management module of the host after the first virtual machine is successfully created, a VLAN subinterface creation request sent by a cloud management platform, where the VLAN subinterface creation request carries the VLAN identifier of the VF that is allocated to the first virtual machine; and sending, by the device management module, a notification message to the VMM, so that the VMM creates the VLAN subinterface corresponding to the VF of the first virtual machine, where the VLAN subinterface corresponding to the VF of the first virtual machine has a same VLAN identifier as the VF of the first virtual machine.

[0016] According to a third aspect, an embodiment of the present invention provides a cloud computing system. The cloud computing system includes a cloud management platform and the host described in the first aspect. The cloud management platform is configured to create a first virtual machine on the host, and after successfully creating the first virtual machine, instruct a VMM of the host to create a VLAN subinterface corresponding to a VF of the first virtual machine. The VLAN subinterface corresponding to the VF of the first virtual machine has a same VLAN identifier as the VF of the first virtual machine.

[0017] According to a fourth aspect, corresponding to the foregoing apparatus, method, and system, an embodiment of the present invention provides a host. The host has a function of implementing the host defined in the first aspect. The function may be implemented by hardware, or may be implemented by executing corresponding software by hardware. The hardware or software includes one or more modules corresponding to the foregoing function. Specifically, the host includes a first processor, a first memory, and at least one network interface card. The network interface card includes a second processor, a second memory, and at least two network ports, a first network port and a second network port in the at least one network interface card support a network interface card virtualization capability, the first network port is corresponding to at least one PF and multiple VFs, VLAN identifiers are configured for the multiple VFs and the VLAN identifiers of the VFs are different from each other, the first memory and the second memory store instructions, the first processor executes a first instruction in the first memory to implement a function of a first virtual machine, the first virtual machine is connected to at least one VF of the first network port, the first processor executes a second instruction in the first memory to implement a function of a VLAN subinterface, the first processor executes a third instruction in the first memory to

by a general-purpose or dedicated server. The server includes a processor, a memory, a system bus, and an input/output interface. The processor is configured to support a corresponding function of the host/the cloud management platform in the system. The input/output interface is configured to communicate with another component in the cloud computing system. The processor executes the instructions stored in the memories.

[0019] According to a fifth aspect, an embodiment of the present invention provides a computer storage medium that is configured to store a computer software instruction used by the foregoing host. The computer software instruction includes a program designed for executing the foregoing aspects.

[0020] According to a sixth aspect, an embodiment of the present invention provides another computer storage medium that is configured to store a computer software instruction used by the switching equipment in the foregoing network interface card. The computer software instruction includes a program designed for executing the foregoing aspects.

[0021] According to a seventh aspect, an embodiment of the present invention provides a computer program. When a physical server runs the computer program, the physical server performs functions of a virtual machine, a VLAN subinterface, and a virtual network function module in a host.

[0022] According to an eighth aspect, an embodiment of the present invention provides a computer program. When a processor or a programmable logic circuit in a network interface card runs the computer program, the network interface card performs a function of the switching equipment in the foregoing aspects.

[0023] In conclusion, a virtual machine sends a data packet from a VF that is connected to the virtual machine, switching equipment of a first network port forcibly forwards the data packet to a second network port, switching equipment of the second network port sends the data packet to a VLAN subinterface on a VMM according to a VLAN identifier carried in the data packet, the VLAN subinterface transfers the data packet to a virtual network function module, and the virtual network function module provides abundant network function processing for the data packet. Based on network interface card passthrough, after sending a data packet from a passthrough VF, a virtual machine may send the data packet to a virtual network function module on a VMM in the foregoing manner. Abundant network functions

embodiments of the present invention more clearly, and do not constitute any limitation on the technical solutions provided in the embodiments of the present invention. A person of ordinary skill in the art may learn that, with the evolution of network architectures and the emergence of a new service scenario, the technical solutions provided in the embodiments of the present invention are also applicable for similar technical issues.

[0033] As shown in FIG. 1A, FIG. 1A is a schematic diagram of a virtualized structure on a host according to an embodiment of the present invention. The host is a physical server. A bottom layer of the physical server is a hardware layer. The hardware layer mainly includes hardware resources such as a central processing unit (CPU, Central Processing Unit), a memory, a hard disk, and a network interface card. Server virtualization implements a virtualized running environment of multiple virtual machines (Virtual Machine, VM) on the physical server by using virtualization software (for example, VMWare ESX or Citrix XEN). A software layer that is installed on the server to implement a virtualized environment is referred to as a virtual machine monitor (Virtual Machine Monitor, VMM). The VMM running on the hardware layer is responsible for scheduling, allocating, and managing the hardware resources at the hardware layer. Multiple virtual machines VMs run on the VMM. The VMM provides each virtual machine with a hardware environment such as a virtualized CPU, memory, storage, IO device (such as a network interface card), and Ethernet switch, to ensure that the multiple virtual machines run in a manner of isolating from each other.

[0034] In a virtualized running environment, the VMM creates a virtual network interface card (Virtual Network Interface Card, vNIC) for each virtual machine, and a virtual switch vSwitch provides a capability of communication between the virtual machines and between the virtual machines and an external network. The virtual network interface card of each virtual machine is corresponding to a logical port of the vSwitch. A physical network interface card of the host is corresponding to a port connecting the vSwitch to an external physical switch. When a data packet sent or received by a virtual machine passes through the VMM, a virtual network function module on the VMM performs network function processing on the data packet that passes through the VMM. The virtual network function module is a software module and can be updated as required. Therefore, the virtual network function module on the VMM can provide abundant network functions for a user.

physical resource of the physical network interface card (for example, a port of the network interface card).

[0039] When a virtual machine on the host sends a data packet by using a passthrough VF, the data packet sent by the virtual machine is directly transferred to the VF of the network interface card by using a VF driver installed on the virtual machine. As a result, the data packet cannot pass through a virtual network function module on a VMM. Switching equipment of the network interface card has a limited MAC table capacity and a limited processing capability, and cannot perform unconstrained function extension. Therefore, the network interface card cannot provide abundant network functions. In a possible scenario, the switching equipment is a virtual Ethernet bridge & classifier.

[0040] It should be noted that this embodiment of the present invention is based on the network interface card passthrough scenario. The foregoing network interface card passthrough may be implemented by means of SR-IOV or MR-IOV, and this embodiment of the present invention imposes no limitation thereon.

[0041] As shown in FIG. 2, FIG. 2 is a schematic diagram of another virtualized architecture on a host according to an embodiment of the present invention. The host includes a VMM and at least one physical network interface card. At least one virtual machine runs on the host. The at least one physical network interface card includes at least three network ports. A first network port and a second network port support a network interface card virtualization capability. A third network port is connected to an external physical switch of the host, and is responsible for sending and receiving cross-host network traffic. The first network port is virtualized into at least one PF and at least two VFs (in FIG. 2, two VFs are used as an example for description). A VF driver is installed on a first virtual machine and a second virtual machine. The first virtual machine and the second virtual machine are separately connected to at least one VF. The first network port and the second network port are directly connected by using a network cable. A PF of the second network port is connected to a VLAN subinterface on the VMM. Therefore, all traffic sent from a virtual machine is detoured to the second network port by using the network cable, and the second network port directs the traffic back to a virtual network function module on the VMM. The network cable may be specifically a cable of any medium, including a twisted pair, an optical fiber, a

network function processing on the data packet.

[0049] In the foregoing process in which the first virtual machine sends the data packet to the second virtual machine, the VLAN identifier of the VF connected to the first virtual machine is the same as a VLAN identifier of the corresponding VLAN subinterface in the VMM. As shown in FIG. 3, FIG. 3 is a schematic diagram of a virtual machine creation process according to an embodiment of the present invention. The process is used to create a virtual machine with a passthrough VF and create a VLAN subinterface corresponding to the passthrough VF.

[0050] Step 301: A computing management module receives a virtual machine creation request, where the virtual machine creation request is used to create a virtual machine with a passthrough VF.

[0051] In a scenario, a virtual machine creation process may be initiated by an administrator or a user. The administrator or the user logs in to, by using a terminal, an externally presented interface of a cloud management platform, selects a specification of a to-be-created virtual machine, and initiates a virtual machine creation request to the computing management module. The virtual machine creation request carries a parameter of the to-be-created virtual machine, and the parameter includes information indicating that the to-be-created virtual machine has a passthrough VF.

[0052] Step 302: The computing management module sends an allocation request to a device management module, to request to allocate a VF to the to-be-created virtual machine.

[0053] Step 303: The device management module returns an identifier of an idle VF to the computing management module, where the idle VF is a VF that is on a host and that is not allocated to a virtual machine.

[0054] Step 304: The computing management module allocates a VLAN identifier to the VF.

[0055] Step 305: The computing management module sends the VLAN identifier to the device management module, and the device management module configures the VLAN identifier for the allocated VF.

[0056] Step 306: The device management module sends information for virtual machine creation to a VMM, where the information for virtual machine creation includes the VLAN

identifier of the VF.

[0057] Step 306: The VMM creates the virtual machine, and sets the VF as a passthrough network port of the virtual machine.

[0058] Step 307: After the virtual machine is successfully created, the computing management module sends a VLAN subinterface creation request to the device management module, where the VLAN subinterface creation request carries the VLAN identifier of the VF.

[0059] Step 308: The device management module receives the VLAN subinterface creation request, creates, on the VMM, a VLAN subinterface corresponding to the VF, and configures a VLAN identifier of the VLAN subinterface to be the same as that of the VF.

[0060] It should be noted that the VLAN subinterface is a virtual network device provided by a linux system, and may be created directly according to an ip command of the Linux system. For example, a command for creating a VLAN subinterface whose VLAN identifier is 100 is as follows: ip link add link eth0 name vlan100 type vlan id 100.

[0061] Step 309: The computing management module receives a VLAN subinterface creation response message, sends a notification message to a network management module, and connects the created VLAN subinterface to a bridge corresponding to the etwork management module.

[0062] It should be noted that a virtual machine may have multiple passthrough VFs. A specific creation process is similar to the foregoing steps, but different VLAN identifiers are set for passthrough VFs and VLAN subinterfaces in a one-to-one correspondence with the passthrough VFs are created on the VMM.

[0063] As shown in FIG. 4, FIG. 4 is a schematic flowchart of a packet processing method in a cloud computing system according to an embodiment of the present invention. In this embodiment of the present invention, an example in which a first virtual machine on a host A communicates with a second virtual machine on a host B is used for description.

[0064] Step 401: The first virtual machine on the host A sends a data packet by using a VF that is connected to the first virtual machine, where a destination address of the data packet is an address of the second virtual machine, and the data packet carries a VLAN identifier of the VF that sends the data packet.

[0065] Step 402: Switching equipment (first switching equipment) of a passthrough

network port receives the data packet, and broadcasts the received data packet. VLAN identifiers of VFs on the passthrough network port are different from each other and the VFs are located on different virtual local area networks. Therefore, other VFs on the passthrough network port do not receive the broadcast data packet, and the data packet is forcibly sent to an alternative network port by using a network cable.

[0066] It should be noted that the passthrough network port and the alternative network port are in a virtual Ethernet port aggregator (Virtual Ethernet Port Aggregator, VEPA) mode. The VEPA mode is used to instruct the passthrough network port and the alternative network port to forcibly forward the received data packet. The VEPA mode is defined in the IEEE802.1Qbg standard.

[0067] Step 403: Switching equipment (second switching equipment) of an alternative network port receives the data packet from the passthrough network port, and sends, according to the VLAN identifier carried in the data packet, the data packet to a VLAN subinterface that is on a VMM and that has a same VLAN identifier as the data packet.

[0068] Step 404: The VLAN subinterface receives the data packet, removes the VLAN identifier in the data packet, and sends the data packet to a first virtual network function module.

[0069] Step 405: The first virtual network function module performs network function processing on the data packet, and sends the data packet to a service network port. The service network port of the host A sends the received data packet to an external network of the host. The data packet is routed to a service network port of the host B on which the second virtual machine is located, and the service network port of the host B sends the data packet to a second virtual network function module of the host B after receiving the data packet.

[0070] Specifically, a tunnel may be established between the first virtual network function module of the host A and the second virtual network function module of the host B. By using a tunnel technology, the data packet is transferred to the virtual network function module of the host B by using a network between the host A and the host B.

[0071] Step 406: After performing network function processing, the second virtual network function module of the host B sends the data packet to a VLAN subinterface corresponding to a VF that is connected to the second virtual machine, where the VF that is

connected to the second virtual machine has a same VLAN identifier as the VLAN subinterface receiving the data packet.

[0072] It should be noted that, in a possible scenario, the VLAN subinterface is connected to a virtual bridge implemented by software, and the virtual bridge provides a layer 2 forwarding capability to send the data packet to the VLAN subinterface. A same VLAN identifier is set for the VLAN subinterface and the corresponding VF. Therefore, a VF that is provided for the second virtual machine in a passthrough manner may receive the data packet.

[0073] Step 407: The VLAN subinterface adds a VLAN identifier to the data packet, where the VLAN identifier is a VLAN identifier of the VLAN subinterface; and sends the data packet to the alternative network port. Because VLAN subinterfaces connected to the alternative network port have different VLAN identifiers, switching equipment (fourth switching equipment) of the alternative network port forcibly forwards the data packet to the passthrough network port by using the network cable.

[0074] Step 408: Switching equipment (third switching equipment) of the passthrough network port sends, according to the VLAN identifier carried in the data packet, the data packet to the VF identified by the VLAN identifier.

[0075] Step 409: The VF removes the VLAN identifier in the data packet, and sends the data packet to the second virtual machine.

[0076] It should be noted that, in this embodiment of the present invention, the passthrough network port, the alternative network port, and the service network port may be located on a same network interface card, or may be independently located on different network interface cards. This embodiment of the present invention imposes no limitation thereon. Further, when the passthrough network port and the alternative network port are located on a same network interface card, each of the passthrough network port and the alternative network port may have independent switching equipment, or the passthrough network port and the alternative network port share same switching equipment.

[0077] It should be noted that, in a specific implementation scenario, the VLAN subinterface in this embodiment of the present invention may be an Open vSwitch.

[0078] In this embodiment of the present invention, VFs of a passthrough network port have different VLAN identifiers. VLAN subinterfaces in one-to-one correspondence with the

VFs of the passthrough network port are set on a VMM. Each VLAN subinterface has a same VLAN identifier as a corresponding VF. After a virtual machine sends a data packet from a VF, the data packet is forcibly forwarded to switching equipment of an alternative network port because the passthrough network port and the alternative network port are directly connected by using a network cable, and the data packet carries a VLAN identifier of the VF. The switching equipment of the alternative network port sends, according to the VLAN identifier carried in the data packet, the data packet to a VLAN subinterface that has a same VLAN identifier as the data packet, so that the data packet is sent to the VMM. Further, a virtual network function module on the VMM performs network function processing and sends the processed data packet to a second virtual machine. In a network interface card passthrough scenario, a data packet is sent to a virtual network function module on a VMM in the foregoing manner. By using software module, network function flexibility is implemented and abundant network functions are provided.

[0079] The embodiment corresponding to FIG. 4 provides a process of transmitting a data packet between two virtual machines on two hosts. A person skilled in the art may understand that a source virtual machine and a destination virtual machine of the data packet may be located on a same host. In this case, after performing network function processing on the data packet, a virtual network function module on the host sends the processed data packet to a VLAN subinterface corresponding to a VF that is connected to a second virtual machine. The VLAN subinterface sends the data packet to the second virtual machine.

[0080] The host and a cloud management platform may use a general-purpose computer device. For example,

[0081] As shown in FIG. 5, FIG. 5 is a schematic structural diagram of hardware of a computer device according to an embodiment of the present invention. The computer device 500 includes at least one processor 501, a communications bus 502, a memory 503, and at least one communications interface 504.

[0082] The processor 501 may be a general-purpose central processing unit (CPU), a microprocessor, an application-specific integrated circuit (application-specific integrated circuit, ASIC), or one or more integrated circuits configured to control program execution in the solutions of the present invention.

[0083] The communications bus 502 may include a path for transferring information between the foregoing components. The communications interface 504 uses an apparatus such as a transceiver to communicate with another device or a communications network such as an Ethernet, a radio access network (RAN), or a wireless local area network (Wireless Local Area Network, WLAN).

[0084] The memory 503 may be a read-only memory (read-only memory, ROM) or another type of static storage device that can store static information and instructions, a random access memory (random access memory, RAM) or another type of dynamic storage device that can store information and instructions, an electrically erasable programmable read-only memory (Electrically Erasable Programmable Read-Only Memory, EEPROM), a compact disc read-only memory (Compact Disc Read-Only Memory, CD-ROM) or another compact disc storage, an optical disc storage (including a compact disc, a laser disc, an optical disc, a digital versatile disc, a Blu-ray disc, or the like), a magnetic disk storage medium or another magnetic storage device, or any other medium that can be used to carry or store expected program code in a form of an instruction or a data structure and that can be accessed by a computer. No limitation is imposed thereon. The memory may exist alone and is connected to the processor by using the bus. Alternatively, the memory may be integrated with the processor.

[0085] The memory 503 is configured to store application program code for executing the solutions in the present invention, and the processor 501 controls the execution. The processor 501 is configured to execute the application program code stored in the memory 503.

[0086] In a specific implementation, in an embodiment, the processor 501 may include one or more CPUs, for example, a CPU0 and a CPU1 in FIG. 5.

[0087] In a specific implementation, in an embodiment, the computer device 500 may include multiple processors, for example, the processor 501 and a processor 508 in FIG. 5. Each of the processors may be a single-core processor, or may be a multi-core processor. The processor herein may be one or more devices, circuits, and/or processing cores that are configured to process data (for example, a computer program instruction).

[0088] In a specific implementation, in an embodiment, the computer device 500 may

computing system according to an embodiment of the present invention. The cloud computing system includes at least one host 601 and a cloud management platform 602. A structure of the host 601 is shown in FIG. 2.

[0094] The host 601 includes a virtual machine monitor VMM and at least one network interface card. A first virtual machine runs on the host 601. The VMM includes multiple VLAN subinterfaces and a virtual network function module, and the at least one network interface card includes switching equipment and at least three network ports. A first network port and a second network port support a network interface card virtualization capability. The first network port is corresponding to at least one PF and multiple VFs. VLAN identifiers are configured for the multiple VFs and the VLAN identifiers of the VFs are different from each other. The first virtual machine is connected to at least one VF of the first network port. A quantity of VLAN subinterfaces is the same as a quantity of VFs of the first network port and the VLAN subinterfaces are in a one-to-one correspondence with the VFs of the first network port. The VLAN subinterfaces have same VLAN identifiers as the VFs corresponding to the VLAN subinterfaces. The first network port and the second network port are connected by using a network cable.

[0095] The cloud management platform 602 is configured to create the first virtual machine on the host, and after successfully creating the first virtual machine, instruct the VMM of the host to create a VLAN subinterface corresponding to the VF of the first virtual machine. The VLAN subinterface corresponding to the VF of the first virtual machine has a same VLAN identifier as the VF of the first virtual machine.

[0096] The first virtual machine is configured to send a data packet to a second virtual machine by using the VF that is connected to the first virtual machine. The data packet carries a VLAN identifier of the VF that sends the data packet, and an address of the second virtual machine.

[0097] Switching equipment of the first network port is configured to receive the data packet, and forcibly forward the data packet to the second network port by using the network cable.

[0098] Switching equipment of the second network port is configured to receive the data packet from the first network port, and send, according to the VLAN identifier carried in the

invention may be provided as a method, an apparatus (device), or a computer program product. Therefore, the present invention may use a form of hardware only embodiments, software only embodiments, or embodiments with a combination of software and hardware. Moreover, the embodiments of the present invention may use a form of a computer program product that is implemented on one or more computer-usable storage media (including but not limited to a disk memory, a CD-ROM, an optical memory, and the like) that include computer-usable program code. The computer program is stored/distributed in a proper medium and is provided as or used as a part of the hardware together with another hardware, or may also use another allocation form, such as by using the Internet or another wired or wireless telecommunications system.

[0105] The computer program instructions in the embodiments of the present invention may be stored in a computer readable memory that can instruct a computer or another programmable data processing device to work in a specific manner. The computer program instructions may be executed to implement functions of components in the foregoing embodiments.

[0106] These computer program instructions may also be loaded onto a computer or another programmable data processing device, so that a series of operations and steps are performed on the computer or the another programmable device, thereby generating computer-implemented processing. Therefore, the instructions executed on the computer or the another programmable device provide steps for implementing a specific function in one or more processes in the flowcharts and/or in one or more blocks in the block diagrams.

[0107] Although the present invention is described with reference to specific features and the embodiments thereof, obviously, various modifications and combinations may be made to them without departing from the scope of the present invention. Correspondingly, the specification and accompanying drawings are merely illustrative description of the present invention defined by the accompanying claims, and are considered as any of or all modifications, variations, combinations or equivalents that cover the scope of the present invention. Obviously, a person skilled in the art can make various modifications and variations to the present invention without departing from the scope of the present invention. The present invention is intended to cover these modifications and variations

CLAIMS

1. A host, wherein the host comprises a virtual machine monitor, VMM, and at least one network interface card, a first virtual machine runs on the host, the VMM comprises multiple VLAN subinterfaces and a virtual network function module, and the at least one network interface card comprises switching equipment and at least three network ports, wherein a first network port and a second network port support a network interface card virtualization capability, the first network port is corresponding to at least one physical function, PF, and multiple virtual functions, VFs, virtual local area network, VLAN, identifiers are configured for the multiple VFs and the VLAN identifiers of the VFs are different from each other, the first virtual machine is connected to at least one VF of the first network port, a quantity of VLAN subinterfaces is the same as a quantity of VFs of the first network port and the VLAN subinterfaces are in a one-to-one correspondence with the VFs of the first network port, the VLAN subinterfaces have same VLAN identifiers as the VFs corresponding to the VLAN subinterfaces, and the first network port and the second network port are connected by using a network cable;

the first virtual machine is configured to send a data packet to a second virtual machine by using the VF that is connected to the first virtual machine, wherein the data packet carries a VLAN identifier of the VF that sends the data packet, and an address of the second virtual machine;

switching equipment of the first network port is configured to receive the data packet, and forcibly forward the data packet to the second network port by using the network cable;

switching equipment of the second network port is configured to receive the data packet from the first network port, and send, according to the VLAN identifier carried in the data packet, the data packet to a VLAN subinterface identified by the VLAN identifier;

the VLAN subinterface is configured to receive the data packet, remove the VLAN identifier in the data packet, and send the data packet to the virtual network function module; and

the virtual network function module is configured to perform network function processing on the changed data packet, and send the processed data packet, wherein a destination address of the processed data packet is the address of the second virtual machine.

2. The host according to claim 1, wherein the first network port and the second network port are in a virtual Ethernet port aggregator, VEPA, mode.

3. The host according to claim 1 or 2, wherein the host further comprises a device management module, wherein the device management module is configured to: after the first virtual machine is successfully created, receive a VLAN subinterface creation request sent by a cloud management platform, wherein the VLAN subinterface creation request carries the VLAN identifier of the VF that is allocated to the first virtual machine; and
the device management module is further configured to send a notification message to the VMM to instruct the VMM to create the VLAN subinterface corresponding to the VF of the first virtual machine, wherein the VLAN subinterface corresponding to the VF of the first virtual machine has a same VLAN identifier as the VF of the first virtual machine.

4. The host according to any one of claims 1 to 3, wherein: the first network port, the second network port, and a third network port are located on a same network interface card; or
the first network port, the second network port, and a third network port are located on two or three network interface cards; or
the first network port and the second network port are located on a same network interface card, and the first network port and the second network port share switching equipment of the same network interface card; or
the first network port and the second network port are located on a same network interface card, and the switching equipment of the first network port is independent of the switching equipment of the second network port.

5. The host according to any one of claims 1 to 4, wherein the second virtual machine and the first virtual machine are located on a same host.

6. The host according to claim 5, wherein: the virtual network function module is further configured to send the processed data packet to a VLAN subinterface corresponding to a VF that is connected

to the second virtual machine, wherein the VF that is connected to the second virtual machine has a same VLAN identifier as the VLAN subinterface receiving the processed data packet;

the VLAN subinterface corresponding to the VF that is connected to the second virtual machine is configured to add a VLAN identifier of the VLAN subinterface to the data packet, and send the data packet to the second network port;

the switching equipment of the second network port is further configured to forcibly forward the data packet to the first network port by using the network cable; and

the switching equipment of the first network port is further configured to send, according to the VLAN identifier carried in the data packet, the data packet to the VF identified by the VLAN identifier in the data packet, so that the data packet is transmitted to the second virtual machine.

7. The host according to any one of claims 1 to 4, wherein the second virtual machine and the first virtual machine are located on different hosts; and

the virtual network function module is specifically configured to establish a tunnel between the virtual network function module and another virtual network function module on a host on which the second virtual machine is located, send the processed data packet to an external physical switch by using the third network port, and send, by using the external physical switch, the processed data packet to the host on which the second virtual machine is located, so that the another virtual network function module sends the processed data packet to the second virtual machine.

8. A packet processing method in a cloud computing system, wherein at least one host in the cloud computing system comprises a virtual machine monitor, VMM, and at least one network interface card, a first virtual machine runs on the host, the VMM comprises multiple VLAN subinterfaces and a virtual network function module, and the at least one network interface card comprises switching equipment and at least three network ports, wherein a first network port and a second network port support a network interface card virtualization capability, the first network port is corresponding to at least one PF and multiple VFs, VLAN identifiers are configured for the multiple VFs and the VLAN identifiers of the VFs are different from each other, the first virtual machine is connected to at least one VF of the first network port, a quantity of VLAN subinterfaces is the same as a quantity of VFs of the first network port and the VLAN subinterfaces are in a one-to-one correspondence with the VFs of

the first network port, the VLAN subinterfaces have same VLAN identifiers as the VFs corresponding to the VLAN subinterfaces, and the first network port and the second network port are connected by using a network cable; and the method comprises:

sending (401), by the first virtual machine, a data packet to a second virtual machine by using the VF that is connected to the first virtual machine, wherein the data packet carries a VLAN identifier of the VF that sends the data packet, and an address of the second virtual machine;

receiving, by switching equipment of the first network port, the data packet, and forcibly forwarding the data packet to the second network port by using the network cable;

receiving, by switching equipment of the second network port, the data packet from the first network port, and sending, according to the VLAN identifier carried in the data packet, the data packet to a VLAN subinterface identified by the VLAN identifier;

receiving, by the VLAN subinterface, the data packet, removing the VLAN identifier in the data packet, and sending the data packet to the virtual network function module; and

performing (406), by the virtual network function module, network function processing on the changed data packet, and sending the processed data packet, wherein a destination address of the processed data packet is the address of the second virtual machine.

9. The method according to claim 8, wherein the first network port and the second network port are in a VEPA mode.

10. The method according to claim 8 or 9, wherein the host further comprises a device management module, and the method further comprises:

receiving, by the device management module after the first virtual machine is successfully created, a VLAN subinterface creation request sent by a cloud management platform, wherein the VLAN subinterface creation request carries the VLAN identifier of the VF that is allocated to the first virtual machine; and

sending, by the device management module, a notification message to the VMM, so that the VMM creates the VLAN subinterface corresponding to the VF of the first virtual machine, wherein the VLAN subinterface corresponding to the VF of the first virtual machine has a same VLAN identifier as the VF of the first virtual machine.

11. The method according to any one of claims 8 to 10, wherein:

the first network port, the second network port, and a third network port are located on a same network interface card; or

the first network port, the second network port, and a third network port are located on two or three network interface cards; or

the first network port and the second network port are located on a same network interface card, and the first network port and the second network port share switching equipment of the same network interface card; or

the first network port and the second network port are located on a same network interface card, and the switching equipment of the first network port is independent of the switching equipment of the second network port.

12. The method according to any one of claims 8 to 11, wherein the second virtual machine and the first virtual machine are located on a same host.

13. The method according to claim 12, wherein the sending the processed data packet to the second virtual machine comprises:

sending, by the virtual network function module, the processed data packet to a VLAN subinterface corresponding to a VF that is connected to the second virtual machine, wherein the VF that is connected to the second virtual machine has a same VLAN identifier as the VLAN subinterface receiving the processed data packet;

adding (407), by the VLAN subinterface corresponding to the VF that is connected to the second virtual machine, a VLAN identifier of the VLAN subinterface to the data packet, and sending the processed data packet to the second network port;

forcibly forwarding, by the switching equipment of the second network port, the processed data packet to the first network port by using the network cable; and

sending, by the switching equipment of the first network port according to the VLAN identifier carried in the processed data packet, the data packet to the VF identified by the VLAN identifier in the data packet, so that the data packet is transmitted to the second virtual machine.

14. The method according to any one of claims 8 to 11, wherein the sending, by the virtual network function module, the processed data packet further comprises:

establishing, by the virtual network function module, a tunnel between the virtual network function module and another virtual network function module on a host on which the second virtual machine is located, sending the processed data packet to an external physical switch by using the third network port, and sending, by using the external physical switch, the processed data packet to the host on which the second virtual machine is located, so that the another virtual network function module sends the processed data packet to the second virtual machine.

15. A cloud computing system, comprising a cloud management platform(602) and at least one host (601), wherein the host comprises a virtual machine monitor VMM and at least one network interface card, a first virtual machine runs on the host, the VMM comprises multiple VLAN subinterfaces and a virtual network function module, and the at least one network interface card comprises switching equipment and at least three network ports, wherein a first network port and a second network port support a network interface card virtualization capability, the first network port is corresponding to at least one PF and multiple VFs, VLAN identifiers are configured for the multiple VFs and the VLAN identifiers of the VFs are different from each other, the first virtual machine is connected to at least one VF of the first network port, a quantity of VLAN subinterfaces is the same as a quantity of VFs of the first network port and the VLAN subinterfaces are in a one-to-one correspondence with the VFs of the first network port, the VLAN subinterfaces have same VLAN identifiers as the VFs corresponding to the VLAN subinterfaces, and the first network port and the second network port are connected by using a network cable;

the cloud management platform is configured to create the first virtual machine on the host, and after successfully creating the first virtual machine, instruct the VMM of the host to create a VLAN subinterface corresponding to the VF of the first virtual machine, wherein the VLAN subinterface corresponding to the VF of the first virtual machine has a same VLAN identifier as the VF of the first virtual machine;

the first virtual machine is configured to send a data packet to a second virtual machine by using the VF that is connected to the first virtual machine, wherein the data packet carries a VLAN identifier of the VF that sends the data packet, and an address of the second virtual machine;

switching equipment of the first network port is configured to receive the data packet, and forcibly forward the data packet to the second network port by using the network cable;

switching equipment of the second network port is configured to receive the data packet from the first network port, and send, according to the VLAN identifier carried in the data packet, the data packet to the VLAN subinterface that has a same VLAN identifier as the data packet;

the VLAN subinterface is configured to receive the data packet, remove the VLAN identifier in the data packet, and send the data packet to the virtual network function module; and

the virtual network function module is configured to perform network function processing on the changed data packet, and send the processed data packet, wherein a destination address of the processed data packet is the address of the second virtual machine.

16. The system according to claim 15, wherein the first network port and the second network port are in a VEPA mode.

17. The system according to claim 15 or 16, wherein the host further comprises a device management module, wherein the device management module is configured to: after the first virtual machine is successfully created, receive a VLAN subinterface creation request sent by the cloud management platform, wherein the VLAN subinterface creation request carries the VLAN identifier of the VF that is allocated to the first virtual machine; and

the device management module is further configured to send a notification message to the VMM to instruct the VMM to create the VLAN subinterface corresponding to the VF of the first virtual machine, wherein the VLAN subinterface corresponding to the VF of the first virtual machine has a same VLAN identifier as the VF of the first virtual machine.

18. The system according to any one of claims 15 to 17, wherein the second virtual machine and the first virtual machine are located on a same host;

the virtual network function module is further configured to send the processed data packet to a VLAN subinterface corresponding to a VF that is connected to the second virtual machine, wherein the VF that is connected to the second virtual machine has a same VLAN identifier as the VLAN subinterface receiving the processed data packet;

the VLAN subinterface corresponding to the VF that is connected to the second virtual machine is configured to add a VLAN identifier of the VLAN subinterface to the data packet, and send the data packet to the second network port;

the switching equipment of the second network port is further configured to forcibly forward the data packet to the first network port by using the network cable; and

the switching equipment of the first network port is further configured to send, according to the VLAN identifier carried in the data packet, the data packet to the VF identified by the VLAN identifier in the data packet, so that the data packet is transmitted to the second virtual machine.

19. The system according to any one of claims 15 to 17, wherein the second virtual machine and the first virtual machine are located on different hosts; and

the virtual network function module is specifically configured to establish a tunnel between the virtual network function module and another virtual network function module on a host on which the second virtual machine is located, send the processed data packet to an external physical switch by using a third network port, and send, by using the external physical switch, the processed data packet to the host on which the second virtual machine is located, so that the another virtual network function module sends the processed data packet to the second virtual machine.