

基于公有云技术的服务器系统及其访问方法

本申请要求于2024年01月12日提交的申请号为202410050423.8、发明名称为“管理虚拟机的方法”的中国专利申请的优先权，以及于2024年04月29日提交的申请号为202410547732.6、发明名称为“基于公有云技术的服务器系统及其访问方法”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本申请涉及云服务技术领域，特别涉及一种基于公有云技术的服务器系统及其访问方法。

背景技术

随着云计算的应用越来越广泛，云上服务器上使用的外部设备呈现为增加的趋势。在外部设备逐渐增加的趋势下，执行地址翻译的重映射设备需要记录的GPA与HPA的对应关系的数量随之增加，但由于重映射设备能够记录的GPA与HPA的对应关系的数量是固定，导致部分GPA与HPA的对应关系可能没有被记载，则当地址翻译需要基于未被记载的GPA与HPA的对应关系执行时，就无法实现GPA与HPA之间的地址翻译，导致重映射设备的地址翻译性能下降，进而影响外部设备的操作性能。

发明内容

本申请提供了一种基于公有云技术的服务器系统及其访问方法。本申请有效地提高了服务器系统的地址翻译性能，有助于提高外部设备的操作性能（如DMA性能）和对内存进行管理的效率。本申请提供的技术方案如下：

第一方面，本申请提供了一种基于公有云技术的服务器系统。该服务器系统包括服务器和插置于服务器的外部设备。服务器和外部设备建立有连接通道。服务器中设置有虚拟机和虚拟机管理器，虚拟机管理器用于为虚拟机提供虚拟硬件，虚拟硬件包括虚拟外部设备和虚拟内存，虚拟外部设备基于外部设备进行设备模拟得到，虚拟内存基于为外部设备配置的内存进行设备模拟得到，虚拟机中设置有虚拟外部设备的虚拟设备驱动，虚拟设备驱动用于在虚拟外部设备访问虚拟内存的目标客户机物理地址时，向外部设备发送访问请求，访问请求携带有目标客户机物理地址。外部设备用于基于访问请求，查询客户机物理地址与主机物理地址的第一对应关系，得到目标客户机物理地址对应的目标主机物理地址，获取目标主机物理地址记载的目标数据，并向虚拟设备驱动发送目标数据。

由上可知，该服务器系统相当于将地址翻译功能下沉至外部设备中，使得外部设备对为虚拟机分配的内存进行直接管理，分担对访问请求中目标客户机物理地址进行地址翻译的任务。这样一来，就不需要服务器中的IOMMU/SMMU等地址翻译设备执行该地址翻译的任务，不需要服务器中的IOMMU/SMMU等地址翻译设备的TLB记录该外部设备的虚拟外部设备访问的目标客户机物理地址的对应关系，减少了该TLB中需要记录的对应关系的数量，缓解了因该TLB的页表条目较少和外部设备数量过多的矛盾，降低了地址翻译过程中出现TLB缺页问题的概率，有效地提高了服务器系统的地址翻译性能，有助于提高外部设备的操作性能（如DMA性能）和对内存进行管理的效率。

在一种可能的实现方式中，服务器中还设置有地址管理模块。外部设备还用于在第一对应关系未记载有目标客户机物理地址对应的目标主机物理地址时，向地址管理模块发送指示匹配失败的消息，消息携带有目标客户机物理地址；地址管理模块用于向虚拟机管理器转发消息；虚拟机管理器用于基于消息，在内存中为目标客户机物理地址分配内存块，建立内存块的主机物理地址与目标客户机物理地址的第二对应关系，并向地址管理模块发送第二对应关系；地址管理模块用于基于第二对应关系更新第一对应关系。

由上可知，本申请提供的服务器系统能够在基于目标客户机物理地址查询对应关系未命中时执行相关的处理，如执行缺页异常处理，以解决未命中的问题。这样一来，在虚拟机启动之前就可以免pin内存，能够解除全局内存锁定。在云场景中外部设备直通虚拟机的情况下，内存免pin使得能够在虚拟机用户无感知（不对虚拟机系统有特殊要求）的情况下对虚拟机内存进行超分复用，能够提升服务器内存的整体利用率，并保证服务器的性能不下降，该特点在内存压力大的场景中表现的尤其明显。同时，由于内存免pin，使得无需因虚拟机预占内存存在云场景中配置大量的内存，能够降低云场景中内存的使用成本，

解决公有云内存成本逐渐增加的问题，进而增强公有云的市场竞争力。

在一种可能的实现方式中，服务器中还设置有地址管理模块。虚拟机管理器用于基于虚拟机的规格，为虚拟机进行设备模拟得到虚拟内存，并配置虚拟内存的客户机物理地址空间，建立内存中为外部设备配置的内存块的主机物理地址与客户机物理地址空间中客户机物理地址的对应关系，并向地址管理模块发送对应关系。

在一种可能的实现方式中，外部设备的配置信息指示外部设备支持已经过翻译的请求，且访问请求指示访问请求携带的地址为主机物理地址。这样一来，当服务器兼容原有地址翻译功能时，能够保证外部设备能力的访问请求能够由外部设备自身处理，不由服务器原有的地址翻译功能处理，即绕过（bypass）服务器原有的地址翻译功能。

在一种可能的实现方式中，为外部设备配置的内存包括以下一种或多种：插置于服务器的内存或服务器的共享内存。

第二方面，本申请提供了一种基于公有云技术的服务器系统的访问方法。该服务器系统包括服务器和插置于服务器的外部设备，服务器和外部设备建立有连接通道，服务器中设置有虚拟机和虚拟机管理器，虚拟机管理器用于为虚拟机提供虚拟硬件，虚拟硬件包括虚拟外部设备和虚拟内存，虚拟外部设备基于外部设备进行设备模拟得到，虚拟内存基于为外部设备配置的内存进行设备模拟得到，虚拟机中设置有虚拟外部设备的虚拟设备驱动，该方法包括：虚拟设备驱动在虚拟外部设备访问虚拟内存的目标客户机物理地址时，向外部设备发送访问请求，访问请求携带有目标客户机物理地址；外部设备基于访问请求，查询客户机物理地址与主机物理地址的第一对应关系，得到目标客户机物理地址对应的目标主机物理地址，获取目标主机物理地址记载的目标数据，并向虚拟设备驱动发送目标数据。

在一种可能的实现方式中，服务器中还设置有地址管理模块。该方法还包括：外部设备在第一对应关系未记载有目标客户机物理地址对应的目标主机物理地址时，向地址管理模块发送指示匹配失败的消息，消息携带有目标客户机物理地址；地址管理模块向虚拟机管理器转发消息；虚拟机管理器基于消息，在内存中为目标客户机物理地址分配内存块，建立内存块的主机物理地址与目标客户机物理地址的第二对应关系，并向地址管理模块发送第二对应关系；地址管理模块基于第二对应关系更新第一对应关系。

在一种可能的实现方式中，服务器中还设置有地址管理模块。该方法还包括：虚拟机管理器基于虚拟机的规格，为虚拟机进行设备模拟得到虚拟内存，并配置虚拟内存的客户机物理地址空间，建立内存中为外部设备配置的内存块的主机物理地址与客户机物理地址空间中客户机物理地址的对应关系，并向地址管理模块发送对应关系。

在一种可能的实现方式中，外部设备的配置信息指示外部设备支持已经过翻译的请求，且访问请求指示访问请求携带的地址为主机物理地址。

在一种可能的实现方式中，为外部设备配置的内存包括以下一种或多种：插置于服务器的内存或服务器的共享内存。

第三方面，本申请提供了一种计算设备，包括存储器和处理器，存储器存储有程序指令，处理器运行程序指令以执行本申请第二方面以及其任一种可能的实现方式中提供的方法。

第四方面，本申请提供了一种计算设备集群，包括多个计算设备，多个计算设备包括多个处理器和多个存储器，多个存储器中存储有程序指令，多个处理器运行程序指令，使得计算设备集群执行本申请第二方面以及其任一种可能的实现方式中提供的方法。

第五方面，本申请提供了一种计算机可读存储介质，该计算机可读存储介质为非易失性计算机可读存储介质，该计算机可读存储介质包括程序指令，当程序指令在计算设备上运行时，使得计算设备执行本申请第二方面以及其任一种可能的实现方式中提供的方法。

第六方面，本申请提供了一种包含指令的计算机程序产品，当计算机程序产品在计算机上运行时，使得计算机执行本申请第二方面以及其任一种可能的实现方式中提供的方法。

附图说明

图1是本申请实施例提供的一种GPA与HPA的对应示意图；

图2是本申请实施例提供的一种本申请涉及的实施场景的结构示意图；

图3是本申请实施例提供的一种云资源部署的示意图；

图4是本申请实施例提供的一种服务器系统的结构示意图；

图5是本申请实施例提供的一种服务器系统的逻辑示意图；
图6是本申请实施例提供的另一种服务器系统的逻辑示意图；
图7是本申请实施例提供的又一种服务器系统的逻辑示意图；
图8是本申请实施例提供的一种查询对应关系的示意图；
图9是本申请实施例提供的另一种服务器系统的结构示意图；
图10是本申请实施例提供的一种上下文条目的示意图；
图11是本申请实施例提供的一种TLP头格式的示意图；
图12是本申请实施例提供的一种用户态与内核态的交互示意图；
图13是本申请实施例提供的一种ioctl命令格式的示意图；
图14是本申请实施例提供的一种内核态与外部设备的交互示意图；
图15是本申请实施例提供的一种服务器系统包括多个外部设备的示意图；
图16是本申请实施例提供的一种基于公有云技术的服务器系统的访问方法的流程图；
图17是本申请实施例提供的另一种基于公有云技术的服务器系统的访问方法的流程图；
图18是本申请实施例提供的一种计算设备的结构示意图；
图19是本申请实施例提供的一种计算设备集群的结构示意图；
图20是本申请实施例提供的另一种计算设备集群的结构示意图。

具体实施方式

为使本申请的目的、技术方案和优点更加清楚，下面将结合附图对本申请实施方式作进一步地详细描述。

为便于理解，下面先对本申请实施例中涉及的技术和背景进行介绍。

云数据中心是基于互联网（Internet）网络，为集中式收集、存储、处理和发送数据的设备提供运行维护的设施以及相关的服务体系。从概念上可以将其理解为公共的商业化的Internet“机房”，同时它也是一种信息技术（information technology, IT）专业服务，是IT工业的重要基础设施。云数据中心不仅是一个服务概念，而且是一个网络的概念，它构成了网络基础设施的一部分，就像骨干网、接入网一样，提供了一种高端的数据传输（data delivery）的服务和高速接入服务。

云计算（cloud computing）：云计算是分布式计算的一种，指对大量计算资源和存储资源进行统一管理和调度，向用户提供按需服务的网络。该计算资源和存储资源均通过设置在数据中心的计算设备提供。并且，云计算可以为用户提供多种类型的服务类型，例如，可以提供基础设置即服务（infrastructure as a service, IaaS）、平台即服务（platform as a service, PaaS）和软件即服务（software as a service, SaaS）等。基础设施即服务的作用是提供虚拟机或者其他资源作为服务提供给租户。平台即服务的作用是将一个开发平台作为服务提供给租户。软件即服务的作用是将应用程序（application, App，也称应用）作为服务提供给客户。

主机（physical machine, PM）：用于承载虚拟化技术的物理资源。主机也称物理机。通常的，用于部署虚拟实例的主机为物理服务器。物理机具有多个物理设备。例如，物理服务器具有处理器和存储器等物理设备。一台主机中可以部署多个虚拟实例，部署在同一台主机上的多个虚拟实例共享主机的物理资源。按照不同的使用情况，一台主机中部署的多个虚拟实例可选属于同一个租户，或者分别属于不同的租户。

虚拟化（virtualization）是一种资源管理技术。虚拟化能够将主机的各种实体资源，如计算资源、网络资源及存储资源等，予以抽象、转换后呈现出来，以打破主机的实体结构间的不可切割的障碍，使租户可以采用比原本的组态更好的方式来应用这些资源。通过虚拟化得到的资源称为虚拟化资源，虚拟化资源不受现有实体资源的架设方式、设置地域或物理组态所限制。

虚拟化资源通常以虚拟实例的形式向租户提供。物理服务器的硬件资源能以虚拟实例的粒度给一个或多个租户使用。不同虚拟实例之间相互隔离，使租户能在安全隔离的前提下方便灵活地使用物理资源，并且能极大提升物理资源的利用率。在本申请中，虚拟实例可以为虚拟机、容器或独立进程（例如函数）等。虚拟实例也可称为云服务器（elastic compute service, ECS）、弹性实例（不同的云服务提供商有不同的叫法）。

虚拟机（virtual machine, VM）指通过软件模拟的具有完整硬件系统功能的、运行在一个完全隔离环

境中的完整的计算机系统。在服务器中能够完成的工作在虚拟机中都能够实现。在服务器中创建虚拟机时，需要将实体机的部分硬盘和内存容量作为虚拟机的硬盘和内存容量，每个虚拟机都有独立的硬盘和操作系统，虚拟机的租户可以像使用服务器一样对虚拟机进行操作。不同虚拟机中的运行环境（如虚拟机应用、操作系统和虚拟硬件）是完全隔离的，不同虚拟机之间要进行通信需经过虚拟管理器转发网络报文。

容器提供了一个轻量级的虚拟运行环境。容器可以通过将租户的应用程序的所有代码、库和依赖关系等打包成镜像得到。执行镜像时，镜像运行在虚拟运行环境中。此时容器是镜像的运行实例，类似于一个轻量级的沙箱子，它可以被启动、开始、停止、删除。该镜像不会与其他镜像共享宿主机的内存、处理器（如中央处理器（central processing unit, CPU））以及磁盘等资源，实现了该镜像与主机之间、该镜像与其他镜像之间的容器隔离，保证了容器内的进程不能监控容器外的任何进程或资源。容器的基础设施可以是服务器的硬件，或者云上的虚拟机（即虚拟机中也可以部署容器），操作系统采用linux内核，支持namespace和cgroup，其中namespace用来实现进程之间的隔离，cgroup用来实现进程资源的分配，资源具体为分给该进程的虚拟处理器和内存。容器引擎类似虚拟机管理器，运行在操作系统中，用来管理容器。跟虚拟机自带操作系统的特性相比，容器不存在操作系统，容器是作为进程运行在宿主机的操作系统中，所以容器的启动速度比虚拟机快，特别适合轻量级的应用，并且一个宿主机可以同时运行上千个容器（进程）。

直接存储器访问（direct memory access, DMA）：也可以称为直接内存操作或者组数据传输方式，是指外部设备不经过计算机的中央处理器（central processing unit, CPU），而直接从计算机的内存存取数据的数据交互模式。在DMA模式下传输数据时，由计算机的CPU向DMA控制器下达指令，以指示DMA控制器控制数据的传输，DMA控制器在完成数据传输后向CPU反馈传输完成的信息。可以看出，在该DMA传输数据的过程中，无需计算机的CPU执行传输操作，可以省去计算机的CPU取指令、取数、送数等操作，减轻了计算机的CPU的资源占有率，能够节省系统资源。

DMA可以包括远程DMA（remote direct memory access, RDMA）和本地DMA。RDMA是指通过网络将数据直接从一台计算机的内存传输到另一台计算机，无需双方操作系统的介入的数据传输方式。本地DMA是指无需通过网络进行的数据传输方式。

外围组件互连快速（peripheral component interconnect express, PCIe）总线：一种高速串行计算机扩展总线。

内存（memory）：也可以称为内存存储器或者主存储器，其作用是暂时存放CPU中的运算数据，以及与硬盘等外部存储器交换的数据。

外部设备（external device），简称“外设”。计算机系统中输入、输出设备（包括外存储器）的统称。对数据和信息起着传输、转送和存储的作用。是计算机系统中的重要组成部分。外围设备涉及到主机以外的任何设备。外围设备是附属的或辅助的与计算机连接起来的设备。外围设备能扩充计算机系统。外部设备例如是网卡、外部存储器（如硬盘、磁盘和显卡）、图形处理器（graphics processing unit, GPU）、智能处理单元（intelligent processing unit, IPU）和数据流处理器单元（data Processing Unit, DPU）等。其中，网卡也可以称为网络接口控制器（network interface controller, NIC）、网络适配器、或者局域网接收器，是一种被设计为允许主机或计算设备在网络上进行通信的计算机硬件。

内存虚拟化技术的目的是为了提供给虚拟机一个从0地址开始的连续物理内存空间，在虚拟机之间有效隔离和调度内存资源。内存虚拟化技术主要涉及客户机虚拟地址（guest virtual address, GVA）→客户机物理地址（guest physical address, GPA）→宿主机虚拟地址（host virtual address, HVA）→宿主机物理地址（host physical address, HPA）的转换。

而在虚拟化技术中，物理主机上往往运行有多个虚拟机，每个虚拟机均以为自己独占物理主机的内存空间，因此虚拟机以GPA表示虚拟机所拥有的内存空间，其中该内存空间被虚拟机认为是连续的（即可以理解为虚拟机认为自己拥有一个完整的物理内存条）。

GVA是虚拟机的操作系统对GPA进行映射而形成的地址，虚拟机的操作系统将GVA提供给设置在虚拟机的操作系统上的进程或应用软件使用，虚拟机的操作系统记录GVA到GPA的映射关系，GVA到GPA的转化是由虚拟机的操作系统的页表实现的。

HPA是实际的物理内存地址，HVA是宿主机的操作系统对HPA进行映射而形成的地址，宿主机的操

作系统将HVA提供给操作系统上的进程（例如虚拟机）使用，宿主机的操作系统记录HVA到HPA的映射关系，HVA到HPA的转化是由宿主机的操作系统的页表实现的。

在图1中，虚拟机1和虚拟机2设置在同一个服务器（下称宿主机）中，宿主机的虚拟机管理器（又可称为Hypervisor）将虚拟机1的GPA地址范围设置为0-5GB，其对应于物理内存上的HPA地址范围1.5GB-4.5GB以及6.5GB-8.5GB。并且，宿主机的虚拟机管理器将虚拟机2的GPA地址范围设置为0-4GB，对应于物理内存上的HPA地址范围9GB-11GB以及13GB-15GB。因此虚拟机1独占使用0-5GB的GPA地址范围，虚拟机2独占使用0-4GB的GPA地址范围。该0-5GB的GPA地址范围和该0-4GB的GPA地址范围均可对应至物理内存上的不同的HPA地址范围，从而实现虚拟机内存的隔离。

其中，GPA地址范围与上文的虚拟机规格相关。例如，租户可以在云管理平台中设置虚拟机规格中内存大小为5G的虚拟机1，此时虚拟机管理器被云管理平台通知需创建GPA为0-5G的虚拟机1。租户可以在云管理平台中设置虚拟机规格中内存大小为4G的虚拟机2，此时虚拟机管理器被云管理平台通知需创建GPA为0-4G的虚拟机2。

目前，在云场景中，外部设备直通给虚拟机使用的情况下，Hypervisor需要一次性地向虚拟机分配内存，并将分配给虚拟机的内存块锁定（pin），使得虚拟机在启动之前需要对预占的内存进行初始化，保证在虚拟机的生命周期内分配给虚拟机的内存块不能给其他内存使用者使用。同时，Hypervisor会为虚拟机分配GPA空间，并建立该GPA空间中多个GPA与为虚拟机分配的内存块的HPA的对应关系。GPA与HPA的对应关系由宿主机的操作系统维护。

其中，直通技术是服务器的外部设备（如PCIe设备）支持的虚拟化技术。直通是指跳过Hypervisor将外部设备直接提供给虚拟机使用。直通技术支持外部设备设置多个虚拟功能（virtual function, VF），外部设备将存储资源或网络资源设置在一个或多个VF中，并将一个或多个VF提供给一个租户的虚拟机，该虚拟机可直接使用一个或多个VF提供的资源。举例而言，外部设备将一个存储资源中的划分的一个40G的逻辑磁盘与一个VF绑定，并将该VF提供给虚拟机1，虚拟机1在挂载VF之后，可如访问本地磁盘一样访问逻辑磁盘。或者，外部设备将网络资源中的一个网卡接口eth0与一个VF绑定，并将该VF提供给虚拟机1，虚拟机1在挂载VF之后，可如使用本地网卡一样使用eth0。

虚拟机上运行的虚拟机的操作系统通常不知道Hypervisor为虚拟机分配的内存块的HPA，但虚拟机的操作系统知道为虚拟机分配的GPA空间。当虚拟机通过直通给其使用的外部设备访问为其分配的内存块时，其会向宿主机的操作系统发送DMA请求，DMA请求携带有需要访问的GPA。宿主机的操作系统接收DMA请求后，基于DMA请求携带的GPA查询GPA与HPA的对应关系，得到该虚拟机请求访问的GPA对应的HPA，并读取该HPA指示的内存块记载的数据，然后向虚拟机提供该数据。其中，宿主机的操作系统基于GPA得到对应的HPA的过程称为地址翻译过程或者地址重映射过程，该过程是将DMA请求的GPA，翻译成外部设备真正能用的主机物理地址（即HPA），是对内存进行管理的过程。用于执行地址重映射过程的设备称为重映射设备（remapping hardware）。目前重映射设备通常有输入/输出内存管理单元（input/output memory management unit, IOMMU）和系统内存管理单元（system memory management unit, SMMU）。IOMMU和SMMU均设置在主机中。IOMMU和SMMU通过旁路转换缓冲（translation lookaside buffer, TLB）记录GPA与HPA的对应关系。该TLB的页表条目（page table entry, PTE）的数量通常是固定的，即指定IOMMU和SMMU能够记录的GPA与HPA的对应关系的数量是固定的。其中，PTE由一个有效位和n位地址字段组成，地址字段记载GPA对应的HPA，即GPA对应的内存页的起始位置。

在外部设备逐渐增加的趋势下，IOMMU和SMMU需要记录的GPA与HPA的对应关系的数量随之增加，但由于IOMMU和SMMU能够记录的GPA与HPA的对应关系的数量是固定，导致部分GPA与HPA的对应关系可能没有被记载，则当地址翻译需要基于未被记载的GPA与HPA的对应关系执行时，就无法实现GPA与HPA之间的地址翻译，导致IOMMU和SMMU的地址翻译性能下降，进而影响外部设备的操作性能。例如，在当前云场景下，主机的外部设备不断增加，外部设备的资源与DMA操作不断增加，但IOMMU/SMMU能够记录的GPA与HPA的对应关系的数量较少，导致IOMMU/SMMU的地址翻译性能下降。以IOMMU为例，当前x86 IOMMU TLB仅有64个页表条目，在云的大规格虚拟功能（virtual function, VF）场景下，需要IOMMU TLB记录的页表条目通常大于64个，导致部分GPA与HPA的对应关系没有被记载，当需要查询这部分GPA对应的HPA时，地址翻译性能就会受到影响。例如，当VF数量超过64后，会发生严重的TLB缺页，在VF数量为128时，地址翻译性能下降40%左右。

目前，地址翻译功能不支持缺页处理，这样就需要虚拟机直通的外部设备预占内存。由于外部设备

预占的内存处于锁定状态，被预占的内存不能够被其他内存使用者使用，这就导致内存的利用率较低。例如，当前云场景下，在外部设备直通虚拟机的情况下，Hypervisor需要一次性将内存分配成功并pin分配的内存存在主机上，已经分配的内存不能被复用，导致内存的整体利用率较低。同时，虚拟机在启动之前需要对预占的内存进行初始化，还会导致虚拟机启动慢。

虽然，目前提出了一种虚拟IOMMU技术（coIOMMU）。但是该技术需要修改虚拟机的操作系统才能实现，导致该技术在公有云落地具有较大困难。

鉴于此，本申请提供了一种基于公有云技术的服务器系统及其访问方法。该服务器系统包括服务器和插置于服务器的外部设备。服务器和外部设备建立有连接通道。该服务器中设置有虚拟机和虚拟机管理器。虚拟机管理器用于为虚拟机提供虚拟硬件。虚拟硬件包括虚拟外部设备和虚拟内存。虚拟外部设备基于外部设备进行设备模拟得到。虚拟内存基于为外部设备配置的内存进行设备模拟得到。虚拟机中设置有虚拟外部设备的虚拟设备驱动。虚拟设备驱动用于在虚拟外部设备访问虚拟内存的目标客户机物理地址时，向外部设备发送访问请求。访问请求携带有目标客户机物理地址。外部设备用于基于访问请求，查询客户机物理地址与主机物理地址的第一对应关系，得到目标客户机物理地址对应的目标主机物理地址，获取目标主机物理地址记载的目标数据，并向虚拟设备驱动发送目标数据。

由上可知，该服务器系统相当于将地址翻译功能下沉至外部设备中，使得外部设备对为虚拟机分配的内存进行直接管理，分担对访问请求中目标客户机物理地址进行地址翻译的任务。这样一来，就不需要服务器中的IOMMU/SMMU等地址翻译设备执行该地址翻译的任务，不需要服务器中的IOMMU/SMMU等地址翻译设备的TLB记录该外部设备的虚拟外部设备访问的目标客户机物理地址的对应关系，减少了该TLB中需要记录的对应关系的数量，缓解了因该TLB的页表条目较少和外部设备数量过多的矛盾，降低了地址翻译过程中出现TLB缺页问题的概率，有效地提高了服务器系统的地址翻译性能，有助于提高外部设备的操作性能（如DMA性能）和对内存进行管理的效率。

需要说明的是，本申请中虽然以虚拟机使用外部设备为例进行说明，但不排除其他类型的虚拟实例使用外部设备，本申请实施例对其不做具体限定。该其他类型的虚拟实例例如是容器等。当其他类型的虚拟实例使用外部设备时，其通过外部设备访问内存的实现方式请相应参考本申请实施例中虚拟机通过外部设备访问内存的实现方式，本文不再对其进行赘述。

本文从实施场景、方法流程、硬件装置、软件装置等多个角度，对本申请的技术方案进行详细介绍。下面先对本申请实施例的实施场景举例说明。

图2是本申请实施例提供的一种本申请涉及的实施场景的结构示意图。如图2所示，该实施场景包括：数据中心1和客户端2。数据中心1和客户端2之间能够通过网络建立通信连接。可选的，该网络可以为互联网，还可以为其它网络，本申请实施例不作限定。租户能够通过客户端2与数据中心1进行交互。例如，租户可以通过客户端2向数据中心1发送虚拟实例创建请求等信息。数据中心1用于基于客户端2发送的信息进行响应。

数据中心1中部署有云服务提供商拥有的大量基础设施，如计算资源、存储资源和网络资源等。例如，计算资源可以是能够提供计算能力的计算设备，如服务器和具有计算能力的GPU、IPU和DPU等外部设备。存储资源可以是能够提供存储能力的设备，如硬盘、磁盘和显卡等外部设备。网络资源可以是能够提供网络传输能力的设备，如网卡等外部设备。如图2所示，数据中心1包括云管理平台和基础设施（图2中未示出）。云管理平台和基础设施之间通过数据中心内部网络连接。云管理平台用于管理基础设施。基础设施用于提供公有云服务。基础设施包括多个服务器。云服务可选部署在服务器中。租户能够通过其使用的客户端2向服务器发送云服务请求，服务器能够对该云服务请求进行处理，并基于经过处理的云服务请求提供云服务。

云管理平台从逻辑功能上能够划分为：租户控制台、计算管理服务、网络管理服务、存储管理服务、鉴权服务和镜像管理服务。租户控制台提供界面或应用程序接口（application program interface, API）与租户交互。计算管理服务用于管理运行虚拟实例的服务器以及裸金属服务器。网络管理服务用于管理网络服务（如网关、防火墙等）。存储管理服务用于管理存储服务（如数据桶服务）。鉴权服务用于管理租户的账号和密码。镜像管理服务用于管理虚拟实例的镜像。

在图2所示的实施场景中，一个数据中心中设置有多台服务器。服务器包括硬件层和软件层。硬件层

为服务器的常规配置。硬件层部署有处理器、内存、网卡、磁盘和总线等硬件设备。软件层包括安装并运行在服务器上的操作系统。相对虚拟机的操作系统可称为宿主机操作系统。宿主机操作系统中运行有虚拟机管理器（又可称为Hypervisor）。虚拟机管理器的作用是实现虚拟机的计算虚拟化、网络虚拟化以及存储虚拟化，并负责管理虚拟机。

虚拟机管理器中运行有云管理平台客户端。云管理平台客户端能够接收云管理平台发送的控制面命令，根据控制面控制命令在服务器上创建虚拟实例，并对虚拟实例进行全生命周期管理。例如，云管理平台客户端能够实时检测所在服务器的硬件资源的使用情况并上报至云管理平台。云管理平台确认要在某个服务器上创建虚拟实例时，会向该服务器上的云管理平台客户端发送虚拟实例创建命令，云管理平台客户端接收该命令后，在该服务器上创建虚拟实例。这样一来，租户可通过云管理平台在数据中心中创建、管理、登录和操作虚拟实例。

服务器可用于运行不同规格的虚拟机。虚拟机规格分为：通用计算型、内存优化型、超大内存型等等，每个类型下面还有具体规格，租户选择好虚拟机规格后，云管理平台在数据中心中选择支持该规格的服务器，并确定该服务器上具有足够的空闲硬件资源，然后在该服务器上创建设置有该规格的虚拟机。通过云管理平台对服务器进行配置，能够实现对服务器硬件资源的分析与规划，根据服务器的硬件性能规划物理硬件的对应计算产品，如规划不同规格的虚拟机，以满足不同租户的差异化诉求。并且，根据不同规格的虚拟机的性能差异，能够差异化定价策略。例如，将高性能规格的虚拟实例以更高的价格对外售卖，将普通性能规格的虚拟实例以较低价格对外售卖，以便于租户按需购买虚拟实例。

在一种实现方式中，如图3所示，数据中心中基础资源的位置可以通过云资源部署区域（region）和可用区（availability zone, AZ）进行描述。租户可选基于特定的region、AZ中的资源部署云服务。其中，region是从地理位置和网络时延维度进行划分的。同一个region内使用同一个资源池，可以理解为共享弹性计算、块存储、对象存储、虚拟私有云（virtual private cloud, VPC）网络、弹性网络互连协议（elastic internet protocol, EIP）地址和镜像等公共服务。region分为通用region和专属region。通用region指面向公共租户提供通用云服务的region。专属region指承载同一类业务或面向特定租户提供业务服务的专用region。一个region通常包括多个AZ。一个region中的多个AZ间通过高速光纤相连，以满足租户跨AZ构建高可用性系统的需求。一个AZ是一个或多个图3所示的数据中心的集合。AZ内的计算、网络和存储等资源在逻辑上被划分成多个集群。

租户可通过其使用的客户端2向云管理平台发送指令，以在服务器中创建、管理、登录和操作虚拟实例，并使用该虚拟实例提供的云服务。示例地，云管理平台能够提供访问接口。访问接口可选以界面或API的形式提供。租户可操作客户端远程接入访问接口，以在云管理平台注册云账号和密码，并使用云账号和密码登录云管理平台。云管理平台还能够对云账号和密码进行鉴权。鉴权成功后，租户可进一步在云管理平台选择并付费购买特定规格（处理器、内存、磁盘）的虚拟实例。租户付费购买虚拟实例成功后，云管理平台向租户提供所购买的虚拟实例的远程登录账号和密码。租户可使用远程登录账号和密码，在客户端远程登录该虚拟实例，在该虚拟实例中安装并运行租户的应用，以通过该应用实现租户的业务。

客户端2可选是计算机、个人电脑、膝上型计算机、移动电话、智能手机、平板电脑、云主机、便携式移动终端、多媒体播放器、电子书阅读器、可穿戴式设备、智能家电、人工智能设备、智能可穿戴设备、智能车载设备或物联网设备等。

在一种实现方式中，本申请实施例提供的基于公有云技术的服务器系统的访问方法，可以通过数据中心1中的计算设备运行可执行程序实现。可选地，本申请实施例提供的基于公有云技术的服务器系统的访问方法可选应用于基于公有云技术的服务器系统的访问系统。该基于公有云技术的服务器系统的访问系统部署于云管理平台管理的服务器中。该基于公有云技术的服务器系统的访问系统通过运行本申请实施例提供的基于公有云技术的服务器系统的访问方法的可执行程序，能够实现本申请实施例提供的基于公有云技术的服务器系统的访问方法。并且，实现该基于公有云技术的服务器系统的访问方法的可执行程序，可选以应用程序安装包的形式呈现，服务器安装该应用程序安装包后，能够通过运行其中的可执行程序，实现本申请实施例提供的基于公有云技术的服务器系统的访问方法。

应当理解的是，以上内容是对本申请实施例提供的基于公有云技术的服务器系统的访问方法的实施场景的示例性说明，并不构成对于该基于公有云技术的服务器系统的访问方法的实施场景的限定，本领域普通技术人员可知，随着业务需求的改变，其实施场景可以根据应用需求进行调整，本申请实施例对其不做一一列举。

下面对本申请实施例提供的基于公有云技术的服务器系统的实现方式进行说明。图4为本申请实施例提供的一种服务器系统的示意图。如图4所示，该服务器系统1包括服务器11和插置于服务器的外部设备12。服务器和外部设备建立有连接通道。示例地，服务器的主板上具有设备插槽，外部设备具有与该插槽匹配的接口，外部设备通过该接口插置于服务器的插槽中，外部设备和服务器之间可以通过总线等方式连接。可选地，本申请中的外部设备可选为GPU、IPU、DPU、硬盘、磁盘、显卡和网卡等，且本申请的外部设备还可为此处举例以外的其他类型的设备，本申请实施例对其不做一一举例。

服务器具有硬件111。服务器中设置有虚拟机管理器112和虚拟机113。虚拟机管理器用于基于服务器具有的硬件和服务器的外部设备，为虚拟机提供虚拟硬件1121。虚拟硬件包括虚拟外部设备1121a和虚拟内存1121b。虚拟外部设备由虚拟机管理器基于外部设备进行设备模拟得到。虚拟内存由虚拟机管理器基于为外部设备配置的内存进行设备模拟得到。应当理解的是，虚拟机管理器为虚拟机提供的虚拟硬件还有其他类型的虚拟硬件，如虚拟处理器等，以满足虚拟机的性能要求，此处不对其进行一一举例。虚拟机中设置有虚拟外部设备的虚拟设备驱动。该虚拟设备驱动用于实现虚拟外部设备的访问过程。虚拟设备驱动用于在虚拟外部设备访问虚拟内存的目标客户机物理地址时，向外部设备发送访问请求。访问请求携带有目标客户机物理地址。例如，虚拟机中设置有应用程序，应用程序运行的过程中需要使用外部设备的能力时，在外部设备直通给虚拟机使用的情况下，虚拟外部设备会访问虚拟内存，以向应用程序提供该外部设备的能力。虚拟外部设备访问虚拟内存的操作，会触发访问请求。虚拟设备驱动能够获取该访问请求，并向外部设备发送该访问请求，以便于外部设备基于访问请求对内存进行访问，实现虚拟外部设备访问虚拟内存的目的。

其中，本申请中的内存可选为服务器上配置的内存（即本地内存），或者为与服务器建立有连接通道的其他计算设备的内存等。与服务器建立有连接通道的其他计算设备的内存，例如是部署在该其他计算设备上，且该其他计算设备向该服务器共享的内存，即共享内存。服务器能够通过远端访问或者高速通信总线访问该其他计算设备的内存。例如，服务器通过远程直接内存访问（remote direct memory access, RDMA）技术访问该其他计算设备的内存。高速通信总线可选为计算高速连线（compute express link, CXL）或灵衢总线（也称UB总线）。

外部设备接收到虚拟设备驱动发送的访问请求后，能够基于访问请求查询客户机物理地址与主机物理地址的第一对应关系，得到目标客户机物理地址对应的目标主机物理地址。然后外部设备从内存中获取目标主机物理地址中记载的目标数据，并向虚拟设备驱动发送目标数据。其中，虚拟设备驱动可选通过工作队列（work queue, WQ）向外部设备发送访问请求。相应的，外部设备能够按照从工作队列中获取工作队列元素（work queue element, WQE）的方式，从工作队列中获取该访问请求，然后在该访问请求中获取该访问请求携带的输入输出虚拟地址（input/output virtual address, IOVA），即得到目标客户机物理地址。

外部设备的地址翻译功能可选通过软件或者硬件实现。当该功能通过软件实现时，外部设备中需要配置实现该功能的可执行程序。当该功能通过硬件实现时，外部设备中需要配置该硬件。该硬件可称为设备内存管理单元（device memory manage unit, DMMU）。

图5、图6和图7为本申请实施例提供的几种从不同角度呈现的服务器系统的逻辑示意图。如图5所示，服务器系统按照逻辑可划分为服务器和外部设备，服务器又分为用户态和内核态。用户态中设置有虚拟机，虚拟机中运行有虚拟机的操作系统，该操作系统管理有虚拟外部设备（图中未示出）、虚拟设备驱动和虚拟内存。内核态管理有内存和DMMU的设备驱动DMMU driver，且记载有第一对应关系的映射表（map table）存储在该内核态中。该图5中的内存为服务器的本地内存。Hypervisor包括面向用户态的部分和面向内核态的部分。外部设备中设置有DMMU和DMA使能组件（combo DMA）。虚拟外部设备触发访问请求后，虚拟设备驱动会获取该访问请求，并向DMMU发送该访问请求。DMMU接收到访问请求后，基于访问请求携带的GPA对映射表进行查询，在查询到GPA对应的HPA后，向DMA使能组件提供HPA。DMA使能组件基于该HPA读取该HPA指示的内存块记载的数据，并向虚拟机提供该数据。图6与图5不同的是，内存有多种实现形态。按照内存的作用，内存可包括多个部分。例如，如图6所示，内存包括真实分配的内存、按需页面内存（page on demand memory, pod mem）、延迟分配内存（copy on write memory, cow mem）和换出换入内存（swapped memory, swapped mem）。当内存为双倍速率同步动态随机存储器（double data rate synchronous dynamic random access memory, DDR SDRAM）时，真实分配

的内存又称为real DDR。图7与图5不同的是，图7中的内存为部署在其他服务器（如图7中的远端节点）上的共享内存。DMA使能组件在获取HPA后，若HPA指示的内存块为共享内存的内存块，DMA使能组件采用RDMA方式或者通过高速通信总线访问共享内存，以读取该HPA指示的内存块记载的数据。DMA使能组件在获取HPA后，若HPA指示的内存块为共享内存的内存块，DMA使能组件通过高速通信总线访问共享内存，以读取该HPA指示的内存块记载的数据。并且，图7中服务器与外部设备之间通过CXL/UB建立连接通道，外部设备与远端节点之间通过RDMA/UB实现对共享内存的互访。另外，图7中示出了虚拟机中设置的应用程序出发访问后，向DMMU提供GPA的过程。DMMU中设置有页表缓存（cache）和地址映射查找组件。地址映射查找组件用于基于GPA查询对应关系。页表cache其用于存储EPT中使用较频繁的对应关系。地址映射查找组件在进行地址翻译时，可先在该页表cache中查找对应关系，当在该页表cache中查询不到时，再到地址管理模块或EPT中进行查询。当在页表cache中能够查询到目标GPA对应的目标HPA时，能够进一步加速地址翻译。该图7中还派只有大内存管理，其用于对为外部设备配置的内存进行管理。

在一种可能的实现方式中，虚拟机管理器在为虚拟机提供虚拟外部设备时，可选以VF的方式向虚拟机提供资源。不同的VF通过VF的标识（即VFID）进行区分。并且，为外部设备分配的内存也按照VF进行区分，则第一对应关系可选以VF为粒度进行管理。例如，当第一对应关系以页表的方式存储时，每个页表用于记载属于一个VF的客户机物理地址与主机物理地址的对应关系。属于一个VF的客户机物理地址与主机物理地址的对应关系，用于记载向该VF分配的客户机物理地址空间中客户机物理地址与向该VF分配的物理内存空间中主机物理地址的对应关系。当虚拟机管理器以VF的方式向虚拟机提供资源时，若虚拟机对以VF的方式提供的资源进行访问，外部设备接收到的访问请求会携带有VFID。挖补设备接收到访问请求后，能够基于访问请求得到VFID。然后，如图8所示，外部设备能够基于该VFID，从外部设备的配置空间中，查询到该VFID指示的VF的内存请求上下文（memory request context, MR context），然后从该内存请求上下文中，查找到记载有属于该VF的客户机物理地址与主机物理地址的对应关系的页表的基地址，即图8中mr_ctx->mtt_base_addr。然后，外部设备再基于访问请求携带的目标客户机物理地址，对该基地址指示的页表进行查询，得到目标客户机物理地址对应的目标主机物理地址。外部设备基于目标客户机物理地址对页表进行查询的过程是逐级查找的过程。

如图8所示，访问请求携带的iova包括多个地址段，每个地址段通过采用指定位数的二进制数表示的地址字段表示。图8所示的目标客户机物理地址包括四个地址段，分别为：第0至11二进制位表示的第0地址段，第12至20二进制位表示的第1地址段，第21至29二进制位表示的第2地址段，第30至38二进制位表示的第3地址段。外部设备得到页表的基地址后，能够基于该基地址对第2段页表（如图8中的seg 2 table）进行查询。然后，基于第3地址段指示的地址，在第2段页表中查找到第1段页表（如图8中的seg 1 table）的基地址。然后，在第1段页表的基地址指示的第1段页表中，基于第2地址段指示的地址进行查询，得到第0段页表（如图8中的seg 0 table）的基地址。然后，在第0段页表的基地址指示的第0段页表中，基于第1地址段指示的地址进行查询，得到记载有目标客户机物理地址与目标主机物理地址的对应关系的页表条目在第0段页表中的偏移，然后根据第0地址段指示的地址在该页表条目中查询得到目标主机物理地址（即图8中的PA）。需要说明的是，图8中地址字段位数等信息仅为一种示例，其可根据应用需求进行变化。

其中，外部设备基于访问请求得到目标客户机物理地址对应的目标主机物理地址的过程，可选通过设置在外设设备中的微码实现，或者通过设置在该外设设备中的硬件引擎实现，本申请实施例对其不做具体限定。

在一种可能的实现方式中，如图9所示，服务器中还设置有地址管理模块1111。内存中内存块的主机物理地址与客户机物理地址空间中客户机物理地址的对应关系可存储在该地址管理模块中。即图5、图6和图7中映射表存储在地址管理模块中。相应地，地址管理模块可选部署在服务器的内核态中。图9为该地址管理模块设置在硬件111中的示意图。

此时，虚拟机管理器用于向地址管理模块提供主机物理地址与客户机物理地址的对应关系。其中，虚拟机管理器能够基于虚拟机的规格，从内存中为虚拟机分配内存块，然后基于向虚拟机分配的内存块为虚拟机进行设备模拟得到虚拟内存，并配置虚拟内存的客户机物理地址空间，然后建立内存中内存块的主机物理地址与客户机物理地址空间中客户机物理地址的对应关系，并向地址管理模块提供该对应关系。例如，如图5、图6和图7所示，Hypervisor能够向地址管理模块提供MR信息，该MR信息携带有内存

中内存块的主机物理地址与客户机物理地址空间中客户机物理地址的对应关系。并且，该对应关系还携带有该对应关系所属的虚拟机的标识（VMID），以标识该对应关系所属的虚拟机，使得在缺页时能够基于该标识查找对应的虚拟机。其中，虚拟机管理器为虚拟机模拟得到虚拟内存之前，需要先基于虚拟机的规格确定分配给虚拟机使用的内存块，然后基于分配给虚拟机使用的内存块进行设备模拟得到虚拟内存。因此，虚拟机管理器能够得到客户机物理地址空间中客户机物理地址对应的内存块的主机物理地址，并基于其建立主机物理地址与客户机物理地址的对应关系。可选的，服务器的内核中设置有DMMU的驱动。此时，虚拟机管理器可通过DMMU的驱动向地址管理模块发送MR信息。

在本申请中，服务器可兼容原有的地址翻译功能。例如，如图5和图7所示，服务器中还设置有MMU和IOMMU。IOMMU用于接收虚拟机发送的访问请求，并基于访问请求携带的GPA，获取该GPA对应的HPA，从该HPA中读取数据，并向虚拟机提供读取的数据。该IOMMU使用的对应关系也由Hypervisor提供。当服务器兼容原有地址翻译功能时，需要在服务器中做一些改动，使得使用外部设备能力的访问请求能够由外部设备自身处理，不由服务器原有的地址翻译功能处理，即绕过（bypass）服务器原有的地址翻译功能。在一种可能的实现方式中，该功能可通过对外部设备的配置空间进行设置，和/或，对访问请求进行设置实现。例如，可设置外部设备的配置信息指示访问请求直通发送至外部设备，且访问请求指示访问请求携带的地址为通过翻译得到的主机物理地址。这样一来，虚拟设备驱动在虚拟外部设备访问虚拟内存的目标客户机物理地址时，就能够向外部设备发送访问请求，而不向服务器原有的重映射设备发送该访问请求，能够实现对服务器原有的重映射设备的bypass。下面以服务器原有的重映射设备为IOMMU为例，对设置配置信息和访问请求的实现方式进行说明。当重映射设备为其他类型的重映射设备时，设置配置信息和访问请求的实现方式请相应参考该实现方式，此处不再赘述。

在一种可能的实现方式中，通过配置使得服务器原有的重映射设备将访问请求视为已经过翻译的请求，这样服务器原有的重映射设备就不会再对访问请求携带的目标客户机物理地址进行翻译，使得访问请求被当作已经过翻译的请求发送至外部设备，以便于外部设备对其进行处理。已经过翻译的请求是指请求携带的地址为主机物理地址。要实现这个目标，一方面需要外部设备支持已经过翻译的请求，即能够对服务器原有的重映射设备认为的已经过翻译的请求携带的目标客户机物理地址进行翻译。另一方面，要保证访问请求的类型为已经过翻译的请求。

当外部设备的当前状态支持已经过翻译的请求时，此时无需对外部设备进行配置。当外部设备的当前状态不支持已经过翻译的请求，但外部设备具有支持已经够翻译的请求的能力时，需要通过配置使得外部设备的当前状态支持已经过翻译的请求。在一种可能的实现方式中，由于外部设备的配置空间的上下文条目（context entry）中通常记载有外部设备的IOMMU页表信息，该IOMMU页表信息能够指示外部设备对已经过翻译、未经过翻译和无需进行翻译的请求的支持能力。如图10所示，该上下文条目使用128位表示了多个字段，多个字段分别指示不同的信息。其中第3和2位表示的TT字段用于指示外部设备对已经过翻译、未经过翻译和无需进行翻译的请求的支持能力。表1为TT字段为不同的赋值时，外部设备对已经过翻译、未经过翻译和无需进行翻译的请求的支持情况。根据该表1可知，当TT字段的值为“01”时，外部设备的当前状态支持已经过翻译的请求（Translated requests）。因此，若需要对服务器原有的重映射设备进行bypass，可选将TT字段的值改为“01”，使得外部设备的当前状态支持已经过翻译的请求。其中，无需翻译请求（Untranslated request）是指请求是一个没有翻译需求的请求。例如，当访问请求携带的已经是主机物理地址时，访问请求可视为Untranslated request，该访问请求没有翻译需求，因此不存在地址是否为经过翻译的或未经过翻译的情况。已经过翻译的请求（Translated requested requests）是指请求中携带的地址是已经过翻译的地址。翻译请求（Translation Request）指示请求中携带的地址需要进行地址翻译。如果外部设备不具有支持已经够翻译的请求的能力，此时至少有以下两种方案可以规避内核的限制：在第一种实现方式中，直接修改内核，将TT字段设置成“01b”，使得IOMMU认为外部设备支持已经过翻译的请求。在第二种实现方式中，外部设备向服务器上报告外部设备具有支持已经过翻译的请求的能力，即使外部设备并不真正支持已经过翻译的请求。同时由于设备不会发起Translation request，就不会引发非法访问导致的异常。

表1

位（Bits）	字段（Field）	描述（Description）
3:2	翻译 （Translation	00b: Untranslated request会使用由SLPTPTR指向的第二级页表结构进行翻译。Translated requested requests和Translation requests会被阻

	， TT)	挡。 01b: Untranslated、Translated和Translation requests都支持。 10b: Untranslated requests不会翻译而是直通，SLPTPTR字段会被硬件忽略。Translated和Translation requests会被阻挡。 11b: 预留 (reserved)。
--	-------	---

由上可知，为了对服务器原有的重映射设备的bypass，要保证访问请求的类型为已经过翻译的请求。因此，为配合该功能的实现，本申请还需要对访问请求的类型进行设置。对访问请求的类型进行设置，可通过对访问请求的事务层数据包（transaction layer packet，TLP）头中的字段进行设置。PCIe请求的TLP头格式如图11所示，其中AT字段用于指定地址类型，即指示访问请求中的地址是否需要翻译。表2为AT字段为不同的赋值时，访问请求中的地址是否需要翻译的情况。根据该表2可知，本申请可将AT字段设置为10，以指示访问请求携带的地址为不需要进行翻译的地址，实现对重映射设备的bypass。这种情况在IOMMU看来相当于虚拟设备驱动发送的访问请求为已经过翻译的请求，该访问请求携带的GPA为已经过翻译的地址，该访问请求可视为不需要翻译的请求。

表2

AT字段	操作类型
0b00	TLP所带地址没有进行翻译，需要进行翻译。
0b01	TLP为翻译请求（Translation Request）。 TLP所带地址为需要翻译的地址和需要翻译的长度。
0b10	TLP所带地址为已经过翻译的请求，可以直接使用地址访问存储器内容。

外部设备在基于目标客户机物理地址查询第一对应关系时，可能能够查到目标客户机物理地址对应的目标主机物理地址（即命中），也有可能查不到目标客户机物理地址对应的目标主机物理地址（即未命中）。例如，当地址管理模块未记载有目标客户机物理地址与其对应的目标主机物理地址的对应关系时，外部设备在基于目标客户机物理地址查询第一对应关系时，即无法查到目标客户机物理地址对应的目标主机物理地址。此时需要针对这种情况做一些处理，以便于能够基于目标客户机物理地址查询到对应的目标主机物理地址。当对应关系通过页表记载时，外部设备无法在页表中查到目标客户机物理地址对应的目标主机物理地址，可称为出现了缺页异常。此时针对其做的上述处理也称为缺页异常处理。

在一种可能的实现方式中，外部设备还用于在第一对应关系未记载有目标客户机物理地址对应的目标主机物理地址时，向地址管理模块发送指示匹配失败的消息。该消息携带有目标客户机物理地址。地址管理模块用于在接收到该消息后，向虚拟机管理器转发该消息。虚拟机管理器用于基于该消息，在内存中为目标客户机物理地址重新分配内存块，建立重新分配的内存块的主机物理地址与目标客户机物理地址的第二对应关系，并向地址管理模块发送第二对应关系。地址管理模块用于在接收到该第二对应关系后，基于第二对应关系更新第一对应关系。其中，虚拟机管理器得到该第二对应关系的过程中可能需要进行页填充，如在页表中填充未命中的客户机物理地址与其对应的主机物理地址的对应关系。确定未命中的客户机物理地址与其对应的主机物理地址的对应关系的实现过程，请相应参考其得到第一对应关系的实现过程，此处不做赘述。当服务器的内核中设置有扩展内存管理（extend page table，EPT）时，在针对未命中的情况做处理时，还可选向该EPT同步处理结果。

如图5所示，地址管理模块部署在服务器的内核中，且服务器的内核中部署有EPT。外部设备中部署有DMMU，该DMMU基于目标客户机物理地址查询对应关系。当基于目标客户机物理地址的查询未命中时，服务器系统基于该缺页异常执行了缺页异常处理，且向EPT同步了处理结果。在该缺页异常处理过程中，DMMU向地址管理模块发送指示匹配失败的通知消息。该消息携带有目标客户机物理地址。地址管理模块在接收到该消息后，向Hypervisor转发该消息。Hypervisor基于该消息，在内存中为目标客户机物理地址重新分配内存块，建立重新分配的内存块的主机物理地址与目标客户机物理地址的第二对应关系，并向地址管理模块发送第二对应关系。地址管理模块基于第二对应关系更新第一对应关系。其中，服务器的内核中设置有DMMU的驱动DMMU driver，DMMU通过DMMU driver向地址管理模块发送指示匹配失败的消息，Hypervisor通过DMMU driver向地址管理模块发送第二对应关系，地址管理模块通过DMMU driver向Hypervisor转发该消息。图6所示的实现方式与图5所示的实现方式不同的是，在图6所示的实现方

式中，DMMU在未命中时通过DMA使能组件和DMMU driver向地址管理模块发送指示匹配失败的消息，以及向EPT同步处理结果，且EPT会根据缺页的情况执行换入换出过程。图7执行缺页异常处理的过程与图5的过程基本一致，此处对其不再赘述。

需要说明的是，基于第一对应关系更新第二对应关系时，需要考虑使用该第一对应关系的任务的排空。也即是，若存在正在使用该第一对应关系的任务在执行，则不执行基于第一对应关系更新第二对应关系的操作，待正在使用该第一对应关系的任务完成后，再执行基于第一对应关系更新第二对应关系的操作。这样一来，能够保证正在使用该第一对应关系的任务能够被正确执行，以提高更新过程的可靠性。

由上可知，本申请提供的服务器系统能够在基于目标客户机物理地址查询对应关系未命中时执行相关的处理，如执行缺页异常处理，以解决未命中的问题。这样一来，在虚拟机启动之前就可以免pin内存，能够解除全局内存锁定。在云场景中外部设备直通虚拟机的情况下，内存免pin使得能够在虚拟机用户无感知（不对虚拟机系统有特殊要求）的情况下对虚拟机内存进行超分复用，能够提升服务器内存的整体利用率，并保证服务器的性能不下降，该特点在内存压力大的场景中表现的尤其明显。同时，由于内存免pin，使得无需因虚拟机预占内存存在云场景中配置大量的内存，能够降低云场景中内存的使用成本，解决公有云内存成本逐渐增加的问题，进而增强公有云的市场竞争力。

在本申请中，用户态与内核态之间可选通过命令实现交互。内核态与外部设备之间可选通过命令进行交互。如图12所示，用户态与内核态之间通过ioctl命令进行交互，内核态与外部设备之间通过命令队列（command queue，Cmdq）进行交互。例如，用户态与内核态之间通过ioctl命令实现HPA地址的感知。在业务运行过程中若发生了缺页或者改表等事件，内核态与外部设备之间就需要采用cmdq进行交互。其中，ioctl命令主要有映射命令、解除映射命令和查询命令。如图12和图13所示，DMMU的映射命令为DMMU_CMD_MAP（简称为MAP），其用于将一个虚拟机的全部地址空间作为内存区域（memory region）进行映射。当页表按照虚拟机的粒度进行管理时，DMMU_CMD_MAP携带有vmid字段。vmid字段用于指示页表记载的客户机物理地址所属的虚拟机。需要说明的是，页表也可按照其他粒度进行管理，当页表按照其他粒度进行管理时，DMMU_CMD_MAP携带有用于指示页表记载的客户机物理地址所属的粒度主体的字段。DMMU_CMD_MAP中的page_shift字段用于指示页面偏移，表示用于记录客户机物理地址与主机物理地址的对应关系的页面的大小。DMMU_CMD_MAP中的vaddr字段用于指示进行映射的内存区域的虚拟地址。DMMU_CMD_MAP中的iova字段用于指示映射后设备访问的起始地址，在这个场景下也就是GPA的起始地址，通常为0。DMMU_CMD_MAP中的length字段用于按字节表指示内存区域的长度。DMMU_CMD_MAP中的cnt字段用于指示页表中对应关系的总数。DMMU_CMD_MAP中的bdf_array字段用于指示执行地址翻译操作的设备。DMMU的解除映射命令为DMMU_CMD_UNMAP（简称为UNMAP），用于解除虚拟机中的客户机物理地址与主机物理地址的对应关系。DMMU的查询命令为DMMU_CMD_QUERY（简称为QUERY），用于指示对命令中携带的客户机物理地址进行翻译，并将翻译结果返回给用户态程序。DMMU_CMD_QUERY中的paddr字段用于指示映射到的物理地址。DMMU_CMD_MAP中的rsvd字段为保留位。DMMU_CMD_UNMAP和DMMU_CMD_QUERY中与DMMU_CMD_MAP中相同的字段指示的内容，请见DMMU_CMD_MAP中对应字段的说明。图13中位于字段上方的u32、u64和u16表示其下方的字段占用的长度。例如，u16指示其下方的字段占用16个比特（bit）。

在本申请中，可选采用流并发的方式提高内核态与外部设备之间的交互效率。如图14所示，本申请中外部设备呈现的物理功能（physical function，PF）设备在服务器侧能够被DMMU的驱动初始化成cmdq交互的通道。例如，外部设备呈现的PF4设备在服务器侧能够被DMMU的驱动dmmu.ko初始化成cmdq交互的通道，以实现内核态与外部设备之间的交互。类似地，用户态和外部设备之间也可通过外部设备呈现的PF设备实现交互的通道，如图14中的PF0、PF1、PF2、PF3。

本申请中，同一个服务器上可插置一个或多个外部设备，即能够支持多个外部设备共存。需要说明的是，多个外部设备之间需要隔离。图15是服务器上插置有多个外部设备，每个外部设备中配置有DMMU的逻辑示意图。如图15所示，多个DMMU的页表资源均存放在主机侧的DDR中。主机的内核中为每个DMMU配置了DMMU的设备驱动，由DMMU的设备驱动对每个虚拟机进行管理。每个DMMU还配置有DMMU内存映像用于支持访问内存。并且，每个虚拟机作为一个域（domain），连接到其上的多个VF共享同一份页表资源，所以需要在DMMU的设备驱动侧维护一份虚拟机到VF的映射表。外部设备侧存放外部设备自身的基本context信息和页表项缓存，context信息对于每个VF独立，不同外部设备之间相互

隔离。另外，图15为多个外部设备均为网卡，且每张网卡使用一个专用的PF用于配置的示意图。在该场景中，内核驱动会对每张网卡进行探测（probe）操作，然后导出一个dmmu设备文件。在启动虚拟机时，qemu能够根据总线设备功能（bus device function, BDF）信息找到对应的设备文件，然后通过它进行MR下发等操作。其中，BDF信息用于唯一标识VF。

另外，本申请中使用的页表的粒度可基于应用需求进行配置。并且，本申请对页表查找可选用多级页表映射查找和哈希（hash）映射查找。例如，对于4千字节（KB）/8KB粒度的内存页，可采用多级页表查找以节省内存空间，对于较大粒度的大页采用hash映射进行查找，以降低查找时延。

本申请相当于系统层协同虚拟功能输入输出（virtual function input/output, VFIO）设备驱动与外部设备上的内存管理单元（如DMMU）联合解决目前地址翻译存在的问题。Hypervisor对虚拟机内存进行换入/换出时，利用系统API对VFIO驱动进行操作，进而反馈到设备的DMMU中保证外部设备DMA的正确性，同时刷新EPT表保证虚拟机地址的正确性。外部设备利用地址翻译能力将外部设备的内存地址管理下沉到具体的设备中，能够缓减IOMMU/SMMU翻译的压力，提升虚拟地址管理的性能。

由上可知，本申请的改进主要具有以下两方面的价值：

1.随着主机设备VF数量逐渐增多的情况下，本申请的内存管理功能能够使得外部设备的操作性能（如DMA性能）不下降，相较于IOMMU/SMMU的地址翻译功能，本申请使能外部设备对内存进行直接管理，能够缓解IOMMU/SMMU的性能瓶颈，提升设备地址翻译的效率，实现对地址翻译的加速。

2.本申请能够提升公有云内存利用率，降低云内存成本，解决内存超分的关键技术瓶颈，增强公有云市场竞争力。

本申请初步能够应用在公有云的外部设备（如网卡）中，后期可以逐渐推到云核、存储等产品线，形成标准后可以逐渐推送社区，形成IPU/DPU的核心标准。

下面对本申请实施例提供的基于公有云技术的服务器系统的访问方法的实现过程进行说明。图16是本申请实施例提供的基于公有云技术的服务器系统的访问方法的流程图。如图16所示，该基于公有云技术的服务器系统的访问方法包括以下步骤：

步骤1601、云管理平台获取租户输入的虚拟机创建请求，虚拟机创建请求携带有待创建的虚拟机的目标规格，在云管理平台管理的多个服务器中选择可提供目标规格的服务器，在选择的服务器上创建虚拟机。

云管理平台用于管理基础设施，所述基础设施包括多个服务器。租户在需要基于云管理平台管理的基础设施创建虚拟机时，可在租户使用的客户端上执行指定操作，以触发虚拟机创建请求，使得云管理平台在该虚拟机创建请求的指示下为租户创建虚拟机。租户触发虚拟机创建请求后，云管理平台能够获取到虚拟机创建请求，并从该虚拟机创建请求中获取到目标规格。云管理平台获取租户设置的目标规格后，即可在基础设施中选择可提供该目标规格的服务器，并在选择的服务器中创建符合该目标规格的虚拟机。在创建虚拟机的过程中，服务器中的虚拟机管理器需要基于目标规格为虚拟机提供虚拟硬件。示例的，目标规格指示虚拟机需要使用服务器的外部设备的能力和服务器内存的能力，则虚拟机管理器需要基于外部设备进行设备模拟得到虚拟外部设备及其虚拟设备驱动，以使得虚拟机能够使用外部设备的能力，以及，虚拟机管理器需要基于服务器的内存进行设备模拟得到虚拟内存，以使得虚拟机能够使用内存的能力。其中，服务器的包括以下一种或多种：插置于服务器的内存或服务器的共享内存。在虚拟机管理器得到虚拟内存的过程中，需要配置虚拟内存的客户机物理地址空间，以及建立内存中为外部设备配置的内存块的主机物理地址与客户机物理地址空间中客户机物理地址的对应关系。同时，虚拟机管理器该需要向服务器内核态中的地址管理模块发送该对应关系。

步骤1602、虚拟设备驱动在虚拟外部设备访问虚拟内存的目标客户机物理地址时，向外部设备发送访问请求，访问请求携带有目标客户机物理地址。

该步骤1602的实现过程请相应参考前面服务器系统中的相关描述。需要说明的是，当服务器兼容原有的地址翻译功能时，需要在服务器中做一些改动，使得使用外部设备能力的访问请求能够由外部设备自身处理，不由服务器原有的地址翻译功能处理，即绕过（bypass）服务器原有的地址翻译功能。在一种可能的实现方式中，该功能可通过对外部设备的配置空间进行设置，和/或，对访问请求进行设置实现。例如，可设置外部设备的配置信息指示访问请求直通发送至外部设备，且访问请求指示访问请求携带的地址为通过翻译得到的主机物理地址。示例的外部设备的配置信息指示外部设备支持已经过翻译的请求，

且访问请求指示访问请求携带的地址为主机物理地址。其实现原理请相应参考前面服务器系统中的相关描述，此处不再赘述。

步骤1603、外部设备基于访问请求，查询客户机物理地址与主机物理地址的第一对应关系。

该步骤1603的实现过程请相应参考前面服务器系统中的相关描述，此处不再赘述。

步骤1604、外部设备基于访问请求查询到目标客户机物理地址对应的目标主机物理地址后，获取目标主机物理地址记载的目标数据，并向虚拟设备驱动发送目标数据。

该步骤1604的实现过程请相应参考前面服务器系统中的相关描述，此处不再赘述。

外部设备在基于目标客户机物理地址查询第一对应关系时，可能能够查到目标客户机物理地址对应的目标主机物理地址（即命中），也有可能查不到目标客户机物理地址对应的目标主机物理地址（即未命中）。例如，当地址管理模块未记载有目标客户机物理地址与其对应的目标主机物理地址的对应关系时，外部设备在基于目标客户机物理地址查询第一对应关系时，即无法查到目标客户机物理地址对应的目标主机物理地址。此时需要针对这种情况做一些处理，以便于能够基于目标客户机物理地址查询到对应的目标主机物理地址。因此，如图17所示，该访问方法可选还包括：

步骤1605、外部设备在第一对应关系未记载有目标客户机物理地址对应的目标主机物理地址时，向地址管理模块发送指示匹配失败的消息，消息携带有目标客户机物理地址。

该步骤1605的实现过程请相应参考前面服务器系统中的相关描述，此处不再赘述。

步骤1606、地址管理模块向虚拟机管理器转发消息。

该步骤1606的实现过程请相应参考前面服务器系统中的相关描述，此处不再赘述。

步骤1607、虚拟机管理器基于消息，在内存中为目标客户机物理地址分配内存块，建立内存块的主机物理地址与目标客户机物理地址的第二对应关系，并向地址管理模块发送第二对应关系。

该步骤1607的实现过程请相应参考前面服务器系统中的相关描述，此处不再赘述。

步骤1608、地址管理模块基于第二对应关系更新第一对应关系。

该步骤1608的实现过程请相应参考前面服务器系统中的相关描述，此处不再赘述。

由上可知，该服务器系统相当于将地址翻译功能下沉至外部设备中，使得外部设备对为虚拟机分配的内存进行直接管理，分担对访问请求中目标客户机物理地址进行地址翻译的任务。这样一来，就不需要服务器中的IOMMU/SMMU等地址翻译设备执行该地址翻译的任务，不需要服务器中的IOMMU/SMMU等地址翻译设备的TLB记录该外部设备的虚拟外部设备访问的目标客户机物理地址的对应关系，减少了该TLB中需要记录的对应关系的数量，缓解了因该TLB的页表条目较少和外部设备数量过多的矛盾，降低了地址翻译过程中出现TLB缺页问题的概率，有效地提高了服务器系统的地址翻译性能，有助于提高外部设备的操作性能（如DMA性能）和对内存进行管理的效率。

需要说明的是，本申请实施例提供的基于公有云技术的服务器系统的访问方法的步骤先后顺序可以进行适当调整，步骤也可以根据情况进行相应增减。任何熟悉本技术领域的技术人员在本申请揭露的技术范围内，可轻易想到变化的方法，都应涵盖在本申请的保护范围之内，因此不再赘述。

下面对本申请实施例涉及的基本硬件结构举例说明。

本申请还提供一种计算设备。如图18所示，计算设备1800包括：总线1802、处理器1804、存储器1806和通信接口1808。处理器1804、存储器1806和通信接口1808之间通过总线1802通信。计算设备1800可以是服务器或终端设备。应理解，本申请不限定计算设备1800中的处理器、存储器的个数。

总线1802可以是外设部件互连标准（peripheral component interconnect, PCI）总线或扩展工业标准结构（extended industry standard architecture, EISA）总线等。总线可以分为地址总线、数据总线、控制总线等。为便于表示，图18中仅用一条线表示，但并不表示仅有一根总线或一种类型的总线。总线1802可包括在计算设备1800各个部件（例如，存储器1806、处理器1804、通信接口1808）之间传送信息的通路。

处理器1804可以包括中央处理器（central processing unit, CPU）、图形处理器（graphics processing unit, GPU）、微处理器（micro processor, MP）或者数字信号处理器（digital signal processor, DSP）等处理器中的任意一种或多种。

存储器1806用于存储计算机程序，计算机程序包括操作系统和可执行代码（即程序指令）。存储器1806可以包括易失性存储器（volatile memory），例如随机存取存储器（random access memory, RAM）。处理器1804还可以包括非易失性存储器（non-volatile memory），例如只读存储器（read-only memory, ROM），快

闪存存储器, 机械硬盘 (hard disk drive, HDD) 或固态硬盘 (solid state drive, SSD)。

存储器1806中存储有可执行的程序代码, 处理器1804执行该可执行的程序代码能够实现本申请实施例提供的基于公有云技术的服务器系统的访问方法。也即, 存储器1806上存有用于执行基于公有云技术的服务器系统的访问方法的指令。

通信接口1808使用例如但不限于网络接口卡、收发器一类的收发模块, 来实现计算设备1800与其他设备或通信网络之间的通信。

本申请实施例还提供了一种计算设备集群。该计算设备集群包括至少一台计算设备。该计算设备可以是服务器, 例如是中心服务器、边缘服务器, 或者是本地数据中心中的本地服务器。在一些实施例中, 计算设备也可以是台式机、笔记本电脑或者智能手机等终端设备。

如图19所示, 计算设备集群包括至少一个计算设备1800。计算设备集群中的一个或多个计算设备1800中的存储器1806中可以存有相同的用于执行基于公有云技术的服务器系统的访问方法的指令。

在一些可能的实现方式中, 该计算设备集群中的一个或多个计算设备1800的存储器1806中也可以分别存有用于执行基于公有云技术的服务器系统的访问方法的部分指令。换言之, 一个或多个计算设备1800的组合可以共同执行用于执行基于公有云技术的服务器系统的访问方法的指令。

需要说明的是, 计算设备集群中的不同的计算设备1800中的存储器1806可以存储不同的指令, 分别用于执行基于公有云技术的服务器系统的访问方法的部分功能。也即, 不同的计算设备1800中的存储器1806存储的指令可以实现基于公有云技术的服务器系统的访问方法中的部分功能。

在一些可能的实现方式中, 计算设备集群中的一个或多个计算设备可以通过网络连接。其中, 网络可以是广域网或局域网等等。图20示出了一种可能的实现方式。如图20所示, 两个计算设备1800A和1800B之间通过网络进行连接。具体地, 通过各个计算设备中的通信接口与网络进行连接。

应理解, 图20中示出的计算设备1800A的功能也可以由多个计算设备1800完成。同样, 计算设备1800B的功能也可以由多个计算设备1800完成。

本申请实施例还提供了另一种计算设备集群。该计算设备集群中各计算设备之间的连接关系可以类似的参考图19和图20计算设备集群的连接方式。不同的是, 该计算设备集群中的一个或多个计算设备1800中的存储器1806中可以存有相同的用于执行基于公有云技术的服务器系统的访问方法的指令。

在一些可能的实现方式中, 该计算设备集群中的一个或多个计算设备1800的存储器1806中也可以分别存有用于执行基于公有云技术的内存管理方法的部分指令。换言之, 一个或多个计算设备1800的组合可以共同执行用于执行基于公有云技术的服务器系统的访问方法的指令。

本申请实施例还提供了一种服务器。该服务器用于实现本申请提供的服务器系统中服务器的功能。该服务器的工作原理及实现方式请相应参考前面服务器系统中的相关描述。

本申请实施例还提供了一种由本申请提供的服务器执行的基于公有云技术的服务器的访问方法。服务器通过执行该方法, 能够实现本申请提供的基于公有云技术的服务器系统的访问方法中服务器的功能。其工作原理及实现方式请相应参考前面服务器系统中的相关描述。

本申请实施例还提供了一种外部设备。该外部设备用于实现本申请提供的服务器系统中外部设备的功能。该外部设备的工作原理及实现方式请相应参考前面服务器系统中的相关描述。

本申请实施例还提供了一种由本申请提供的外部设备执行的基于公有云技术的服务器的访问方法。外部设备通过执行该方法, 能够实现本申请提供的基于公有云技术的服务器系统的访问方法中外部设备的功能。其工作原理及实现方式请相应参考前面服务器系统中的相关描述。

本申请实施例还提供了一种包含指令的计算机程序产品。计算机程序产品可以是包含指令的, 能够运行在计算设备上或被储存在任何可用介质中的软件或程序产品。当计算机程序产品在至少一个计算设备上运行时, 使得至少一个计算设备执行基于公有云技术的服务器系统的访问方法。

本申请实施例还提供了一种计算机可读存储介质。计算机可读存储介质可以是计算设备能够存储的任何可用介质或者是包含一个或多个可用介质的数据中心等数据存储设备。可用介质可以是磁性介质，（例如，软盘、硬盘、磁带）、光介质（例如，DVD）、或者半导体介质（例如固态硬盘）等。该计算机可读存储介质包括指令，指令指示计算设备执行基于公有云技术的服务器系统的访问方法，或指示计算设备执行基于公有云技术的服务器系统的访问方法。

本领域普通技术人员可以理解实现上述实施例的全部或部分步骤可以通过硬件来完成，也可以通过程序来指令相关的硬件完成，该程序可以存储于一种计算机可读存储介质中，上述提到的存储介质可以是只读存储器，磁盘或光盘等。

需要说明的是，本申请所涉及的信息（包括但不限于用户设备信息、用户个人信息等）、数据（包括但不限于用于分析的数据、存储的数据、展示的数据等）以及信号，均为经用户授权或者经过各方充分授权的，且相关数据的收集、使用和处理需要遵守相关国家和地区的相关法律法规和标准。例如，本申请中涉及到的原始数据和可执行代码等都是在充分授权的情况下获取的。

在本申请实施例中，术语“第一”、“第二”和“第三”仅用于描述目的，而不能理解为指示或暗示相对重要性。术语“至少一个”是指一个或多个，术语“多个”指两个或两个以上，除非另有明确的限定。

本申请中术语“和/或”，仅仅是一种描述关联对象的关联关系，表示可以存在三种关系，例如，A和/或B，可以表示：单独存在A，同时存在A和B，单独存在B这三种情况。另外，本文中字符“/”，一般表示前后关联对象是一种“或”的关系。

最后应说明的是：以上实施例仅用以说明本申请的技术方案，而非对其限制；尽管参照前述实施例对本申请进行了详细的说明，本领域的普通技术人员应当理解：其依然可以对前述各实施例所记载的技术方案进行修改，或者对其中部分技术特征进行等同替换；而这些修改或者替换，并不使相应技术方案的本质脱离本申请各实施例技术方案的保护范围。

权 利 要 求 书

1. 一种基于公有云技术的服务器系统，其特征在于，所述服务器系统包括服务器和插置于所述服务器的外部设备，所述服务器和所述外部设备建立有连接通道；

所述服务器中设置有虚拟机和虚拟机管理器，所述虚拟机管理器用于为所述虚拟机提供虚拟硬件，所述虚拟硬件包括虚拟外部设备和虚拟内存，所述虚拟外部设备基于所述外部设备进行设备模拟得到，所述虚拟内存基于为所述外部设备配置的内存进行设备模拟得到，所述虚拟机中设置有所述虚拟外部设备的虚拟设备驱动，所述虚拟设备驱动用于在所述虚拟外部设备访问所述虚拟内存的目标客户机物理地址时，向所述外部设备发送访问请求，所述访问请求携带有所述目标客户机物理地址；

所述外部设备用于基于所述访问请求，查询客户机物理地址与主机物理地址的第一对应关系，得到所述目标客户机物理地址对应的目标主机物理地址，获取所述目标主机物理地址记载的目标数据，并向所述虚拟设备驱动发送所述目标数据。

2. 如权利要求1所述的服务器系统，其特征在于，所述服务器中还设置有地址管理模块；

所述外部设备还用于在所述第一对应关系未记载有所述目标客户机物理地址对应的目标主机物理地址时，向所述地址管理模块发送指示匹配失败的消息，所述消息携带有所述目标客户机物理地址；

所述地址管理模块用于向所述虚拟机管理器转发所述消息；

所述虚拟机管理器用于基于所述消息，在所述内存中为所述目标客户机物理地址分配内存块，建立所述内存块的主机物理地址与所述目标客户机物理地址的第二对应关系，并向所述地址管理模块发送所述第二对应关系；

所述地址管理模块用于基于所述第二对应关系更新所述第一对应关系。

3. 如权利要求1或2所述的服务器系统，其特征在于，所述服务器中还设置有地址管理模块；

所述虚拟机管理器用于基于所述虚拟机的规格，为所述虚拟机进行设备模拟得到所述虚拟内存，并配置所述虚拟内存的客户机物理地址空间，建立所述内存中为所述外部设备配置的内存块的主机物理地址与所述客户机物理地址空间中客户机物理地址的对应关系，并向所述地址管理模块发送所述对应关系。

4. 如权利要求1至3任一所述的服务器系统，其特征在于，所述外部设备的配置信息指示所述外部设备支持已经过翻译的请求，且所述访问请求指示所述访问请求携带的地址为主机物理地址。

5. 如权利要求1至4任一所述的服务器系统，其特征在于，为所述外部设备配置的内存包括以下一种或多种：插置于所述服务器的内存或所述服务器的共享内存。

6. 一种基于公有云技术的服务器系统的访问方法，其特征在于，所述服务器系统包括服务器和插置于所述服务器的外部设备，所述服务器和所述外部设备建立有连接通道，所述服务器中设置有虚拟机和虚拟机管理器，所述虚拟机管理器用于为所述虚拟机提供虚拟硬件，所述虚拟硬件包括虚拟外部设备和虚拟内存，所述虚拟外部设备基于所述外部设备进行设备模拟得到，所述虚拟内存基于为所述外部设备配置的内存进行设备模拟得到，所述虚拟机中设置有所述虚拟外部设备的虚拟设备驱动，所述方法包括：

所述虚拟设备驱动在所述虚拟外部设备访问所述虚拟内存的目标客户机物理地址时，向所述外部设备发送访问请求，所述访问请求携带有所述目标客户机物理地址；

所述外部设备基于所述访问请求，查询客户机物理地址与主机物理地址的第一对应关系，得到所述目标客户机物理地址对应的目标主机物理地址，获取所述目标主机物理地址记载的目标数据，并向所述虚拟设备驱动发送所述目标数据。

7. 如权利要求6所述的方法，其特征在于，所述服务器中还设置有地址管理模块，所述方法还包括：

所述外部设备在所述第一对应关系未记载有所述目标客户机物理地址对应的目标主机物理地址时，向所述地址管理模块发送指示匹配失败的消息，所述消息携带有所述目标客户机物理地址；

所述地址管理模块向所述虚拟机管理器转发所述消息；

所述虚拟机管理器基于所述消息，在所述内存中为所述目标客户机物理地址分配内存块，建立所述内存块的主机物理地址与所述目标客户机物理地址的第二对应关系，并向所述地址管理模块发送所述第二对应关系；

所述地址管理模块基于所述第二对应关系更新所述第一对应关系。

8. 如权利要求6或7所述的方法，其特征在于，所述服务器中还设置有地址管理模块，所述方法还包括：

所述虚拟机管理器基于所述虚拟机的规格，为所述虚拟机进行设备模拟得到所述虚拟内存，并配置所述虚拟内存的客户机物理地址空间，建立所述内存中为所述外部设备配置的内存块的主机物理地址与所述客户机物理地址空间中客户机物理地址的对应关系，并向所述地址管理模块发送所述对应关系。

9. 如权利要求6至8任一所述的方法，其特征在于，所述外部设备的配置信息指示所述外部设备支持已经过翻译的请求，且所述访问请求指示所述访问请求携带的地址为主机物理地址。

10. 如权利要求6至9任一所述的方法，其特征在于，为所述外部设备配置的内存包括以下一种或多种：插置于所述服务器的内存或所述服务器的共享内存。

11. 一种计算设备，其特征在于，包括处理器和存储器，所述存储器中存储有程序指令，所述处理器运行程序指令，使得所述计算设备执行权利要求6至10任一所述的方法。

12. 一种计算机可读存储介质，其特征在于，包括程序指令，当所述程序指令在计算设备上运行时，使得所述计算设备执行如权利要求6至10任一所述的方法。

13. 一种包含指令的计算机程序产品，其特征在于，当所述指令被计算设备运行时，使得所述计算设备执行如权利要求6至10任一项所述的方法。