SECURITY SCANNING METHOD FOR CLOUD COMPUTING, RELATED DEVICE, AND STORAGE MEDIUM

[0001] The present disclosure claims priority to Chinese Patent Application No. 202311296787.6, filed with the China National Intellectual Property Administration on October 8, 2023 and entitled "SECURITY SCANNING METHOD FOR CLOUD COMPUTING, RELATED DEVICE, AND STORAGE MEDIUM", which is incorporated herein by reference in its entirety.

TECHNICAL FIELD

[0002] The present disclosure relates to the field of cloud computing technologies, and in particular, to a security scanning method for cloud computing, a related device, and a storage medium.

BACKGROUND

[0003] As cloud computing technologies develop rapidly, a cloud computing instance in a form of a virtual machine (VM) has become one of core technologies of modern data centers. As a quantity of cloud computing instances increases, host security becomes an important scenario for cloud security confrontation. In host security, scanning a disk of a host is an important information collection means for detecting whether the cloud computing instance has an attack behavior against the host, collecting a malicious sample, and reconstructing an attack path.

[0004] A disk scanning manner in the industry mainly includes an in-band agent scanning manner, to be specific, an agent program for disk scanning is embedded within the cloud computing instance to complete a disk scanning operation, which has an advantage of high scanning efficiency. However, because the agent program for disk scanning runs in the same computing environment as a process, a kernel, and the like of the cloud computing instance, the agent program is easily affected by an intrusion behavior, and may also occupy computing resources originally allocated to the cloud computing instance.

SUMMARY

[0005] Various aspects of the present disclosure provide a security scanning method for cloud computing, a related device, and a storage medium, so that a security scanning program

is not susceptible to intrusion or interference from an internal process of a virtualization instance, and does not occupy computing resources of the virtualization instance.

[0006] An embodiment of the present disclosure provides a host device. A virtualization manager and a first virtualization instance managed by the virtualization manager are deployed on the host device, the first virtualization instance has target physical storage space, and the target physical storage space supports the virtualization manager to perform a read/write operation on the target physical storage space in a multi-queue read/write manner; the virtualization manager is further configured to: in a case that determining that the first virtualization instance uses a security scanning service, create a shared queue and a shared memory for a security scanning program located external to the first virtualization instance; and provide target data in the target physical storage space to the security scanning program based on the shared queue and the shared memory, for the security scanning program to perform security scanning on the host device based on the target data; where the first virtualization instance performs the read/write operation on the target physical storage space through at least one storage queue, and the shared queue is different from the at least one storage queue.

[0007] An embodiment of the present disclosure further provides a security scanning method for cloud computing, applied to a virtualization manager on a host device. The virtualization manager manages a first virtualization instance on the host device. The method includes: in a case that the first virtualization instance uses a security scanning service, creating a shared queue and a shared memory for a security scanning program located external to the first virtualization instance, where the first virtualization instance has target physical storage space, and the target physical storage space supports the virtualization manager to perform a read/write operation on the target physical storage space in a multi-queue read/write manner; and providing target data in the target physical storage space to the security scanning program based on the shared queue and the shared memory, for the security scanning program to perform security scanning on the host device based on the target data, where the first virtualization instance performs the read/write operation on the target physical storage space through at least one storage queue, and the shared queue is different from the at least one storage queue.

[0008] An embodiment of the present disclosure further provides a security scanning method for cloud computing, applied to a security scanning program external to a first virtualization instance. The method includes: in a case that a virtualization manager corresponding to the first virtualization instance creates a shared queue and a shared memory for the security scanning program, writing a read request into the shared queue, for the virtualization manager to read target data from target physical storage space of the first

virtualization instance based on the read request, and writing the target data into the shared memory; and reading the target data from the shared memory, and performing security scanning based on the target data, where the target physical storage space supports the virtualization manager to perform a read/write operation on the target physical storage space in a multi-queue read/write manner, the first virtualization instance performs the read/write operation on the target physical storage space through at least one storage queue, and the shared queue is different from the at least one storage queue.

[0009] An embodiment of the present disclosure further provides a virtualization management apparatus, implementable as a virtualization manager on a host device, and configured to manage a first virtualization instance. The apparatus includes: a creation module and a providing module. The creation module is configured to: in a case that the first virtualization instance uses a security scanning service, create a shared queue and a shared memory for a security scanning program located external to the first virtualization instance, where the first virtualization instance has target physical storage space, and the target physical storage space supports the virtualization manager to perform a read/write operation on the target physical storage space in a multi-queue read/write manner. The providing module is configured to provide target data in the target physical storage space to the security scanning program based on the shared queue and the shared memory, for the security scanning program to perform security scanning on the host device based on the target data, where the first virtualization instance performs the read/write operation on the target physical storage space through at least one storage queue, and the shared queue is different from the at least one storage queue.

[0010] An embodiment of the present disclosure further provides a host device, including a memory and a processor. The memory stores a security scanning program, and the processor is coupled to the memory and configured to execute the security scanning program, to implement steps in the security scanning method for cloud computing provided by the embodiments of the present disclosure.

[0011] An embodiment of the present disclosure further provides a computer-readable storage medium storing a computer program. When the computer program is executed by a processor, the processor is caused to implement steps in the security scanning method for cloud computing provided by the embodiments of the present disclosure.

[0012] An embodiment of the present disclosure provides a computer program product, including a computer program. When the computer program is executed by a processor, steps in the foregoing security scanning method for cloud computing are implemented.

[0013] In the embodiments of the present disclosure, physical storage space of the virtualization instance supports the virtualization manager to perform the read/write operation on the physical storage space in the multi-queue read/write manner. Based on this, functionality extension is performed on the virtualization manager, so that in addition to creating, for the virtualization instance, a storage queue required for performing the read/write operation on the physical storage space, the virtualization manager can further create the shared queue and the shared memory to the security scanning program located external to the virtualization instance, and further provide data in the physical storage space to the security scanning program based on the shared queue and the shared memory, for the security scanning program to perform host security scanning based on the data in the physical storage space of the virtualization instance. Because the security scanning program is located external to the virtualization instance, the security scanning program is not susceptible to intrusion or interference from an internal process of the virtualization instance, and does not occupy computing resources of the virtualization instance. In addition, host security scanning is performed based on the data in the physical storage space of the virtualization instance, so that real-time and highly efficient scanning can be implemented, and an immediately occurring intrusion behavior can be quickly responded to in real time.

BRIEF DESCRIPTION OF DRAWINGS

[0014] The accompanying drawings described herein are used to provide a further understanding of the present disclosure, and form a part of the present disclosure. Exemplary embodiments of the present disclosure and descriptions thereof are used to explain the present disclosure, and do not constitute any inappropriate limitation to the present disclosure. In the accompanying drawings:

FIG. 1a is a schematic structural diagram of a host device provided by an exemplary embodiment of the present disclosure;

FIG. 1b is a schematic structural diagram of a host device and another device provided by an exemplary embodiment of the present disclosure;

FIG. 2 is a schematic structural diagram of another host device and a security scanning program provided by an exemplary embodiment of the present disclosure;

FIG. 3 is a schematic diagram of an internal structure of another host device and a security scanning program provided by an exemplary embodiment of the present disclosure;

FIG. 4 is a schematic flowchart of a security scanning method for cloud computing

provided by an exemplary embodiment of the present disclosure;

FIG. 5 is a schematic flowchart of another security scanning method for cloud computing provided by an exemplary embodiment of the present disclosure;

FIG. 6 is a schematic structural diagram of a virtualization management apparatus provided by an exemplary embodiment of the present disclosure; and

FIG. 7 is a schematic structural diagram of a host device provided by an exemplary embodiment of the present disclosure.

DESCRIPTION OF EMBODIMENTS

[0015] To make the objectives, technical solutions, and advantages of the present disclosure clearer, the technical solutions of the present disclosure will be clearly and completely described below with reference to specific embodiments of the present disclosure and corresponding accompanying drawings. It is clear that the described embodiments are merely some rather than all of the embodiments of the present disclosure. All other embodiments obtained by a person of ordinary skill in the art based on the embodiments of the present disclosure without creative efforts shall fall within the protection scope of the present disclosure.

[0016] It should be noted that all user information (including, but not limited to, user equipment information, user personal information, and the like) and data (including, but not limited to, data for analysis, stored data, displayed data, and the like) involved in the present disclosure are all information and data that are authorized by users or fully authorized by all parties. The collection, use, and processing of related data need to conform to related laws, regulations, and standards of related countries and regions, and corresponding operation entries are provided for users to choose whether to grant or deny authorization.

[0017] For a problem existing in the prior art, embodiments of the present disclosure provide a security scanning solution for cloud computing. By combining a multi-queue design used during reading/writing of a physical storage (such as a disk), and utilizing a virtualized memory sharing and the physical storage (such as the disk) protocol design, data in physical storage space of a virtualization instance is shared with a security scanning program external to the virtualization instance in a virtualized manner, for the security scanning program to perform security scanning.

[0018] Specifically, in the embodiments of the present disclosure, the physical storage space of the virtualization instance supports a virtualization manager to perform a read/write

operation on the physical storage space in a multi-queue read/write manner. Based on this, functionality extension is performed on the virtualization manager, so that in addition to creating, for the virtualization instance, a storage queue required for performing the read/write operation on the physical storage space, the virtualization manager can further create a shared queue and a shared memory for the security scanning program located external to the virtualization instance, and further provide the data in the physical storage space to the security scanning program external to the virtualization instance based on the shared queue and the shared memory, for the security scanning program to perform host security scanning. Because the security scanning program is located external to the virtualization instance, the security scanning program is not susceptible to intrusion or interference from an internal process of the virtualization instance, has a good stealth capability, and does not occupy computing resources of the virtualization instance. In addition, host security scanning is performed based on the data in the physical storage space of the virtualization instance, so that real-time and highly efficient scanning can be implemented, and an immediately occurring intrusion behavior can be quickly responded to, thereby achieving high timeliness.

[0019] The following describes the solution provided by the embodiments of the present disclosure in detail with reference to the accompanying drawings.

[0020] FIG. 1a is a schematic structural diagram of a host device provided by an exemplary embodiment of the present disclosure. As shown in FIG. 1a, a virtualization manager 102 and a virtualization instance 103 managed by the virtualization manager 102 are deployed on the host device 101.

[0021] In this embodiment, the host device 101 may be any physical device capable of providing a hardware resource, for example, may be a mobile phone, a notebook computer, a storage device, or a conventional server, or may be a cloud device having hardware resources such as a cloud server or a cloud host. In this embodiment, the host device 101 is responsible for providing hardware resources required for running of the virtualization manager 102 and the virtualization instance. The hardware resources include, for example, but are not limited to: a processor (Central Processing Unit, CPU), a storage (such as a memory), a network interface card, a persistent storage, and an input/output device.

[0022] The host device 101 not only has the hardware resources, but also includes an operating system (OS) on the hardware resources. The virtualization manager 102 is run on the operating system. The virtualization manager 102 is configured to create an isolated running environment on the host device on which the virtualization manager 102 is located, and start or create and manage the virtualization instance in the isolated running environment. The

virtualization manager 102 may be any software program, component, or module that can create, based on a virtualization technology, the isolated running environment on the host device on which the virtualization manager 102 is located, and start or create and manage the virtualization instance in the isolated running environment. The virtualization manager 102 may further provide independent resource allocation, access control, and support of different operating systems for the virtualization instance. This is not limited. Optionally, the virtualization manager 102 may be a virtual machine monitor (VMM), which may also be referred to as a Hypervisor. Optionally, the Hypervisor or the VMM may be specifically implemented as a virtual operating system emulator (Quick EMUlator, QEMU) or VMware. However, this is not limited thereto. The virtualization instance may be implemented as a cloud host, an elastic compute service (ECS), a virtual machine (VM), a container, or the like. One virtualization manager 102 may manage one or more virtualization instances, and each virtualization instance is managed by one virtualization manager 102. FIG. 1a is shown by using an example in which one virtualization manager corresponds to one virtualization instance. However, this is not limited thereto.

[0023] In this embodiment, each virtualization manager 102 manages the virtualization instance in the same or similar manner and process. Any one of the virtualization instances is used as an example for description in the following embodiments. For ease of distinction and description, the any one of the virtualization instances is referred to as a first virtualization instance 103.

[0024] In this embodiment, the first virtualization instance provides the isolated running environment, a user space program may be deployed in the isolated running environment, and the user space program may also be referred to as an application program. In addition, the first virtualization instance has target physical storage space for storing an operating system, a user space program, and various generated data of the first virtualization instance 103. The target physical storage space is storage space allocated by the virtualization manager to the first virtualization instance from physical storage space of the host device. The physical storage space of the host device may use various persistent storage media such as a disk and a flash memory. In subsequent embodiments, the physical storage space of the host device is described by using a disk as an example. However, this is not limited thereto. Correspondingly, the target physical storage space allocated to the first virtualization instance may also be referred to as disk space of the first virtualization instance for short.

[0025] In this embodiment, the first virtualization instance carries the user space program, for example, a social user space program, a shopping user space program, or a gaming user

space program. In a process of running the user space program, data is generated, and the generated data needs to be written into the target physical storage space for storage. Certainly, in the process of starting or running the user space program, data also needs to be read from the target physical storage space, which involves a read/write operation performed on the target physical storage space. To implement the read/write operation on the target physical storage space, the virtualization manager creates a storage device for the target physical storage space, and creates at least one storage queue for the user space program in the first virtualization instance. The storage queue may be created by a boot program in a process of starting the first virtualization instance. The virtualization manager may perform, according to a read/write request of the user space program based on the storage device and the at least one storage queue, the read/write operation on the target physical storage space in a multi-queue read/write manner. In other words, the target physical storage space supports the virtualization manager to perform the read/write operation on the target physical storage space in the multi-queue read/write manner. The storage device (for example, a disk device) may be a virtual device, or may be a physical storage device. The user space program in the first virtualization instance writes a read/write instruction for the target physical storage space into an idle storage queue in the at least one storage queue, where a function of the at least one storage queue is to receive and store the read/write instruction from the user space program, and one of functions of the storage device is to: read the read/write instruction from the at least one storage queue, read, from the target physical storage space based on the read instruction that is read, data required by the user space program, and return the read data to the user space program; and read related data from memory space of the first virtualization instance based on a memory address in the write instruction that is read, and write the read data into the target physical storage space.

[0026] The following describes, with reference to a disk read/write architecture shown in FIG. 2, a work flow of performing disk reading/writing in a multi-queue manner in a cloud computing scenario. The disk read/write architecture shown in FIG. 2 involves three parts: a user space program, target physical storage space (where disk space is used as an example) of a first virtualization instance, and a kernel driver corresponding to the target physical storage space (a kernel driver of a disk for short). The kernel driver of the disk belongs to an operating system of the first virtualization instance. A disk read/write process mainly involves three steps: 1. The user space program interacts with the kernel driver of the disk, and through the kernel driver of the disk, writes a read/write instruction into a storage queue in a virtualization manager and interacts with the disk device (an example of a storage device); 2. The kernel driver of the disk further includes a core protocol module and a peripheral component

interconnect (PCI) bus module, where the core protocol module writes the read/write instruction into the storage queue through the PCI bus module, and is responsible for interacting with the disk device; 3. The disk device emulated by the virtualization manager provides a capability of accessing the target physical storage space of the first virtualization instance. A specific process is as follows.

[0027] (1) The user space program invokes and accesses a file system based on the operating system of the first virtualization instance, the file system forwards a read/write request for the target physical storage space to the kernel driver, and the kernel driver interacts with the disk device to implement reading/writing of the target physical storage space.

[0028] (2) The PCI bus module performs command control on the target physical storage space through a disk device I/O (Input/Output). The core protocol module allocates memory space, creates a data interaction module to send a protocol command and collect an execution result, and finally completes data transmission to and from the target physical storage space.

[0029] (3) To improve I/O data interaction efficiency, the data interaction module is usually in a multi-queue form. For example, the data interaction module includes a plurality of storage queues, and the PCI bus module submits the read/write request to any idle storage queue and processes any storage queue that returns a message notification, thereby improving I/O efficiency in a concurrent manner.

[0030] (4) The PCI bus module submits, through the storage queue, a command to read data from a specified address on the target physical storage space. The disk device parses the corresponding command, writes data at the specified address on the target physical storage space into the memory space specified by the storage queue, and sends a message to notify the PCI bus module to read the corresponding data.

[0031] As cloud computing technologies develop rapidly, a cloud computing instance in a form of a virtual machine (VM) has become one of core technologies of modern data centers. As a quantity of cloud computing instances increases, host security becomes an important scenario for cloud security confrontation. Host security is a security method for detecting a risk vulnerability and an intrusion behavior on a host based on collection and analysis of information such as a memory or a disk of a virtualization instance. In host security, scanning a disk of the host is an important information collection means for detecting whether the virtualization instance has an attack behavior against the host, collecting a malicious sample, and reconstructing an attack path.

[0032] In this embodiment, the security scanning program is innovatively provided external to the first virtualization instance, and functionality extension is performed on the

virtualization manager through a multi-queue design used during reading/writing of the target physical storage space, so that the virtualization manager can cooperate with the security scanning program external to the first virtualization instance to perform host security scanning on the data in the target physical storage space of the first virtualization instance, thereby ensuring security of the host device.

[0033] The security scanning program 104 is mainly configured to perform security scanning on the target physical storage space of the first virtualization instance 103. For example, in a case that the first virtualization instance is implemented as a virtual machine, the security scanning program is implemented as a secure virtual machine supervisor (Secure VM Supervisor).

[0034] The security scanning program belongs to agentless scanning, and the agentless scanning is a technical branch of a virtual machine host security detection technology. Through an interface provided in the virtualization technology, the agentless scanning can collect and analyze a running state of a scanned virtualization instance without installing any agent software in the virtualization instance. The agentless scanning can avoid a denial of service (DoS) attack or a tampering attack against an agent in the virtualization instance, is not affected by a network connection requirement or performance of the virtualization instance itself, and does not need to consider an increasingly complex intrusion countermeasure, thereby ensuring high data reliability and service stability.

[0035] A deployment manner of the security scanning program is not limited. The security scanning program may be deployed on the host device on which the first virtualization instance is located, or may be deployed on another device communicatively connected to the host device. Optionally, the security scanning program is implemented as a second virtualization instance deployed on the host device. Further, the security scanning program is implemented as a user space program deployed in the second virtualization instance on the host device. The second virtualization instance and the first virtualization instance may be managed by a same virtualization manager, or may be managed by different virtualization managers. Alternatively, the security scanning program is implemented as a second virtualization instance on another device, and is further implemented as a user space program in the second virtualization instance on another device. Certainly, the security scanning program may be directly deployed, as the user space program, on the host device on which the first virtualization instance is located or another device. In FIG. 1a, FIG. 2, and FIG. 3, an example in which the security scanning program is deployed on the host device is used for illustration. However, this is not limited thereto. In FIG. 1b, an example in which the security scanning program is deployed on another

device communicatively connected to the host device is used for illustration. However, this is not limited thereto.

[0036] In the embodiments, in a case that the virtualization manager 102 performs the read/write operation on the target storage space in the multi-queue manner, in addition to creating at least one storage queue for the first virtualization instance 103, the virtualization manager 102 further creates a shared queue and a shared memory for the security scanning program external to the first virtualization instance, and provides, based on the shared queue and the shared memory, the data in the target physical storage space to the security scanning program in a manner in which the user space program normally reads the target physical storage space, so that the security scanning program can perform host security scanning based on the data in the target physical storage space of the first virtualization instance without the awareness of the operating system and the user space program of the first virtualization instance, to ensure security of the host device and the first virtualization instance. Because the security scanning program is external to the virtualization instance, the security scanning program is not susceptible to intrusion or interference from an internal process of the first virtualization instance, and does not occupy computing resources of the first virtualization instance.

[0037] A function of the shared queue is different from a function of the at least one storage queue. The shared queue mainly shares the data in the target physical storage space with the security scanning program, so that the security scanning program can perform host security scanning based on the data in the target physical storage space of the first virtualization instance without the awareness of the operating system and the user space program of the first virtualization instance, to ensure security of the host device and the first virtualization instance. The storage queue is mainly provided for the user space program on the first virtualization instance to read/write the data in the target physical storage space.

[0038] In the embodiments, data of the target physical storage space of the virtualization instance is intercepted in the virtualization manager without modifying any upper-layer components (for example, the operating system of the first virtualization instance, as well as the disk driver, the user space program, and the like on the operating system) of the virtualization instance, and is forwarded to the security scanning program, to complete disk scanning of the virtualization instance.

[0039] In addition, host security scanning is performed based on the data in the target physical storage space of the first virtualization instance, so that real-time and highly efficient scanning can be implemented, and an immediately occurring intrusion behavior can be quickly responded to.

[0040] In addition, the data in the target physical storage space is provided to the security scanning program based on the shared queue and the shared memory, so that snapshots or disk images do not need to be retained for storage of the first virtualization instance, and storage space does not need to be provided for the disk images or the snapshots, thereby saving the storage space, and further ensuring efficiency of host security scanning.

[0041] Further, by using characteristics of cloud computing virtualization, good out-of-band control over the operating system on the first virtualization instance, the operation of the user space program, and the storage device can effectively resist in-band intrusion attacks and interference while exhibiting good concealment against a potential ongoing attack behavior, thereby helping security personnel to better perform targeted detection and analysis on the intrusion behavior.

[0042] Further, by fully using the characteristics of cloud computing virtualization, the security scanning program can be encapsulated in an independent virtualization instance (for example, a VM), disk scanning on virtualization instances is implemented through a custom kernel driver. A cloud computing virtualization manager is used to perform resource control on the virtualization instance in which the security scanning program is deployed, to limit computing resource consumption of the security scanning program, a one-to-many scanning framework for a single security scanning program and a plurality of virtualization instances as scanning objects is implemented by using a resource sharing capability conveniently implemented in cloud computing, and results in high scalability.

[0043] In the embodiments, an implementation in which the virtualization manager 102 creates the shared queue for the security scanning program is not limited, and mainly includes a direct manner and an indirect manner. The direct manner means that after the boot program starts the first virtualization instance, the virtualization manager 102 creates the shared queue for the security scanning program. The indirect manner means that in the process of starting the first virtualization instance, the virtualization manager 102 controls the boot program to create the shared queue for the security scanning program. The boot program in the embodiments of the present disclosure includes, but is not limited to: a basic input/output system (Basic Input Output System, BIOS) or a unified extensible firmware interface (UEFI).

[0044] In an optional embodiment, the direct manner is as follows. In a case that the virtualization manager 102 creates the shared queue for the security scanning program after the first virtualization instance is started, the virtualization manager 102 creates the shared queue for the security scanning program by using an input/output (I/O) virtualization protocol. The I/O virtualization protocol is an I/O virtualization technology for a virtualization platform,

provides a standardized interface to implement high-performance and low-overhead I/O communication between the virtualization instance and a host machine, and supports a plurality of queues, so that the I/O virtualization technology can simultaneously process a plurality of input/output (I/O) requests. In a disk read/write manner based on the I/O virtualization protocol, disk data of the first virtualization instance may be mapped to the security scanning program external to the first virtualization instance through the shared queue via the virtualization manager, to implement disk read/write sharing, so that the security scanning program scans the disk space of the first virtualization instance. Specifically, in a case that the virtualization manager 102 uses the I/O virtualization protocol, the virtualization manager 102 executes a first boot program corresponding to the first virtualization instance, to start the first virtualization instance. The first boot program is a boot program required for starting the virtualization instance. The shared queue is not created in the process of starting the first virtualization instance 103 based on the first boot program. After the first virtualization instance is started, the virtualization manager 102 creates the shared queue for the security scanning program. Optionally, the I/O virtualization protocol may be implemented as a Virtio protocol. The Virtio protocol is an I/O virtualization technology for a virtualization platform, and belongs to a manner of managing a disk by the virtualization manager 102. Virtio provides a standardized interface to implement high-performance and low-overhead I/O communication between the virtualization instance and the host device. Virtio supports a plurality of queues, and simultaneously processes a plurality of I/O requests. In Virtio, each queue has its own independent control structure and buffer, and may be associated with different processor cores or threads, thereby implementing concurrent processing.

[0045] In another optional embodiment, the indirect manner is as follows. In the process of starting the first virtualization instance, the virtualization manager 102 controls the boot program to create a shared queue for the security scanning program. A passthrough non-volatile memory express (NVMe) protocol is a protocol for disk device communication, and can satisfy a high requirement on read/write efficiency. The NVMe protocol, designed with multi-queue support and oriented toward peripheral component interconnect express (PCI-express, PCIe), is a more preferred protocol for enabling current fast block devices, especially solid-state drives, to fully deliver their performance. A device supporting the NVMe protocol is referred to as an NVMe device. The NVMe protocol is an inherent attribute of the disk, and needs cooperation of the virtualization manager to perform a multi-queue read/write operation or control on the disk. In a disk read/write manner of a device-passthrough-based NVMe protocol, a plurality of storage queues may be created by using a multi-queue disk protocol supported by the target

physical storage space, and disk data of the first virtualization instance is mapped to the security scanning program external to the first virtualization instance through the shared queue, to implement disk read/write sharing, thereby achieving the same effect. Specifically, when the device-passthrough-based NVMe protocol is used, the boot program for starting the first virtualization instance is improved, so that an improved second boot program can create the shared queue in the process of starting the first virtualization instance. The first boot program is different from the second boot program. The virtualization manager executes the second boot program corresponding to the first virtualization instance, to start the first virtualization instance. In addition, during startup, the second boot program creates the shared queue for the security scanning program. After creating the shared queue, the second boot program returns control of the first virtualization instance to the operating system of the first virtualization instance. The second boot program is independent software that is independent of the operating system and the user space program of the first virtualization instance. In a cloud computing scenario, the second boot program is controlled by a cloud provider, and is not aware of by the operating system or the user space program of the first virtualization instance.

[0046] It is noted herein that both the first boot program and the second boot program can be considered as a part of the virtualization manager.

[0047] In an optional embodiment, regardless of the foregoing implementation of creating the shared queue, the implementation in which the virtualization manager creates the shared memory may be implemented as follows. A target memory is allocated to the security scanning program from the memory of the first virtualization instance, and the target memory is set as a shared attribute for the security scanning program, to obtain the shared memory.

[0048] In an optional embodiment, an implementation in which the virtualization manager provides the target data in the target physical storage space to the security scanning program based on the shared queue and the shared memory is not limited. The following provides a specific implementation. In a case that the security scanning program needs to perform security scanning, the security scanning program sends a read request to the virtualization manager based on the shared queue, for example, writes the read request into the shared queue. Correspondingly, the virtualization manager receives, based on the shared queue, the read request sent by the security scanning program, and reads the target data from the target physical storage space based on the read request. For example, the read request includes: address information and a data length of the target physical storage space of the first virtualization instance, where the address information is a logic address. The disk device parses the read request, converts the logic address in the read request into a physical address, and reads the

target data from the target storage space based on the physical address; and writes the target data into the shared memory, for the security scanning program to read the target data from the shared memory. For example, after the target data is written into the shared memory by using an interrupt redirection mechanism, interruption is generated, and notification information is no longer sent to the user space program but sent to the security scanning program, to notify the security scanning program to read the target data from the shared memory to perform security scanning.

[0049] In the embodiments, the first virtualization instance may use a security scanning service, or may not use a security scanning service. An implementation of using the security scanning service is not limited. In an optional embodiment, a human-computer interaction interface is provided for a user, and the user may choose, on the human-computer interaction interface, whether a virtualization instance purchased by the user uses the security scanning service. For the first virtualization instance, in a case that the user determines that the first virtualization instance uses the security scanning service, the virtualization manager may obtain configuration information of the first virtualization instance, where the configuration information includes: indication information indicating that the first virtualization instance uses the security scanning service; and may determine, based on the indication information, that the first virtualization instance uses the security scanning service. Correspondingly, the virtualization manager creates the shared queue and the shared memory for the first virtualization instance. In another optional embodiment, a security scanning service binding relationship between the security scanning program and the first virtualization instance is pre-established. The security scanning program continuously detects a virtualization instance in a startup state, and sends a command indicating to create the shared queue and the shared memory to a virtualization manager corresponding to the virtualization instance that is in the startup state and with which the security scanning service binding relationship is established. The virtualization manager receives the command indicating to create the shared queue and the shared memory. When receiving the command, the virtualization manager determines that the first virtualization instance uses the security scanning service, and creates the shared queue for the security scanning program through the second boot program in the process of starting the virtualization instance. Alternatively, after starting the first virtualization instance by executing the first boot program, the virtualization manager creates the shared queue for the security scanning program.

[0050] In an optional embodiment, an internal structure of the virtualization manager is not limited. The following exemplarily provides an internal structure of the virtualization manager.

As shown in FIG. 3, the virtualization manager 102 includes: a creation module 102a, a first communication module 102b, and a storage device 102c allocated to a first virtualization instance. The storage device corresponds to a disk device.

[0051] In a case that the first virtualization instance uses a security scanning service, the creation module 102a creates a shared queue and a shared memory for a security scanning program external to the first virtualization instance, and provides address information of the shared queue and the shared memory to the first communication module 102b.

[0052] The first communication module 102b is responsible for communicating with the security scanning program, receiving a read request sent by the security scanning program, and writing the read request into the shared queue, so that the storage device 102c reads the read request from the shared queue. The storage device 102c reads the read request from the shared queue, reads target data from target physical storage space based on the read request, and writes the target data into the shared memory. Further, the first communication module 102b reads the target data from the shared memory and sends the target data to the security scanning program, for the security scanning program to perform security scanning on a host device based on the target data.

[0053] It is noted herein that the storage device 102c not only reads the read request from the shared queue, reads the target data from the target physical storage space based on the read request, and writes the target data into the shared memory, but also may perform a read/write operation on the target physical storage space based on a read/write request that is in at least one storage queue and that is from a user space program in the first virtualization instance. The storage device 102c may periodically poll or obtain, from a corresponding storage queue, a read request or a write request from the user space program based on an interrupt signal. For the read request from the user space program, the storage device 102c obtains, from the target physical storage space, first data that needs to be read by the user space program, writes the first data into memory space (a non-shared memory) of the first virtualization instance, notifies, through the storage queue, a kernel driver to read the first data, and sends the first data to the user space program. For the write request from the user space program, the storage device 102c obtains, from the memory space of the first virtualization instance, second data that needs to be written by the user space program, writes the second data into the target physical storage space, and notifies, through the storage queue, the kernel driver that the writing is completed. The kernel driver notifies the user space program that the second data has been written into the target physical storage space. The shared queue is different from the at least one storage queue.

[0054] Correspondingly, this embodiment further exemplarily provides an internal

structure of the security scanning program 104. As shown in FIG. 3, the security scanning program 104 includes: a second communication module 104a, a scanning module 104b, and a data analysis module 104c.

[0055] The data analysis module 104c determines disk address information and a data length of the first virtualization instance in response to a security scanning instruction or event. Real-time scanning may be set, and the security scanning instruction may be generated in real time; or a scanning cycle may be set to perform scanning, and the security scanning instruction may be generated when the scanning cycle is reached. A specific event is not limited. For example, the specific event may be an event that writing is monitored in the target physical storage space. The read request is generated based on the information, and the read request is provided to the scanning module 104b. The scanning module 104b provides the read request to the second communication module 104a. The second communication module 104a sends the read request to the first communication module 102b. The first communication module 102b receives the read request sent by the security scanning program, and writes the read request into the shared queue, so that the storage device 102c reads the read request from the shared queue. The storage device 102c reads the read request from the shared queue, reads the target data from the target physical storage space based on the read request, and writes the target data into the shared memory. The first communication module 102b reads the target data from the shared memory and sends the target data to the second communication module 104a, the second communication module 104a provides the target data to the scanning module 104b, and the scanning module 104b provides the target data to the data analysis module 104c. The data analysis module 104c detects, based on the target data, whether a host intrusion behavior occurs, and issues an alert to a user when the host intrusion behavior occurs. Manners of detecting, based on the target data, whether the host intrusion behavior occurs include, but are not limited to: determining whether data in a virus database or a risk database exists in the target data, determining whether log information included in the target data is cleared, determining whether the target data includes abnormal process traffic, determining whether an account with abnormal login exists in the target data, and the like.

[0056] The security scanning solution provided in the present disclosure can achieve the following technical effects.

[0057] **Strong timeliness:** The security scanning program interacts with the shared queue in the virtualization instance, so that the security scanning program obtains and scans the target data in a manner in which the user space program normally reads a file, reads the data that needs to be scanned in the virtualization instance in an instant and targeted manner, and does

not need to save mirror image or snapshot data of the entire virtualization instance before performing scanning. Therefore, a risk file may be quickly located and an attack behavior may be traced.

[0058] **Resource isolation:** By using characteristics of cloud computing virtualization, in this solution, all scanning-related components with high I/O and computational loads run external to the virtualization instance, and operate by using an independent special virtualization instance. For example, the security scanning program is implemented independent of the first virtualization instance, and may run on the same host device as the first virtualization instance, or may run on a host device different from the host device on which the first virtualization instance runs. After the user purchases the virtualization instance, the usage of computing resources purchased by the user is not affected by running of an in-band scanning agent (a security scanning program deployed internal to the virtualization instance), and significant I/O fluctuation caused by a scanning behavior of the agent does not occur, so that good isolation and protection can be implemented for a service resource.

[0059] **User data control:** Compared with a method of performing out-of-band scanning by using images/snapshots, in this solution, no additional storage resource is used to save a disk image or snapshot of the virtualization instance. Storage resources consumed by the cloud provider for saving user data may be saved. In addition, copying of the user data and a data leakage may be avoided, and the user data is protected from leaving an original computing region.

[0060] **Virtualization instance unawareness:** In the embodiments, because modifications are made at a virtualization manager layer under the virtualization instance, in the solution, concealment from the operating system of the virtualization instance and an upper-layer application thereof may be implemented, and the operating system, along with a kernel module and a user space program in the upper layer of the operating system, remains unaware. In addition, because performance overheads reside in the virtualization manager of the virtualization instance, the performance overheads do not directly affect performance of code running in the virtualization instance. An I/O queue that does not occupy the operating system of the virtualization instance has little impact on I/O performance. This further reduces a possibility of determining the existence of the agent through performance analysis.

[0061] In addition to providing the host device embodiment, embodiments of the present disclosure further provide a security scanning method for cloud computing. The following describes a process of the security scanning method for cloud computing provided by the embodiments of the present disclosure.

[0062] FIG. 4 is a schematic flowchart of a security scanning method for cloud computing provided by an exemplary embodiment of the present disclosure. The method is applied to a virtualization manager on a host device. The virtualization manager manages a first virtualization instance on the host device. As shown in FIG. 4, the method includes the following steps.

[0063] 401: In a case that the first virtualization instance uses a security scanning service, create a shared queue and a shared memory for a security scanning program located external to the first virtualization instance; the first virtualization instance has target physical storage space, and the target physical storage space supports the virtualization manager to perform a read/write operation on the target physical storage space in a multi-queue read/write manner.

[0064] 402: Provide target data in the target physical storage space to the security scanning program based on the shared queue and the shared memory, for the security scanning program to perform security scanning on the host device based on the target data, where the first virtualization instance performs the read/write operation on the target physical storage space through at least one storage queue.

[0065] In an optional embodiment, the provide target data in the target physical storage space to the security scanning program based on the shared queue and the shared memory includes: receiving, based on the shared queue, a read request sent by the security scanning program, and reading the target data from the target physical storage space based on the read request; and writing the target data into the shared memory, for the security scanning program to read the target data from the shared memory.

[0066] In an optional embodiment, the method provided by the embodiment of the present disclosure further includes: obtaining configuration information of the first virtualization instance, where the configuration information includes indication information indicating that the first virtualization instance uses the security scanning service, and determining, based on the indication information, that the first virtualization instance uses the security scanning service; or receiving a command that is sent by the security scanning program and that indicates to create the shared queue and the shared memory, and upon receipt of the command, determining that the first virtualization instance uses the security scanning service, where the command is sent in a case that the security scanning program establishes a security scanning service binding relationship with the first virtualization instance.

[0067] In an optional embodiment, the create a shared queue for a security scanning program located external to the first virtualization instance includes: in a case of using an I/O virtualization protocol (for example, a Virtio protocol), executing a first boot program

corresponding to the first virtualization instance, to start the first virtualization instance, and creating the shared queue for the security scanning program after the first virtualization instance is started; or in a case of using a device-passthrough-based NVMe protocol, executing a second boot program corresponding to the first virtualization instance, to start the first virtualization instance, and creating the shared queue for the security scanning program by the second boot program during startup.

[0068] FIG. 5 shows another security scanning method for cloud computing provided by an exemplary embodiment of the present disclosure. The method is applied to a security scanning program external to a first virtualization instance. As shown in FIG. 5, the method includes the following steps.

[0069] 501: In a case that a virtualization manager corresponding to the first virtualization instance creates a shared queue and a shared memory for the security scanning program, write a read request into the shared queue, for the virtualization manager to read target data from target physical storage space of the first virtualization instance based on the read request, and write the target data into the shared memory.

[0070] 502: Read the target data from the shared memory, and perform security scanning based on the target data, where the target physical storage space supports the virtualization manager to perform a read/write operation on the target physical storage space in a multi-queue read/write manner, and the first virtualization instance performs the read/write operation on the target physical storage space through at least one storage queue.

[0071] In an optional embodiment, the method provided by the present disclosure further includes: sending to the virtualization manager a command that indicates to create the shared queue and the shared memory, for the virtualization manager to determine, upon receipt of the command, that the first virtualization instance uses a security scanning service, where the command is sent in a case that the security scanning program establishes a security scanning service binding relationship with the first virtualization instance.

[0072] Detailed implementations and beneficial effects of the steps in the methods shown in FIG. 4 and FIG. 5 provided in the embodiments of the present disclosure have been described in detail in the foregoing embodiments, and are not described in detail herein.

[0073] It should be noted that, the steps of the method provided by the foregoing embodiments may be executed by a same device, or the method may be executed by different devices. For example, the execution subject of step 501 and step 502 may be a device A. For another example, the execution subject of step 501 may be the device A, the execution subject of step 502 may be a device B, and the like.

[0074] In addition, some processes described in the foregoing embodiments and accompanying drawings include a plurality of operations that appear in a specific sequence. However, it should be clearly understood that the operations may not be performed in the sequence in which the operations appear in this specification or be performed in parallel. Sequence numbers of the operations, such as 501 and 502, are merely used for distinguishing different operations, and the sequence numbers themselves do not represent any execution sequence. In addition, the processes may include more or fewer operations, and the operations may be performed in sequence or in parallel. It should be noted that, the descriptions of "first", "second", and the like in this specification are used to distinguish different messages, devices, modules, and the like, and do not indicate a sequence and do not limit that "first" and "second" are different types either.

[0075] FIG. 6 is a schematic structural diagram of a virtualization management apparatus provided by an exemplary embodiment of the present disclosure. The apparatus may be implemented as a virtualization manager on a host device, and is configured to manage a first virtualization instance. As shown in FIG. 6, the apparatus includes: a creation module 61 and a providing module 62.

[0076] The creation module 61 is configured to: in a case that the first virtualization instance uses a security scanning service, create a shared queue and a shared memory for a security scanning program located external to the first virtualization instance; the first virtualization instance has target physical storage space, and the target physical storage space supports the virtualization manager to perform a read/write operation on the target physical storage space in a multi-queue read/write manner.

[0077] The providing module 62 is configured to provide target data in the target physical storage space to the security scanning program based on the shared queue and the shared memory, for the security scanning program to perform security scanning on the host device based on the target data, where the first virtualization instance performs the read/write operation on the target physical storage space through at least one storage queue, and the shared queue is different from the at least one storage queue.

[0078] In an optional embodiment, the providing module 62 is specifically configured to: receive, based on the shared queue, a read request sent by the security scanning program, and read the target data from the target physical storage space based on the read request; and write the target data into the shared memory, for the security scanning program to read the target data from the shared memory.

[0079] In an optional embodiment, the virtualization management apparatus provided by

the embodiment of the present disclosure further includes: an obtaining module, a determining module, and a receiving module. The obtaining module is configured to obtain configuration information of the first virtualization instance, where the configuration information includes indication information indicating that the first virtualization instance uses the security scanning service, and the determining module is configured to determine, based on the indication information, that the first virtualization instance uses the security scanning service; or the receiving module is configured to receive a command that is sent by the security scanning program and that indicates to create the shared queue and the shared memory, and the determining module is configured to, upon receipt of the command, determine that the first virtualization instance uses the security scanning service, where the command is sent in a case that the security scanning program establishes a security scanning service binding relationship with the first virtualization instance.

[0080] In an optional embodiment, the creation module is specifically configured to: in a case of using an I/O virtualization protocol (for example, a Virtio protocol), execute a first boot program corresponding to the first virtualization instance, to start the first virtualization instance, and create the shared queue for the security scanning program after the first virtualization instance is started; or in a case of using a device-passthrough-based NVMe protocol, execute a second boot program corresponding to the first virtualization instance, to start the first virtualization instance, and create the shared queue for the security scanning program by the second boot program during startup.

[0081] Detailed implementations and beneficial effects of the steps in the apparatus shown in FIG. 6 provided in the embodiment of the present disclosure have been described in detail in the foregoing embodiments, and are not described in detail herein.

[0082] FIG. 7 is a schematic structural diagram of another host device provided by an exemplary embodiment of the present disclosure. As shown in FIG. 7, the device includes: a memory 74 and a processor 75.

[0083] The memory 74 is configured to store a security scanning program, and may be configured to store various other data to support operations on the host device. Examples of such data include instructions of any application program or method operating on the host device, and the like.

[0084] The processor 75 is coupled to the memory 74, and is configured to execute the security scanning program in the memory 74. The processor 75 is configured to: in a case that a virtualization manager corresponding to a first virtualization instance creates a shared queue and a shared memory for the security scanning program, write a read request into the shared

queue, for the virtualization manager to read target data from target physical storage space of the first virtualization instance based on the read request, and write the target data into the shared memory; and read the target data from the shared memory, and perform security scanning on the host device based on the target data, where the target physical storage space supports the virtualization manager to perform a read/write operation on the target physical storage space in a multi-queue read/write manner, the first virtualization instance performs the read/write operation on the target physical storage space through at least one storage queue, and the shared queue is different from the at least one storage queue.

[0085] In an optional embodiment, the processor 75 is further configured to send to the virtualization manager a command that indicates to create the shared queue and the shared memory, for the virtualization manager to determine, upon receipt of the command, that the first virtualization instance uses a security scanning service, where the command is sent in a case that the security scanning program establishes a security scanning service binding relationship with the first virtualization instance.

[0086] In an optional embodiment, the security scanning program is a user space program deployed in a second virtualization instance on the host device or another device, or the security scanning program is a user space program directly deployed on the host device or on another device.

[0087] Detailed implementations and beneficial effects of the steps in the device shown in FIG. 7 provided in the embodiment of the present disclosure have been described in detail in the foregoing embodiments, and are not described in detail herein.

[0088] Further, as shown in FIG. 7, the host device further includes: a communication component 76, a power supply component 78, and other components. Only some components are schematically provided in FIG. 7, which does not mean that the host device includes only the components shown in FIG. 7. The host device according to this embodiment is implemented as a cloud server.

[0089] Detailed implementations and beneficial effects of the host device provided in the embodiment of the present disclosure have been described in detail in the foregoing embodiments, and are not described in detail herein.

[0090] Correspondingly, an embodiment of the present disclosure further provides a computer-readable storage medium storing a computer program. When the computer program is executed by a processor, the steps in the methods shown in FIG. 4 and FIG. 5 can be implemented.

[0091] Correspondingly, an embodiment of the present disclosure further provides a

computer program product, including a computer program. When the computer program is executed by a processor, the steps in the methods shown in FIG. 4 and FIG. 5 are implemented.

[0092] The memory may be implemented by any type of volatile or non-volatile storage device or a combination thereof, such as a static random-access memory (SRAM), an electrically erasable programmable read-only memory (EEPROM), an erasable programmable read-only memory (EPROM), a programmable read-only memory (PROM), a read-only memory (ROM), a magnetic memory, a flash memory, a disk, or an optical disc.

[0093] The communication component is configured to facilitate communication between a device in which the communication component is located and another device in a wired or wireless manner. The device in which the communication component is located may access a wireless network based on a communication standard, for example, a mobile communication network such as Wi-Fi, 2G, 3G, 4G/LTE, 5G, or a combination thereof. In an exemplary embodiment, the communication component receives a broadcast signal or broadcast related information from an external broadcast management system through a broadcast channel. In an exemplary embodiment, the communication component further includes a near field communication (NFC) module to facilitate short-distance communication. For example, the NFC module may be implemented based on a radio frequency identification (RFID) technology, an infrared data association (IrDA) technology, an ultra wide band (UWB) technology, a BlueTooth (BT) technology, and other technologies.

[0094] The display includes a screen, and the screen may include a liquid crystal display (LCD) and a touch panel (TouchPanel, TP). If the screen includes the touch panel, the screen may be implemented as a touchscreen to receive an input signal from a user. The touch panel includes one or more touch sensors to sense a touch, a swipe, and a gesture on the touch panel. The touch sensor can not only sense a boundary of a touch or swipe action, but also detect duration and pressure related to the touch or swipe operation.

[0095] The power supply component supplies power to various components of a device in which the power supply component is located. The power supply component may include a power management system, one or more power supplies, and other components associated with generation, management, and distribution of power for the device in which the power supply component is located.

[0096] The audio component may be configured to output and/or input an audio signal. For example, the audio component includes a microphone (MIC). When a device in which the audio component is located is in an operation mode, for example, a call mode, a record mode, and a voice recognition mode, the microphone is configured to receive an external audio signal. The

received audio signal may be further stored in the memory or sent via the communication component. In some embodiments, the audio component further includes a speaker, configured to output an audio signal.

[0097] A person skilled in the art should understand that, the embodiments of the present disclosure may be provided as a method, a system, or a computer program product. Therefore, the present disclosure may use a form of hardware-only embodiments, software-only embodiments, or embodiments combining software and hardware aspects. In addition, the present disclosure may use a form of a computer program product that is implemented on one or more computer-readable storage media (including, but not limited to, a disk storage, a compact disc read-only memory (CD-ROM), an optical storage, and the like) that include computer-usable program code.

[0098] The present disclosure is described with reference to the flowcharts and/or block diagrams of the method, the device (system), and the computer program product according to the embodiments of the present disclosure. It should be understood that computer program instructions may be used to implement each process and/or each block in a flowchart and/or a block diagram and a combination of processes and/or blocks in the flowchart and/or the block diagram. The computer program instructions may be provided to a processor of a general-purpose computer, a dedicated computer, an embedded processor, or other programmable data processing device to generate a machine, so that the instructions executed by the processor of the computer or other programmable data processing device generate an apparatus for implementing a specific function in one or more processes in the flowcharts and/or in one or more blocks in the block diagrams.

[0099] The computer program instructions may be stored in a computer-readable memory that can direct the computer or other programmable data processing device to operate in a specific manner, so that the instructions stored in the computer-readable memory generate an article of manufacture that includes an instruction apparatus. The instruction apparatus implements functions specified in one or more processes in the flowcharts and/or in one or more blocks in the block diagrams.

[0100] The computer program instructions may further be loaded onto a computer or other programmable data processing device, so that a series of operations and steps are performed on the computer or other programmable device, thereby generating computer-implemented processing. Therefore, the instructions executed on the computer or other programmable devices provide steps for implementing functions specified in one or more processes in the flowcharts and/or in one or more blocks in the block diagrams.

[0101] In a typical configuration, a computing device includes one or more processors (Central Processing Units, CPUs), an input/output interface, a network interface, and a memory.

[0102] The memory may include a non-permanent memory, a random access memory (RAM), and/or a non-volatile memory in a form of the computer-readable medium, such as a read-only memory (ROM) or a flash memory (flash RAM). The memory is an example of the computer-readable medium.

[0103] The computer-readable media include a permanent medium and a non-permanent medium, or a removable medium and a non-removable medium. Information storage may be implemented by using any method or technology. The information may be a computer-readable instruction, a data structure, a module of a program, or other data. Examples of computer storage media include, but are not limited to, a phase-change memory (PRAM), a static random access memory (SRAM), a dynamic random access memory (DRAM), other types of random access memory (RAM), a read-only memory (ROM), an electrically erasable programmable read-only memory (EEPROM), a flash memory or another memory technology, a compact disc read-only memory (CD-ROM), a digital versatile disc (Digital Video Disc, DVD) or other optical storage, a magnetic tape cartridge, tape and disk storage or other magnetic storage device, or any other non-transmission media that may be configured to store information accessible to the computing device. According to definitions herein, the computer-readable media does not include transitory computer-readable media (transitory media), such as a modulated data signal and carrier.

[0104] It should be further noted that, the terms "include", "comprise", or any variants thereof are intended to cover a non-exclusive inclusion. Therefore, a process, method, article, or device that includes a series of elements not only includes such elements, but also includes other elements not specified expressly, or may include inherent elements of the process, method, article, or device. Without further limitations, an element defined by a phrase "including one....." does not exclude the presence of additional same elements existing in the process, method, product, or device that includes the element.

[0105] The foregoing descriptions are merely embodiments of the present disclosure, but are not intended to limit the present disclosure. For a person skilled in the art, various modifications and variations can be made to the present disclosure. Any modification, equivalent substitution, and improvement made within the spirit and principle of the present disclosure shall fall within the scope of the claims of the present disclosure.

CLAIMS

1. A host device, wherein a virtualization manager and a first virtualization instance managed by the virtualization manager are deployed on the host device, the first virtualization instance has target physical storage space, and the target physical storage space supports the virtualization manager to perform a read/write operation on the target physical storage space in a multi-queue read/write manner;

the virtualization manager is further configured to: in a case of determining that the first virtualization instance uses a security scanning service, create a shared queue and a shared memory for a security scanning program located external to the first virtualization instance; and provide target data in the target physical storage space to the security scanning program based on the shared queue and the shared memory, for the security scanning program to perform security scanning on the host device based on the target data; and

wherein the first virtualization instance performs the read/write operation on the target physical storage space through at least one storage queue.

2. The host device according to claim 1, wherein when providing the target data in the target physical storage space to the security scanning program based on the shared queue and the shared memory, the virtualization manager is specifically configured to:

receive, based on the shared queue, a read request sent by the security scanning program, and read the target data from the target physical storage space based on the read request; and

write the target data into the shared memory, for the security scanning program to read the target data from the shared memory.

3. The host device according to claim 1 or 2, wherein the virtualization manager is further configured to:

obtain configuration information of the first virtualization instance, wherein the configuration information comprises indication information indicating that the first virtualization instance uses the security scanning service, and determine, based on the indication information, that the first virtualization instance uses the security scanning service;

or

receive a command that is sent by the security scanning program and that indicates to create the shared queue and the shared memory, and upon receipt of the command, determine that the first virtualization instance uses the security scanning service, wherein the command is sent in a case that the security scanning program establishes a security scanning service binding relationship with the first virtualization instance.

4. The host device according to any one of claims 1 to 3, wherein when creating the shared queue for the security scanning program located external to the first virtualization instance, the virtualization manager is specifically configured to:

in a case of using an Input/Output (I/O) virtualization protocol, execute a first boot program corresponding to the first virtualization instance, to start the first virtualization instance, and create the shared queue for the security scanning program after the first virtualization instance is started;

or

in a case of using a device-passthrough-based non-volatile memory express (NVMe) protocol, execute a second boot program corresponding to the first virtualization instance, to start the first virtualization instance, and create the shared queue for the security scanning program by the second boot program during startup.

5. The host device according to any one of claims 1 to 4, wherein the virtualization manager comprises:

a creation module, configured to: in a case that the first virtualization instance uses the security scanning service, create the shared queue and the shared memory for the security scanning program located external to the first virtualization instance;

a communication module, configured to: receive the read request sent by the security scanning program, and write the read request into the shared queue; and read the target data from the shared memory and send the target data to the security scanning program, for the security scanning program to perform security scanning on the host device based on the target data; and

a storage device, allocated to the first virtualization instance, and configured to: read the read request from the shared queue, read the target data from the target physical storage space based on the read request, and write the target data into the shared memory; and perform the read/write operation on the target physical storage space based on a read/write request that is in the at least one storage queue and that is from a user space program in the first virtualization instance.

6. The host device according to any one of claims 1 to 5, wherein the security scanning program is a second virtualization instance deployed on the host device or another device.

7. A security scanning method for cloud computing, applied to a virtualization manager on a host device, wherein the virtualization manager manages a first virtualization instance on the host device, and the method comprises:

in a case that the first virtualization instance uses a security scanning service, creating a

shared queue and a shared memory for a security scanning program located external to the first virtualization instance, wherein the first virtualization instance has target physical storage space, and the target physical storage space supports the virtualization manager to perform a read/write operation on the target physical storage space in a multi-queue read/write manner; and

providing target data in the target physical storage space to the security scanning program based on the shared queue and the shared memory, for the security scanning program to perform security scanning on the host device based on the target data,

wherein the first virtualization instance performs the read/write operation on the target physical storage space through at least one storage queue.

8. The method according to claim 7, wherein the providing target data in the target physical storage space to the security scanning program based on the shared queue and the shared memory comprises:

receiving, based on the shared queue, a read request sent by the security scanning program, and reading the target data from the target physical storage space based on the read request; and

writing the target data into the shared memory, for the security scanning program to read the target data from the shared memory.

9. The method according to claim 7 or 8, wherein the method further comprises:

obtaining configuration information of the first virtualization instance, wherein the configuration information comprises indication information indicating that the first virtualization instance uses the security scanning service, and determining, based on the indication information, that the first virtualization instance uses the security scanning service;

or

receiving a command that is sent by the security scanning program and that indicates to create the shared queue and the shared memory, and upon receipt of the command, determining that the first virtualization instance uses the security scanning service, wherein the command is sent in a case that the security scanning program establishes a security scanning service binding relationship with the first virtualization instance.

10. The method according to any one of claims 7 to 9, wherein the creating a shared queue for a security scanning program located external to the first virtualization instance comprises:

in a case of using an Input/Output (I/O) virtualization protocol, executing a first boot program corresponding to the first virtualization instance, to start the first virtualization instance, and creating the shared queue for the security scanning program after the first

virtualization instance is started;

or

in a case of using a device-passthrough-based non-volatile memory express (NVMe) protocol, executing a second boot program corresponding to the first virtualization instance, to start the first virtualization instance, and creating the shared queue for the security scanning program by the second boot program during startup.

11. A security scanning method for cloud computing, applied to a security scanning program external to a first virtualization instance, wherein the method comprises:

in a case that a virtualization manager corresponding to the first virtualization instance creates a shared queue and a shared memory for the security scanning program, writing a read request into the shared queue, for the virtualization manager to read target data from target physical storage space of the first virtualization instance based on the read request, and writing the target data into the shared memory; and

reading the target data from the shared memory, and performing security scanning based on the target data,

wherein the target physical storage space supports the virtualization manager to perform a read/write operation on the target physical storage space in a multi-queue read/write manner, and the first virtualization instance performs the read/write operation on the target physical storage space through at least one storage queue.

12. A virtualization management apparatus, implementable as a virtualization manager on a host device, and configured to manage a first virtualization instance on the host device, wherein the apparatus comprises: a creation module and a providing module,

the creation module is configured to: in a case that the first virtualization instance uses a security scanning service, create a shared queue and a shared memory for a security scanning program located external to the first virtualization instance, wherein the first virtualization instance has target physical storage space, and the target physical storage space supports the virtualization manager to perform a read/write operation on the target physical storage space in a multi-queue read/write manner; and

the providing module is configured to provide target data in the target physical storage space to the security scanning program based on the shared queue and the shared memory, for the security scanning program to perform security scanning on the host device based on the target data, wherein the first virtualization instance performs the read/write operation on the target physical storage space through at least one storage queue.

13. A host device, comprising a memory and a processor, wherein the memory stores a

security scanning program, and the processor is coupled to the memory and configured to execute the security scanning program, to implement steps in the method according to any one of claims 7 to 11.

14. A computer-readable storage medium storing a computer program, wherein when the computer program is executed by a processor, the processor is caused to implement steps in the method according to any one of claims 7 to 11.

15. A computer program product, comprising a computer program, wherein when the computer program is executed by a processor, steps in the method according to any one of claims 7 to 11 are implemented.