TO COMMUNICATE WITH VIRTUAL DEVICE
PERFORM DATA TRANSFER OPERATION WITH VIRTUAL DEVICE
302
304
306
308
310
312
314

FIG. 3

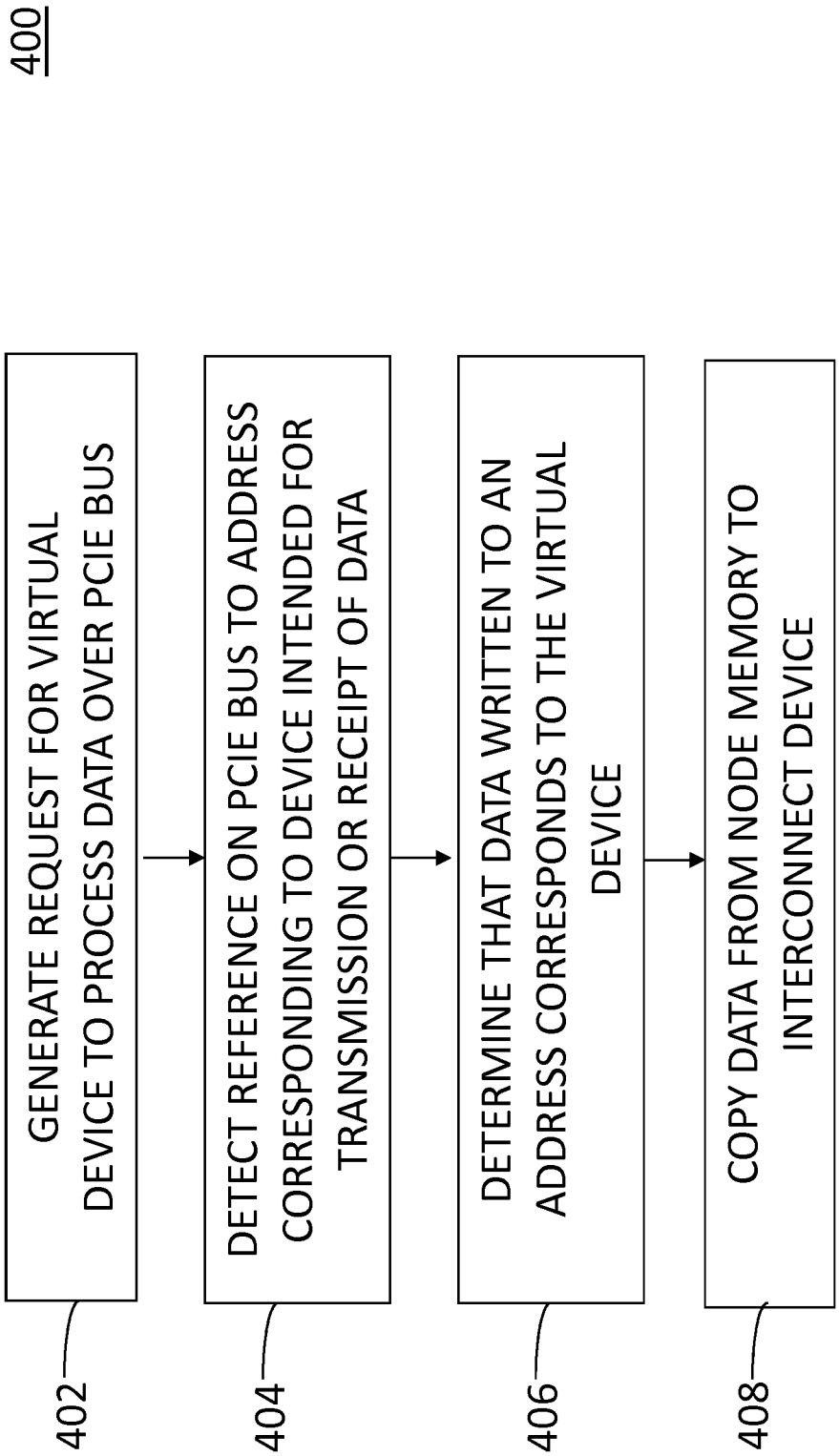
400
GENERATE REQUEST FOR VIRTUAL DEVICE TO PROCESS DATA OVER PCIE BUS
DETECT REFERENCE ON PCIE BUS TO ADDRESS CORRESPONDING TO DEVICE INTENDED FOR TRANSMISSION OR RECEIPT OF DATA
DETERMINE THAT DATA WRITTEN TO AN ADDRESS CORRESPONDS TO THE VIRTUAL DEVICE
COPY DATA FROM NODE MEMORY TO INTERCONNECT DEVICE
402
404
406
408

FIG. 4

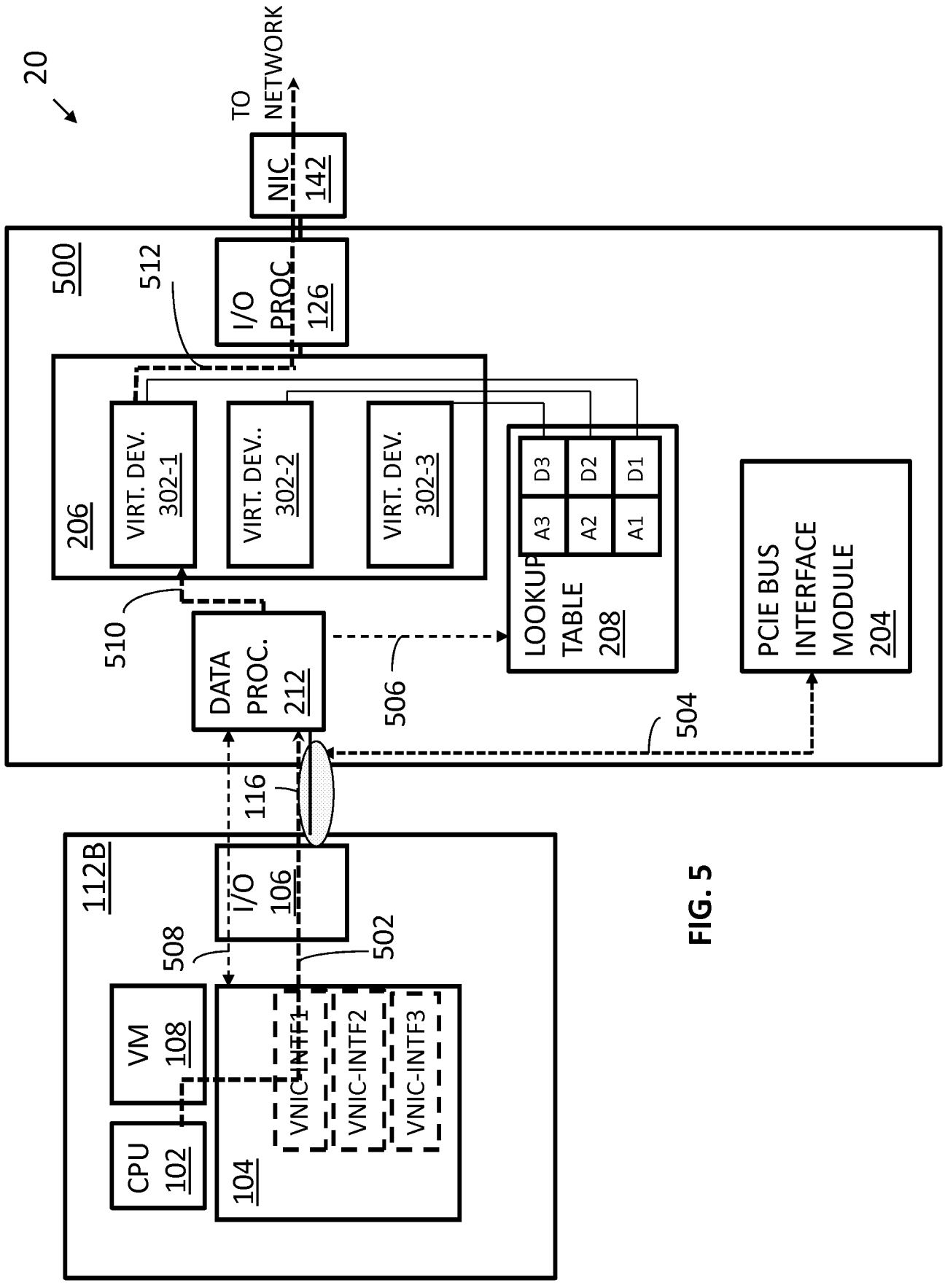
20
TO NETWORK
NIC 142
500
512
I/O PROC. 126
206
VIRT. DEV. 302-1
VIRT. DEV.. 302-2
VIRT. DEV. 302-3
LOOKUP TABLE 208
D3
D2
D1
A3
A2
A1
PCIE BUS INTERFACE MODULE 204
510
DATA PROC. 212
506
504
116
112B
I/O 106
508
502
VM 108
CPU 102
104
VNIC-INTF1
VNIC-INTF2
VNIC-INTF3

FIG. 5

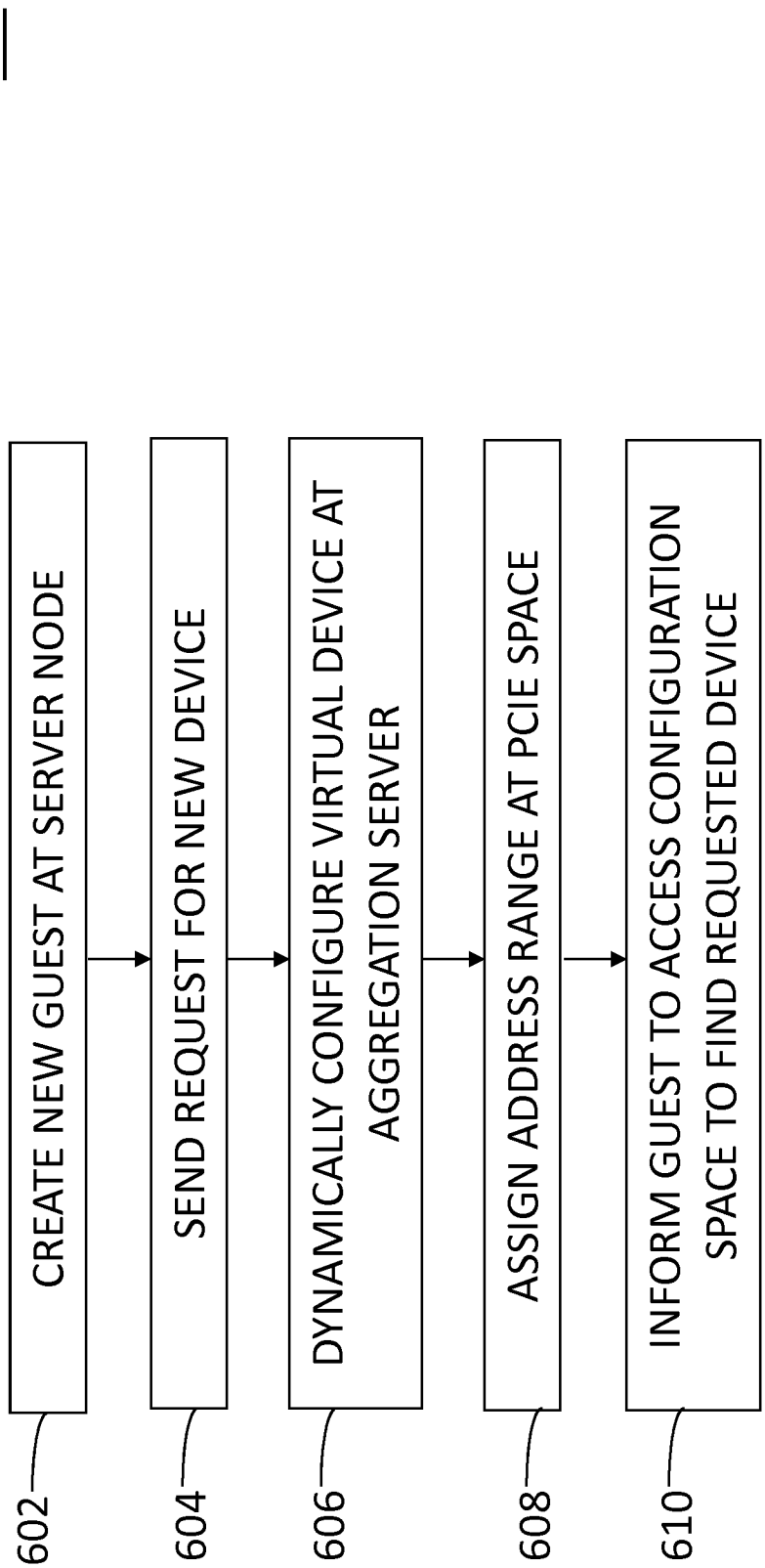
600
CREATE NEW GUEST AT SERVER NODE
SEND REQUEST FOR NEW DEVICE
DYNAMICALLY CONFIGURE VIRTUAL DEVICE AT AGGREGATION SERVER
ASSIGN ADDRESS RANGE AT PCIE SPACE
INFORM GUEST TO ACCESS CONFIGURATION SPACE TO FIND REQUESTED DEVICE
602
604
606
608
610

FIG. 6

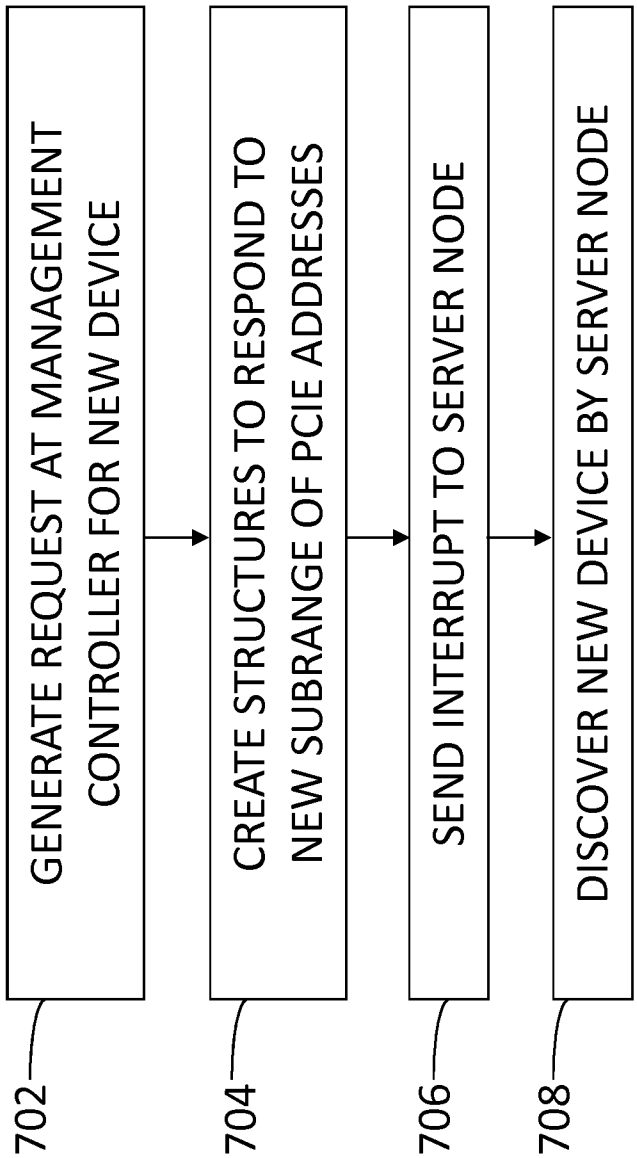
GENERATE REQUEST AT MANAGEMENT CONTROLLER FOR NEW DEVICE
CREATE STRUCTURES TO RESPOND TO NEW SUBRANGE OF PCIE ADDRESSES
SEND INTERRUPT TO SERVER NODE
DISCOVER NEW DEVICE BY SERVER NODE
702
704
706
708

FIG. 7

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2013/055086

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04L12/933
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)
H04L 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 2012/102491 A1 (MAHARANA PARAG R [US]) 26 April 2012 (2012-04-26) abstract paragraph [0021] - paragraph [0035]; figures 2A, 2B paragraph [0061]; figure 8 -----	1-34
A	US 7 502 884 B1 (SHAH SHREYAS [US] ET AL) 10 March 2009 (2009-03-10) abstract column 4, line 20 - column 5, line 33 column 8, line 30 - line 46 column 13, line 48 - column 14, line 22 column 14, line 37 - line 43 ----- -/--	1,15,21, 34

[X] Further documents are listed in the continuation of Box C.

[X] 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 December 2013	20/12/2013

Name and mailing address of the ISA/	Authorized officer
European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Von Der Straten, G

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2013/055086

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2012/042095 A1 (KOTHA SAIKRISHNA [US] ET AL) 16 February 2012 (2012-02-16) abstract paragraph [0013]; figure 1 paragraph [0018] - paragraph [0023]; figure 2 -----	1,15,21, 34

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No
PCT/US2013/055086

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2012102491 A1	26-04-2012	NONE	
US 7502884 B1	10-03-2009	US 7502884 B1	10-03-2009
		US 7634650 B1	15-12-2009
		US 7937447 B1	03-05-2011
		US 8041875 B1	18-10-2011
		US 8180949 B1	15-05-2012
		US 8291148 B1	16-10-2012
		US 2013145072 A1	06-06-2013
US 2012042095 A1	16-02-2012	NONE	