内存页缺页处理方法以及系统

交叉援引

本公开要求于2023年07月19日提交中国专利局、申请号202310890608.5、申请名称“内存页缺页处理方法以及系统”的中国专利申请的优先权，其全部内容通过引用结合在本公开中。

技术领域

本公开实施例涉及计算机技术领域，特别涉及一种内存页缺页处理方法。

背景技术

在启用直通设备（PASSTHROUGH）的虚拟机场景下，需要到 IOMMU(Input/Output Memory Management Unit，输入/输出内存管理单元)中进行地址翻译，IOMMU 可连接直接内存访问(Direct Memory Access，DMA)I/O（Input/Output，输入/输出）总线和宿主机的内存，把对设备访问的虚拟地址转化成物理地址。

在实际应用中，虚拟机管理器(Virtual Machine Manager，VMM)会回收物理内存，将存储于在物理内存中的数据存储于硬盘中，并释放的物理内存页供其它进程使用。在 I/O 访问的进程再次需要内存时，会出现I/O 内存页缺页(Page Fault)。

市面上主流正在提供服务的 CPU（Central Processing Unit，中央处理器），其 IOMMU 不支持 I/O 缺页功能，在出现 I/O 内存页缺页的情况下，IOMMU 不能直接将缺页请求发送至 VMM，而是返回缺页异常消息至直通设备，如 PCI 设备；直通设备再根据该缺页异常消息依赖 IOMMU 发送缺页请求至 VMM，以恢复缺失的内存页，增加了 I/O 内存页缺页恢复的处理链路，对 I/O 内存页缺页恢复的时间长，导致设备访问内存延迟较大。

发明内容

有鉴于此，本公开实施例提供了一种内存页缺页处理方法。本公开一个或者多个实施例同时涉及一种内存页缺页处理系统，一种缺页请求处理设备，一种计算设备，一种计算机可读存储介质以及一种计算机程序，以解决现有技术中存在的技术缺陷。

根据本公开实施例的第一方面，提供了一种内存页缺页处理方法，包括：

所述数据处理设备，在确定所述虚拟机发送的 I/O 访问对应的内存页缺失的情况下，向所述缺页请求处理设备发送所述缺页请求；

所述缺页请求处理设备，根据所述缺页请求发送缺页消息至缺页异常队列；

所述虚拟机管理器，获取所述缺页异常队列中的所述缺页消息，并根据所述缺页消息进行缺页处理，获得缺失的内存页，并向所述缺页请求处理设备发送针对所述缺失的内存页的处理完成指令；

所述缺页请求处理设备，根据所述处理完成指令更新状态表，以使所述数据处理设备根据更新后的状态表进行所述 I/O 访问处理，其中，所述状态表中包括所述缺失的内存页的存

在状态。

根据本公开实施例的第二方面，提供了一种内存缺页处理系统，包括：

所述数据处理设备，被配置为在确定所述虚拟机发送的 I/O 访问对应的内存页缺失的情况下，向所述缺页请求处理设备发送所述缺页请求；

所述缺页请求处理设备，被配置为根据所述缺页请求发送缺页消息至缺页异常队列；

所述虚拟机管理器，被配置为获取所述缺页异常队列中的所述缺页消息，并根据所述缺页消息进行缺页处理，获得缺失的内存页，并向所述缺页请求处理设备发送针对所述缺失的内存页的处理完成指令；

所述缺页请求处理设备，被配置为根据所述处理完成指令更新状态表，以使所述数据处理设备根据更新后的状态表进行所述 I/O 访问处理，其中，所述状态表中包括缺失的内存页的存在状态。

根据本公开实施例的第三方面，提供了一种缺页请求处理设备，包括缺页消息发送模块以及状态表更新模块，其中：

所述缺页消息发送模块，被配置为根据缺页请求发送缺页消息至缺页异常队列，其中，所述缺页消息为数据处理设备在确定虚拟机发送的 I/O 访问对应的内存页缺失的情况下发送的；

所述状态表更新模块，被配置为根据处理完成指令更新状态表，以使所述数据处理设备根据更新后的状态表进行所述 I/O 访问处理，其中，所述处理完成指令为虚拟机管理器，在获取所述缺页异常队列中的缺页消息，并根据所述缺页消息进行缺页处理，获得缺失的内存页的情况下发送的，所述状态表中包括缺失的内存页的存在状态。

根据本公开实施例的第四方面，提供了一种计算设备，包括：

存储器和处理器；

所述存储器设置为存储计算机可执行指令，所述处理器设置为执行所述计算机可执行指令，该计算机可执行指令被处理器执行时实现上述内存缺页处理方法的步骤。

根据本公开实施例的第五方面，提供了一种计算机可读存储介质，其存储有计算机可执行指令，该指令被处理器执行时实现上述内存缺页处理方法的步骤。

根据本公开实施例的第六方面，提供了一种计算机程序，其中，当所述计算机程序在计算机中执行时，令计算机执行上述内存缺页处理方法的步骤。

本公开一个实施例提供的一种内存页缺页处理方法，应用于内存缺页处理系统，所述内存缺页处理系统包括主机、运行在主机中的虚拟机和虚拟机管理器、部署在所述主机中对所述虚拟机的 I/O 访问进行处理的数据处理设备、部署在所述主机中对所述数据处理设备发送的缺页请求进行处理的缺页请求处理设备，其中，所述方法包括：所述数据处理设备，在确定所述虚拟机发送的 I/O 访问对应的内存页缺失的情况下，向所述缺页请求处理设备发送所述缺页请求；所述缺页请求处理设备，根据所述缺页请求发送缺页消息至缺页异常队列；所

述虚拟机管理器，获取所述缺页异常队列中的所述缺页消息，并根据所述缺页消息进行缺页处理，获得缺失的内存页，并向所述缺页请求处理设备发送针对所述缺失的内存页的处理完成指令；所述缺页请求处理设备，根据所述处理完成指令更新状态表，以使所述数据处理设备根据更新后的状态表进行所述 I/O 访问处理，其中，所述状态表中包括所述缺失的内存页的存在状态。

具体的，该内存页缺页处理方法，通过在数据处理设备中引入缺页请求处理设备，在数据处理设备向缺页请求处理设备发送缺页请求的情况下，缺页请求处理设备根据缺页请求发送缺页消息至虚拟机管理器，实现不依赖于 IOMMU 的 I/O 缺页请求接口，缩短 I/O 内存页缺页恢复的处理链路，提高 I/O 内存页缺页恢复的处理效率，降低设备访问内存延迟。

附图说明

图 1 是本公开一个实施例提供的一种内存页缺页处理方法的场景示意图；

图 2 是本公开一个实施例提供的一种内存页缺页处理方法的流程图；

图 3 是本公开一个实施例提供的一种应用内存页缺页处理方法的DPU辅助虚拟化架构下的 VPRI 架构示意图；

图 4 是本公开一个实施例提供的一种内存页缺页处理方法的处理过程流程图；

图 5 是本公开一个实施例提供的一种缺页请求处理设备的结构示意图；

图 6 是本公开一个实施例提供的一种内存页缺页处理系统的结构示意图；

图 7 是本公开一个实施例提供的一种计算设备的结构框图。

具体实施方式

在下面的描述中阐述了很多具体细节以便于充分理解本公开。但是本公开能够以很多不同于在此描述的其它方式来实施，本领域技术人员可以在不违背本公开内涵的情况下做类似推广，因此本公开不受下面公开的具体实施的限制。

在本公开一个或多个实施例中使用的术语是出于描述特定实施例的目的，而非旨在限制本公开一个或多个实施例。在本公开一个或多个实施例和所附权利要求书中所使用的单数形式的“一种”、“所述”和“该”也旨在包括多数形式，除非上下文清楚地表示其他含义。还应当理解，本公开一个或多个实施例中使用的术语“和/或”是指并包含一个或多个相关联的列出项目的任何或所有可能组合。

应当理解，尽管在本公开一个或多个实施例中可能采用术语第一、第二等来描述各种信息，但这些信息不应限于这些术语。这些术语用来将同一类型的信息彼此区分开。例如，在不脱离本公开一个或多个实施例范围的情况下，第一也可以被称为第二，类似地，第二也可以被称为第一。取决于语境，如在此所使用的词语“如果”可以被解释成为“在……时”或“当……时”或“响应于确定”。

此外，需要说明的是，本公开一个或多个实施例所涉及的用户信息（包括但不限于用户设备信息、用户个人信息等）和数据（包括但不限于设置为分析的数据、存储的数据、展示

的数据等），均为经用户授权或者经过各方充分授权的信息和数据，并且相关数据的收集、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，并提供有相应的操作入口，供用户选择授权或者拒绝。

首先，对本公开一个或多个实施例涉及的名词术语进行解释。

DMA：Direct Memory Access，直接内存访问是一些计算机总线架构提供的功能，它能使数据从附加设备（如磁盘驱动器）直接发送到计算机主板的内存上。

ATS：ATS 全称是 Address Translation Service，顾名思义，就是一个地址翻译服务机制。PCIe 下的 ATS 是以 CPU 为中心，PCIe 总线上的各个设备可以通过 ATS 机制向主机申请未翻译地址对应的物理地址映射以及响应的属性、权限等信息。

IOPF：I/O Page Fault，PCI 设备直接发起的 I/O 缺页请求。

PCI：Peripheral Component Interconnect，一种设置为定义局部总线的标准。

PCI Function：PCI 功能，被设计为每个设备中之内的一个逻辑层次。

VMM ：全称是 Virtual Machine Monitor，虚拟机监控系统，也叫 Hypervisor，是虚拟化层的具体实现，集中管理物理资源，每个虚拟处理器系统需要通过 VMM 访问物理资源。

MMIO：Memory-mapped I/O，即内存映射 I/O，它是 PCI 规范的一部分。

MSI-X 中断：扩充讯息信号中断(MSI-X)是 MSI 中断的增强型版本，MSI(Message Signaled Interrupt) 是 PCI 设备通过写一个特定消息到特定地址，从而触发一个 DPU 中断。MSI-X 中断机制提出目的是扩展 PCIe 设备使用的中断向量个数，同时解决 MSI 中断要求使用中断向量必须保持连续的问题。

PRI：Page Request Interface，是 PCI-SIG 定义的一个标准 PCI 设备能力(PCI Capability)，其存在结合 ATS，可使设备使用的 I/O 内存无需在内存中固定住，缓解了虚拟机占用过多固定内存，降低主机整体可分配内存的瓶颈。

针对 I/O 内存页缺页恢复的时间长，导致设备访问内存延迟较大的技术问题，在本公开一些实施例中，设备侧增设缺页请求服务的旁路 PCI Function，为不具备 PRI 功能的 IOMMU 的主机提供 IOPF 功能，实现大块数据对应的大块 I/O 内存页缺页的恢复，设备可访问已恢复的内存页。相较于现有内存页缺页处理方式，无需依赖 IOMMU 硬件，缩短内存页缺页恢复的处理链路，有助于降低 I/O 内存页缺页恢复时长，进而降低设备访问内存延迟。

在本公开中，提供了一种内存页缺页处理方法，本公开同时涉及一种内存页缺页处理系统，一种计算设备，以及一种计算机可读存储介质，在下面的实施例中逐一进行详细说明。

参见图 1，图 1 示出本公开一个实施例提供的一种内存页缺页处理方法的场景示意图，包括：宿主机 102、虚拟机 104、直通设备 106 和 VPRI-PCI 设备 108。

其中，宿主机 102 可以理解为指具有计算、存储及通信等功能的计算机设备，例如宿主机 102 可以为服务端设备。例如，宿主机 102 可以为单一服务器设备，也可以为云化的服务器阵列等。当然，宿主机 102 也可以为手机、平板电脑、个人电脑、穿戴设备等终端设备。

在本申请实施例中，宿主机 102 部署有虚拟机(Virtual Machine，VM)104。其中，虚拟机 104 也可称为宿主机 102 的客户端(Guest)。虚拟机 104 有独立的 CPU、内存、网络以及磁盘等。

直通设备 106 可以理解为指宿主机 102 挂载的 I/O 硬件设备，可包括：网卡、存储介质等。其中，存储介质可为磁盘、硬盘等可持久存储介质。直通设备 106 可通过通信接口与宿主机 102 连接，根据通信接口的形式不同，直通设备 106 可实现为 PCI 设备、PCIe 设备等。

VMM 可以理解为对虚拟机 104 进行管理的逻辑节点，可部署于宿主机 102，也可部署于与宿主机 102 通信的其它物理机。在 VMM 部署于宿主机 102 的情况下，VMM 可运行于宿主机 102 的处理单元。VMM 可对虚拟机 104 进行调度任务、负载均衡、状态监测等。宿主机 102 的处理单元可为 CPU、DPU 等。

在直通(Pass through)技术中，虚拟机 104 可通过 VMM，直接访问宿主机 102 挂载的直通设备 106，这样虚拟机 104 的 I/O 操作路径几乎和无虚拟化环境下的 I/O 路径相同，对于直通设备 106，可通过直接内存访问(DMA)方式访问宿主机 102 的内存。对于 DMA 方式，直通设备 106 和宿主机 102 内存之间有一条数据通路，实现直通设备 106 和宿主机 102 内存直接的数据传输。在直通设备 106 和宿主机 102 内存之间数据传输过程中，不需要宿主机 102 的 CPU 参与。

在虚拟化技术中，虚拟机 104 上运行的操作系统(guest OS)通常不知道它所访问的宿主机的物理内存地址，也可称为宿主机物理地址(Host physical address，HPA)，简称为物理地址。如果要进行 DMA 操作，就有可能破坏内存。因为直通设备 106 可获知虚拟机的物理内存地址，简称为虚拟地址(Virtual address，VA)，也可称为客户端物理地址 (Guest physical address，GPA)，但直通设备 106 并不知道 GPA 和 HPA 之间的映射关系。因此，在本申请实施例中，在直通技术中引入 IOMMU。

IOMMU 可通信连接于直通设备 106 和宿主机 102 内存之间。可选地，IOMMU 可通过通信接口总线，与直通设备 106 和宿主机 102 内存通信连接。串行接口总线可为 PCI 接口总线、PCIe 接口总线等。IOMMU 可保证直通设备 106 进行 DMA 操作时能够访问宿主机 102 内存。

对于没有 IOMMU 的宿主机 102，直通设备 106 通过 DMA 方式可以访问宿主机 102 的全部内存地址空间。对于设置有 IOMMU 的宿主机，虚拟机 104 的 CPU 可分配直通设备 106 可访问的 GPA，并由 VMM 为 GPA 分配对应的 HPA。IOMMU 可维护 GPA 与 HPA 之间的映射表，该映射表也可称为页表。页表记录有 GPA 与 HPA 之间的对应关系。在直通设备 106 通过 DMA 方式访问宿主机 102 的内存时，VMM 可捕获直通设备 106 发出的 DMA 请求，并将 DMA 请求透传给 IOMMU。IOMMU 可从 DMA 请求中获取待访问的 GPA；之后，将待访问的 GPA 在 IOMMU 存储的页表中进行匹配，以得到待访问的 GPA 对应的 HPA。对于直通设备 106 来说，可通过待访问的 GPA 对应的 HPA 访问宿主机 102 内存对应的内存空间。

在实际应用中，VMM 会回收物理内存，将存储于在物理内存中的数据存储于硬盘中，并释放的物理内存页供其它进程使用。在直通设备 106 的进程再次需要内存时，会出现 I/O

内存缺页(Page Fault)。

以直通设备 106 为 DPU 为例，对本公开实施例提供的内存页缺页处理方法，进行详细说明。

在本公开实施例中，在 DPU 中引入 VPRI-PCI 设备 108，其中，VPRI-PCI 设备 108 为一个独立于 DPU106 的一个旁路 PCI Function，VPRI-PCI 设备可以嵌入在直通设备 106 中，也可以通过接口连接在直通设备 106 上，本公开对此不做限定；VPRI-PCI 设备 108 可存放一个与 GPA 对应的内存页（4K/2M/1G）的存在状态（present state），一个页用一个 bit（位）来记录其存在状态，如存在状态为“1”，则表示 GPA 对应的内存页存在；如存在状态为“0”，则表示 GPA 对应的内存页缺失。

在虚拟机 104 对 DPU 发送 I/O 访问后，DPU 通过查找 VPRI-PCI 设备 108 中的状态表，发现存在 I/O 内存页缺页的情况下，DPU 向 VPRI-PCI 设备 108 发送缺页请求，VPRI-PCI 设备 108 可根据该缺页请求发送缺页消息至 VMM，以使 VMM 根据该缺页消息进行缺页处理，获得缺失的内存页，并向 VPRI-PCI 设备 108 发送针对缺失的内存页的处理完成指令，其中，处理完成指令为数据传达类的指令，设置为传达通过 VMM 的缺页处理获得了缺失的内存页的消息，以更新 VPRI-PCI 设备 108 中的状态表，在缺页处理之前状态表中缺失的内存页的存在状态为 0，而处理完成后的状态表中缺失的内存页的存在状态为 1，表示获得了缺失的内存页，DPU 即可根据更新后的状态表进行 I/O 访问处理。

基于此，本公开实施例提供的内存页缺页处理方法通过在数据处理设备中引入缺页请求处理设备，在数据处理设备向缺页请求处理设备发送缺页请求的情况下，缺页请求处理设备根据缺页请求发送缺页消息至虚拟机管理器，实现不依赖于 IOMMU 的 I/O 缺页请求接口，也就是说，在需要进行缺页处理的情况下，不需要 IOMMU 向数据处理设备发送缺页消息，再由数据处理设备依赖 IOMMU 发送缺页请求，进行缺页处理；可直接由数据处理设备中的缺页请求处理设备发送缺页请求，缩短 I/O 内存页缺页恢复的处理链路，提高 I/O 内存页缺页恢复的处理效率，降低设备访问内存延迟。

参见图 2，图 2 示出本公开一个实施例提供的一种内存页缺页处理方法的流程图，该内存页缺页处理方法，应用于内存缺页处理系统，所述内存缺页处理系统包括主机、运行在主机中的虚拟机和虚拟机管理器、部署在所述主机中对所述虚拟机的 I/O 访问进行处理的数据处理设备、部署在所述主机中对所述数据处理设备发送的缺页请求进行处理的缺页请求处理设备，具体包括以下步骤。

步骤 202：所述数据处理设备，在确定所述虚拟机发送的 I/O 访问对应的内存页缺失的情况下，向所述缺页请求处理设备发送所述缺页请求。

其中，数据处理设备可以理解为，图 1 中的直通设备 106，即 PCI I/O 硬件设备,如标准网卡 NIC（Network Interface Card）、数据处理器 DPU（Data Processing Unit）等；虚拟机可以理解为，运行在宿主机中的，可以像真实机器一样运行程序的宿主机的软件实现；I/O（Input/Output，即输入/输出，通常指数据在内部存储器和外部存储器或其他周边设备之间的

输入和输出）访问可以理解为，输入/输出访问，例如，一虚拟机在对某一磁盘进行 I/O 访问的情况下，I/O 访问可以理解为，虚拟机对磁盘的读写访问。

内存页可以理解为，把物理内存空间切成一段段固定尺寸的大小的空间，存有物理地址；缺页请求处理设备可以理解为，图 1 中的 VPRI-PCI 设备 108；实际应用中，该缺页请求处理设备可以理解为，发送缺页请求的设备，其中，缺页请求可以理解为，针对上述 I/O 访问中内存页缺页的缺页请求。

在本公开一个或多个实施例中，在确定所述虚拟机发送的 I/O 访问对应的内存页缺失之前，数据处理设备首先要接收到 I/O 访问，而一台虚拟机可以发送多个 I/O 访问，为使数据处理设备有序的进行 I/O 访问处理，将接收到的 I/O 访问添加至 I/O 队列中；而查找内存页是否缺失首先要获取 I/O 访问对应的虚拟地址，获得虚拟地址之后才能查找与该虚拟地址对应的物理地址是否存在。具体实现如下所述：

在所述确定所述虚拟机发送的 I/O 访问对应的内存页缺失的情况下之前，还包括：

所述数据处理设备，接收所述虚拟机发送的 I/O 访问，将所述 I/O 访问添加至 I/O 队列，并确定所述 I/O 访问对应的虚拟地址；

调用所述缺页请求处理设备中的状态表，查找与所述虚拟地址对应的内存页的存在状态；

相应地，所述确定所述虚拟机发送的 I/O 访问对应的内存页缺失，包括：

所述数据处理设备，根据所述内存页的存在状态，确定所述虚拟机发送的 I/O 访问对应的内存页缺失。

其中，I/O 队列，可以理解为存放一台虚拟机 I/O 访问的队列，遵循先入先出的原则；状态表可以理解为，存放 GPA 对应的内存页存在状态的表；虚拟地址可以理解为 GPA；内存页可以理解为，与虚拟地址对应的物理地址。

本公开实施例提供的该内存页缺页处理方法，数据处理设备可将接收的 I/O 访问添加至 I/O 队列中，以便有序的进行 I/O 访问处理，且数据处理设备可直接调用缺页请求处理设备中的状态表，查看与该虚拟地址对应的内存页的存在状态，提高查找内存页缺页的效率。

在本公开一个或多个实施例中，在确定所述虚拟机发送的 I/O 访问对应的内存页缺失之前，虚拟机首先要发送 I/O 访问，明确 I/O 访问的访问对象，而该访问对象位于数据处理设备中的，虚拟机发送 I/O 访问的访问对象为数据处理设备中的数据处理组件，以获得数据处理组件中的数据。具体实现如下所述：

所述数据处理设备包括数据处理组件；

相应地，所述接收所述虚拟机发送的 I/O 访问，并确定所述 I/O 访问对应的虚拟地址之前，还包括：

所述虚拟机，向所述数据处理设备发送、针对所述数据处理组件的 I/O 访问。

其中，数据处理组件可以理解为，数据处理设备中的网络设备，磁盘设备等，如 virtio-net，rdma，virtio-blk, NVMe 等。

本公开实施例提供的该内存页缺页处理方法，通过向数据处理设备发送，针对数据处理组件的I/O访问，可获取数据处理组件中的数据。

为了便于理解，以下实施例均以数据处理设备为DPU，缺页请求处理设备为VPRI-PCI设备，以I/O访问为对磁盘设备virtio-blk的读写访问为例，对本公开实施例提供的一种内存页缺页处理方法进行详细说明。

虚拟机向DPU发送，针对DPU中磁盘设备virtio-blk的读写访问，DPU接收虚拟机发送的I/O访问，将该I/O访问添加至该虚拟机对应的I/O队列中，并确定该I/O访问对应的虚拟地址；DPU可调用VPRI-PCI设备中的状态表，查找与该虚拟地址对应的内存页的存在状态，在对应的内存页存在状态为“0”，即DPU在确定虚拟机发送的I/O访问对应的内存页缺失的情况下，向VPRI-PCI设备发送缺页请求。

在本公开一个或多个实施例中，在确定虚拟机发送的I/O访问对应的内存页缺失之后，要中断I/O队列中位于该I/O访问之后的其他的I/O访问，使后续其他的I/O访问与该I/O访问统一查询更新后的映射页表，减少其他的I/O访问的内存页缺页概率，提高线程执行效率。具体实现方式如下所述：

所述确定所述虚拟机发送的I/O访问对应的内存页缺失之后，还包括：

所述缺页请求处理设备，中断所述I/O队列中所述I/O访问的处理、以及所述I/O队列中其他I/O访问的处理，其中，所述其他I/O访问在所述I/O队列中、位于所述I/O访问之后。

实际应用中，可通过VPRI-PCI设备的MSI-X中断方式中断I/O队列中该I/O访问的处理，以及I/O队列中其他I/O访问的处理。

例如，在对磁盘设备virtio-blk的读写访问出现内存页缺失的情况下，VPRI-PCI设备中断该对磁盘设备virtio-blk的读写访问所在的I/O队列，暂停该I/O队列中位于该读写访问后的I/O访问。在磁盘设备virtio-blk存在多个I/O队列的情况下，那该磁盘设备virtio-blk下的其他队列仍然可以继续工作；同样的，另外一个设备（例如virtio-net）中I/O队列中的I/O访问仍然继续工作，不受到磁盘设备virtio-blk缺页状态的影响（除非本队列自身访问的内存也遇到了缺页）。

本公开实施例提供的该内存页缺页处理方法，缺页请求处理设备中断I/O队列中该I/O访问的处理，以及I/O队列中其他I/O访问的处理，使后续其他I/O访问与该I/O访问统一查询更新后的映射页表，减少其他I/O访问的内存页缺页概率，提高线程执行效率。

步骤204：所述缺页请求处理设备，根据所述缺页请求发送缺页消息至缺页异常队列。

其中，缺页消息可以理解为，与缺页请求对应的，携带有与缺失的内存页对应的虚拟地址信息的消息；缺页异常队列，可以理解为存有所有虚拟机I/O访问缺页消息的队列。

沿用上例，所述缺页请求处理设备，根据所述缺页请求发送缺页消息至缺页异常队列，则可以理解，VPRI-PCI设备根据DPU发送的缺页请求，发送携带有与缺失的内存页对应的虚拟地址信息的消息至缺页异常队列中。

在本公开一个或多个实施例中，根据缺页请求发送缺页消息至缺页异常队列，缺页异常队列为驻留在主机内存中的共享内存，共享内存可以理解为，在多处理器的计算机系统中，可以被不同数据处理器访问的大容量内存,为 VPRI-PCI 设备与 VMM 之间的高速通信通道，可以实现以最小的延迟完成 I/O 缺页处理。具体实现方式如下所述：

所述根据所述缺页请求发送缺页消息至缺页异常队列，包括：

所述缺页请求处理设备，根据所述缺页请求发送缺页消息至所述主机内存中的共享内存的缺页异常队列。

沿用上例，所述缺页请求处理设备，根据所述缺页请求发送缺页消息至所述主机内存中的共享内存的缺页异常队列，则可以理解为，VPRI-PCI 设备根据缺页请求发送缺页消息至主机内存中的共享内存的缺页异常队列。

本公开实施例提供的该内存页缺页处理方法，通过将缺页消息发送至主机内存中的共享内存的缺页异常队列，可以实现以最小的延迟完成 I/O 缺页处理。

步骤 206：所述虚拟机管理器，获取所述缺页异常队列中的所述缺页消息，并根据所述缺页消息进行缺页处理，获得缺失的内存页，并向所述缺页请求处理设备发送针对所述缺失的内存页的处理完成指令。

其中，虚拟机管理器可以理解为上述实施例中的 VMM；缺页处理可以理解为，获取缺页消息中的虚拟地址，对该虚拟地址分配物理地址等操作；处理完成指令可以理解为，为缺页消息中虚拟地址，分配对应的物理地址的完成指令。

具体实施时，VMM 获取缺页异常队列中的缺页消息，并根据缺页消息中的虚拟地址进行缺页处理，获得与缺页消息中虚拟地址对应的物理地址，并向 VPRI-PCI 设备发送为缺页消息中虚拟地址分配物理地址的完成指令。

在本公开一个或多个实施例中，在虚拟机管理器中包含有与缺页请求处理设备对应的缺页请求处理驱动，该缺页请求处理驱动为一种可以使计算机和缺页请求处理设备进行互相通信的特殊程序，相当于硬件的接口，通过该缺页请求处理驱动从缺页异常队列中获取缺页消息。具体实现方式如下所述：

所述虚拟机管理器包括与所述缺页请求处理设备对应的缺页请求处理驱动；

相应地，所述获取所述缺页异常队列中的所述缺页消息，包括：

所述虚拟机管理器，通过所述缺页请求处理驱动获取所述缺页异常队列中的所述缺页消息。

其中，缺页请求处理驱动可以理解为，主机中存在与缺页请求处理设备对应的缺页请求处理驱动，该缺页请求处理驱动可以理解为，缺页请求处理设备与操作系统之间的接口，操作系统通过这个接口可以控制缺页请求处理设备的工作。

也就是说，在 VPRI-PCI 设备发送缺页消息至缺页异常队列之后，通过 VMM 中与 VPRI-PCI 设备对应的 VPRI-PCI 驱动，从缺页异常队列中获取 VPRI-PCI 设备发送的缺页消

息。

本公开实施例提供的该内存页缺页处理方法，通过缺页请求处理驱动获取缺页异常队列中的缺页消息，以使 VPRI-PCI 设备与操作系统连接，便于操作系统控制 VPRI-PCI 设备。

在本公开一个或多个实施例中，虚拟管理器进行缺页处理，通过虚拟机管理器中的内存管理单元进行缺页处理，内存管理单元获得缺页消息中的缺失内存页对应的虚拟地址，为该虚拟地址分配物理地址获得缺失的内存页。具体实现方式如下所述：

所述根据所述缺页消息进行缺页处理，获得缺失的内存页，包括：

所述虚拟机管理器中的内存管理单元，根据所述缺页消息中携带的缺失内存页对应的虚拟地址，确定与所述虚拟地址对应的内存页。

其中，内存管理单元可以理解为，对内存进行管理和分配的软件功能单元。

实际应用中，VMM 根据缺页消息中携带的、缺失的内存页对应的虚拟地址，通过 VMM 中的内存管理单元对缺失的内存页进行恢复，具体的，VMM 中的内存管理单元根据缺页消息中携带的缺失的内存页对应的虚拟地址，为该虚拟地址分配物理地址，获得与该虚拟地址对应的内存页，修改该虚拟机的 IOMMU 页表，来更新该虚拟机的 GPA 内存映射状态，以及映射的物理地址。

本公开实施例提供的该内存页缺页处理方法，通过虚拟机管理器中的内存管理单元进行缺页处理，以通过内存管理单元为缺页消息中，携带的缺失内存页对应的虚拟地址，分配合理的物理地址，获得与该虚拟地址对应的内存页。

步骤 208：所述缺页请求处理设备，根据所述处理完成指令更新状态表，以使所述数据处理设备根据更新后的状态表进行所述 I/O 访问处理，其中，所述状态表中包括缺失的内存页的存在状态。

仍然以缺页请求处理设备为 VPRI-PCI 设备为例，对本公开实施例提供的内存页缺页处理方法，进行详细说明。

所述缺页请求处理设备，根据所述处理完成指令更新状态表，以使所述数据处理设备根据更新后的状态表进行所述 I/O 访问处理，则可以理解为，VMM 在获得缺失的内存页之后，向 VPRI-PCI 设备发送处理完成指令，VPRI-PCI 设备接收到该处理完成指令之后，更新状态表，如将该缺失的内存页对应的状态置为“1”，来表示该内存页存在， DPU 根据更新后的状态表进行上述虚拟机的 I/O 访问处理。

在本公开一个或多个实施例中，缺页请求处理设备完成状态表更新之后，给数据处理设备发送处理完成的通知，以使数据处理设备根据该处理完成通知，快速处理 I/O 队列中 I/O 访问。具体实现方式如下所述：

所述根据所述处理完成指令更新状态表，以使所述数据处理设备根据更新状态表进行所述 I/O 访问处理，包括：

所述缺页请求处理设备，根据所述处理完成指令更新状态表，并发送针对更新后的状态

表的更新完成指令，至所述数据处理设备；

所述数据处理设备，响应于所述更新完成指令，根据更新后的状态表进行所述 I/O 访问处理。

实际应用中，VPRI-PCI 设备根据从 VMM 获得的处理完成指令，更新 VPRI-PCI 设备中的状态表，并发送更新完成指令至 DPU；DPU 响应于该更新完成指令，并根据更新后的状态表进行 I/O 访问的处理。

本公开实施例提供的该内存页缺页处理方法，缺页请求处理设备完成状态表更新之后，给数据处理设备发送更新完成指令，数据处理设备接收到该更新完成指令，并根据该更新完成指令，快速处理 I/O 队列中 I/O 访问。

在本公开一个或多个实施例中，虚拟机可以为一个，也可以为多个，在虚拟机为多个的情况下，可对多个虚拟机发送的 I/O 访问对应的缺页请求并发进行处理，以实现更高效的内存页缺页处理效率。具体实现方式如下所述：

所述虚拟机为至少两个；

相应地，所述根据所述缺页请求发送缺页消息至缺页异常队列，包括：

所述缺页请求处理设备，根据所述至少两个缺页请求发送至少两个缺页消息至缺页异常队列，其中，所述至少两个缺页请求为所述数据处理设备，在确定所述至少两个虚拟机发送的 I/O 访问对应的内存页均缺失的情况下，向所述缺页请求处理设备发送；

相应地，所述获取所述缺页异常队列中的所述缺页消息，并根据所述缺页消息进行缺页处理，获得缺失的内存页，并向所述缺页请求处理设备发送针对所述缺失的内存页的处理完成指令，包括：

所述虚拟机管理器，获取所述缺页异常队列中的所述至少两个缺页消息，并根据所述至少两个缺页消息进行缺页处理，获得缺失的至少两个内存页，并向所述缺页请求处理设备发送针对所述缺失的至少两个内存页的处理完成指令。

相应地，所述根据所述处理完成指令更新状态表，以使所述数据处理设备根据更新后的状态表进行所述 I/O 访问处理，包括：

所述缺页请求处理设备，根据所述处理完成指令更新至少两个状态表，以使所述数据处理设备根据更新后的至少两个状态表进行所述至少两个 I/O 访问处理，其中，所述状态表中包括缺失的至少两个内存页的存在状态。

实际应用中，以虚拟机为两个，分别为虚拟机 A，虚拟机 B；I/O 访问分别为虚拟机 A 对磁盘 A 的读写访问 A，虚拟机 B 对磁盘 B 的读写访问 B 为例，进行详细说明。

DPU 在确定虚拟机 A，虚拟机 B 发送的读写访问 A，读写访问 B 对应的内存页均缺失的情况下，向 VPRI-PCI 设备发送缺页请求 A，缺页请求 B，VPRI-PCI 设备根据缺页请求 A，缺页请求 B 发送缺页消息 A，缺页消息 B 至缺页异常队列。

VMM 获取缺页异常队列中的缺页消息 A，缺页消息 B，并根据该两个缺页消息进行缺页

处理，获得缺页消息 A 对应的缺失的内存页 A，以及缺页消息 B 对应的缺失的内存页 B，并向 VPRI-PCI 设备发送针对缺失的内存页 A、缺失的内存页 B 的处理完成指令。

VPRI-PCI 设备根据处理完成指令，更新虚拟机 A 对应状态表 A 中，缺失的内存页 A 的状态，以及更新虚拟机 B 对应状态表 B 中，缺失的内存页 B 的状态，以使 DPU 根据更新后的状态表 A，更新后的状态表 B 进行读写访问 A，读写访问 B 的处理。

本公开实施例提供的该内存页缺页处理方法，在虚拟机为多个的情况下，可对多个虚拟机发送的 I/O 访问对应的缺页请求进行处理，以实现更高效的内存页缺页处理效率。

在本公开一个或多个实施例中，在虚拟机为多个的情况下，在获得多个虚拟机对应的多个缺失的内存页之后，更新各个虚拟机对应的状态表，更新完成后可通过缺页请求处理接口乱序的发送对应的更新完成指令至数据处理设备，以使数据处理设备快速处理多个虚拟机对应的 I/O 访问。具体实现方式如下所述：

所述缺页请求处理设备中包括缺页请求处理接口；

相应的，所述根据所述处理完成指令更新至少两个状态表，以使所述数据处理设备根据更新后的至少两个状态表进行所述至少两个 I/O 访问处理，包括：

所述缺页请求处理设备，根据所述处理完成指令更新至少两个状态表，并通过所述缺页请求处理接口发送针对所述更新后的至少两个状态表的至少两个更新完成指令，至所述数据处理设备；

所述数据处理设备，响应于所述至少两个更新完成指令，根据后的至少两个状态表进行所述 I/O 访问处理。

其中，缺页请求处理接口可以理解为，页面响应接口，该页面响应接口设置为将更新完成指令发送至数据处理设备。

具体的，沿用上例，在 VPRI-PCI 设备，根据针对缺失的内存页 A 的处理完成指令更新状态表 A，根据针对缺失的内存页 B 的处理完成指令更新状态表 B 之后，可通过 VPRI-PCI 设备的页面响应接口，发送对应更新后的状态表 A 的更新完成指令 A，以及对应更新后的状态表 B 的更新完成指令 B 至 DPU；DPU 响应于该更新完成指令 A，以及更新完成指令 B，根据更新后的状态表 A，更新后的状态表 B 进行读写访问 A，读写访问 B 的处理。

实际应用中，VPRI-PCI 设备的页面响应接口，可以乱序发送对应更新后的状态表 A 的更新完成指令 A，以及对应更新后的状态表 B 的更新完成指令 B，不必按照从缺页异常队列中取出的顺序发送更新完成指令，提高了处理效率。

本公开实施例提供的该内存页缺页处理方法，通过缺页请求处理接口乱序的发送对应的更新完成指令至数据处理设备，以使数据处理设备快速处理多个虚拟机对应的 I/O 访问。

基于此，本公开实施例提供的该内存页缺页处理方法，通过在数据处理设备中引入缺页请求处理设备，在数据处理设备向缺页请求处理设备发送缺页请求的情况下，缺页请求处理设备根据缺页请求发送缺页消息至虚拟机管理器，实现不依赖于 IOMMU 的 I/O 缺页请求接

口，缩短 I/O 内存页缺页恢复的处理链路，提高 I/O 内存页缺页恢复的处理效率，降低设备访问内存延迟。

参见图 3，图 3 示出本公开一个实施例提供的一种应用内存页缺页处理方法的 DPU 辅助虚拟化架构下的 VPRI 架构示意图，下面结合具体实施例进行介绍。

该图示出了一种应用内存页缺页处理方法的 DPU 辅助虚拟化架构下的 VPRI 架构，具体包括服务器和 DPU，其中，服务器中包括主机、运行在主机中的多个虚拟机，如虚拟机 1、虚拟机 2...虚拟机 N 等，虚拟机管理器，通过设备接口连接 DPU，DPU 中包括 DPU IO 引擎、VPRI-PCI 设备以及一些其他 PCI 设备，如网络、存储器、磁盘设备等。

该 VPRI 架构中的存在有一个独立的 PCI 功能，即 VPRI-PCI 设备。VPRI-PCI 设备为 DPU 为所有其他网络和存储 PCI 设备一起提供或者模拟的 PCI 设备之一，VPRI-PCI 设备独立于 DPU 上的任何其他 PCI 设备。

然而，其他 PCI 设备背后的 IO 引擎依赖于 VPRI-PCI 设备来检测内存页的页面存在属性，因为 VPRI-PCI 设备提供内存页的页面是否被映射的检测方法，例如在 VPRI-PCI 设备中存放有状态表，存放每个虚拟机每个内存页的映射存在状态（映射/未映射），如虚拟机 1 对应的状态表、虚拟机 2 对应的状态表...虚拟机 N 对应的状态表。因此其他 PCI 设备在进行 DMA 前，均需要首先通过 VPRI-PCI 设备内状态表的页面映射信息，来确定即将访问的内存是否为缺页状态，需要说明的是，VPRI-PCI 设备可将 IOPF 消息发送到虚拟机管理器。

VPRI-PCI 设备提供 MMIO 设备接口及标准 MSI-X 中断，其中， MMIO 可以理解为，一个 PCI 设备的寄存器，通过内存映射的方式供主机端的 CPU 访问 PCI 设备，在这些基础 PCI 模型之上，实现了虚拟机管理器至 VPRI-PCI 设备之间的 IOPF 共享内存队列协商，这个队列称为 Page Fault Queue(PFQ)，即上述实施例中的缺页异常队列。PFQ 可以理解为，驻留在主机内存中的共享内存，与 MSI-X 中断配对的 PFQ 可以理解为，VPRI-PCI 设备和 VMM 之间的高速通信通道，可以实现以最小的延迟完成 I/O 缺页的处理。

具体的，虚拟机通过设备直通的方式可直接访问 DPU 中的 PCI 设备，例如虚拟机 1 可直接访问 DPU 中与虚拟机 1 对应的网络、存储器、磁盘设备；在虚拟机 1 对这些 PCI 设备进行 I/O 访问的情况下，这些 PCI 设备通过 I/O 处理，也就是通过 DPU I/O 引擎调用 VPRI-PCI 设备中，与虚拟机 1 对应的状态表，进行页面查找，在确定虚拟机 1 发送的 I/O 访问对应的内存页缺失的情况下，由 DPU I/O 引擎发送页面请求至 VPRI-PCI 设备，VPRI-PCI 设备将页面请求中的缺页消息入列到 PFQ 中，VPRI-PCI 设备对应的 VPRI-PCI 驱动将缺页消息从 PFQ 中取出，实现消息出列，将该缺页消息发送至虚拟机管理器，在虚拟机管理器中进行缺页处理，获得缺失的内存页，进行映射更新，完成状态表更新。

基于此，本公开实施例提供的内存页缺页处理方法通过在 DPU 中增加独立于 DPU 的旁路 Function，在出现内存页缺页的情况下，依据此旁路 Function 发送缺页消息，实现不依赖 IOMMU PRI 的旁路 PCI Function 提供 PRI 请求服务，缩短 I/O 内存页缺页恢复的处理链路，提高 I/O 内存页缺页恢复的处理效率，降低设备访问内存延迟。

参见图 4，图 4 示出本公开一个实施例提供的一种内存页缺页处理方法的处理过程流程图。

基于图 3 中的架构，对该内存页缺页处理方法，在虚拟机的 I/O 访问对应的内存页缺页的情况的，具体处理过程进行详细说明。

其中，图 4 中的虚拟机 A 和虚拟机 B 可以用图 3 中的虚拟机 1 和虚拟机 2 替代。在虚拟机 A 通过 DMA 的方式对磁盘 A 进行读写访问，虚拟机 B 通过 DMA 的方式对磁盘 B 进行读写访问的情况下，一个 DMA job（DMA 工作）可以理解为，上述实施例中一个 I/O 请求，DMA 队列可以理解为，上述实施例中的 I/O 队列。

步骤一：遇到页面错误。

在 DPU 的 IO 引擎通过一些方式（如 ATS translation，或者设备上的页表镜像等）识别到其中一个 IO 队列（例如 virtio-net、virtio-blk、NVMe、RDMA）的 IO 请求遇到页面错误的情况下，如图中的虚拟机 A 对应的 DMA 队列中某一 DMA job，以及虚拟机 B 对应的 DMA 队列中某一 DMA job 出现页面错误；遇到页面错误即上述实施例中的虚拟机发送的 I/O 访问对应的内存页缺失。

具体的，通过 DPU 的 I/O 引擎识别 I/O 队列的 I/O 请求遇到页面错误。

步骤二：提交页面请求。

其中，页面请求可以理解为，上述实施例中的缺页请求。

在遇到页面错误的情况下，I/O 引擎生成一次 IOPF 消息，将该 IOPF 消息发送至 PFQ 队列，并通过 VPRI-PCI 设备中断 I/O 处理，按照 DMA 队列的排队顺序，将出现页面错误 DMA job 中下面的 DMA job 锁定。

具体的，为确保 I/O 队列语义一致性，在页面错误处理完成之前，遇到 IOPF 设备的 I/O 队列被停止处理，I/O 引擎只暂停发生 IOPF 的 I/O 队列，其他 I/O 队列不受影响。

具体实现可参见上述实施例，在此不再赘述。

步骤三：并发处理 IOPF。

具体的，针对页面错误的排队顺序，并发处理 IOPF，将虚拟机 A 对应的页面请求 A，和虚拟机 B 对应的页面请求 B 在虚拟机管理器的页面请求线程池中同时处理，在页面请求线程 A 处理虚拟机 A 出现的页面请求 A，在页面请求线程 B 处理虚拟机 B 出现的页面请求 B。

步骤 4：无序的页面响应。

其中，页面响应可以理解为，上述实施例中的更新完成指令。

具体的，VPRI-PCI 设备提供的页面响应（Page response）接口可处理乱序的 IOPF 完成事件，确保各虚拟机之间不会出现堵塞和干扰。

具体实现可参见上述实施例，在此不再赘述。

步骤 5：处理下一个 I/O。

在完成出现页面错误的I/O访问之后，解除对I/O队列的中断，继续按照I/O队列的排队顺序处理下一个I/O。

具体实现可参见上述实施例，在此不再赘述。

本公开实施例提供的内存页缺页处理方法，通过在数据处理设备中引入缺页请求处理设备，在数据处理设备向缺页请求处理设备发送缺页请求的情况下，缺页请求处理设备根据缺页请求发送缺页消息至虚拟机管理器，实现不依赖于IOMMU的I/O缺页请求接口，缩短I/O内存页缺页恢复的处理链路，提高I/O内存页缺页恢复的处理效率，降低设备访问内存延迟。

参见图5，图5示出本公开一个实施例提供的一种缺页请求处理设备的结构示意图。如图5所示，该装置包括：

缺页消息发送模块 502，被配置为根据缺页请求发送缺页消息至缺页异常队列，其中，所述缺页消息为数据处理设备在确定虚拟机发送的 I/O 访问对应的内存页缺失的情况下发送的。

状态表更新模块 504，被配置为根据处理完成指令更新状态表，以使所述数据处理设备根据更新后的状态表进行所述 I/O 访问处理，其中，所述处理完成指令为虚拟机管理器，在获取所述缺页异常队列中的缺页消息，并根据所述缺页消息进行缺页处理，获得缺失的内存页的情况下发送的，所述状态表中包括缺失的内存页的存在状态。

具体的，该缺页请求处理设备可以根据缺页请求发送缺页消息至虚拟机管理器，实现不依赖于IOMMU的I/O缺页请求接口，缩短I/O内存页缺页恢复的处理链路，提高I/O内存页缺页恢复的处理效率，降低设备访问内存延迟。

与上述方法实施例相对应，本公开还提供了内存页缺页系统实施例，图6示出了本公开一个实施例提供的一种内存页缺页系统的结构示意图。如图6所示，该系统包括：

数据处理设备602，被配置为在确定所述虚拟机发送的I/O访问对应的内存页缺失的情况下，向所述缺页请求处理设备发送所述缺页请求；

缺页请求处理设备604，被配置为根据所述缺页请求发送缺页消息至缺页异常队列；

虚拟机管理器 606，被配置为获取所述缺页异常队列中的所述缺页消息，并根据所述缺页消息进行缺页处理，获得缺失的内存页，并向所述缺页请求处理设备发送针对所述缺失的内存页的处理完成指令；

缺页请求处理设备 604，被配置为根据所述处理完成指令更新状态表，以使所述数据处理设备根据更新后的状态表进行所述 I/O 访问处理，其中，所述状态表中包括所述缺失的内存页的存在状态。

可选地，所述数据处理设备602，进一步被配置为：

接收所述虚拟机发送的I/O访问，将所述I/O访问添加至I/O队列，并确定所述I/O访问对应的虚拟地址；

调用所述缺页请求处理设备中的状态表，查找与所述虚拟地址对应的内存页的存在状态；

根据所述内存页的存在状态，确定所述虚拟机发送的I/O访问对应的内存页缺失。

可选地，所述系统，还包括：

虚拟机，被配置为向所述数据处理设备发送、针对所述数据处理组件的I/O访问。

可选地，所述缺页请求处理设备604，进一步被配置为：

中断所述I/O队列中所述I/O访问的处理、以及所述I/O队列中其他I/O访问的处理，其中，所述其他I/O访问在所述I/O队列中、位于所述I/O访问之后。

可选地，所述虚拟机管理器606，进一步被配置为：

通过所述缺页请求处理驱动获取所述缺页异常队列中的所述缺页消息。

可选地，所述虚拟机管理器606，进一步被配置为：

根据所述缺页消息中携带的缺失内存页对应的虚拟地址，确定与所述虚拟地址对应的内存页。

可选地，所述缺页请求处理设备604，进一步被配置为：

根据所述处理完成指令更新状态表，并发送针对更新后的状态表的更新完成指令，至所述数据处理设备；

所述数据处理设备，响应于所述更新完成指令，根据更新后的状态表进行所述I/O访问处理。

可选地，所述系统，还包括：

所述缺页请求处理设备，根据所述至少两个缺页请求发送至少两个缺页消息至缺页异常队列，其中，所述至少两个缺页请求为所述数据处理设备，在确定所述至少两个虚拟机发送的I/O访问对应的内存页均缺失的情况下，向所述缺页请求处理设备发送；

所述虚拟机管理器，获取所述缺页异常队列中的所述至少两个缺页消息，并根据所述至少两个缺页消息进行缺页处理，获得缺失的至少两个内存页，并向所述缺页请求处理设备发送针对所述缺失的至少两个内存页的处理完成指令。

所述缺页请求处理设备，根据所述处理完成指令更新至少两个状态表，以使所述数据处理设备根据更新后的至少两个状态表进行所述至少两个I/O访问处理，其中，所述状态表中包括所述缺失的至少两个内存页的存在状态。

所述缺页请求处理设备，根据所述处理完成指令更新至少两个状态表，并通过所述缺页请求处理接口发送针对所述更新后的至少两个状态表的至少两个更新完成指令，至所述数据处理设备；

所述数据处理设备，响应于所述至少两个更新完成指令，根据更新后的至少两个状态表进行所述I/O访问处理。

可选地，所述缺页请求处理设备604，进一步被配置为：

根据所述缺页请求发送缺页消息至所述主机内存中的共享内存的缺页异常队列。

具体的，该内存页缺页处理系统，通过在数据处理设备中引入缺页请求处理设备，在数据处理设备向缺页请求处理设备发送缺页请求的情况下，缺页请求处理设备根据缺页请求发送缺页消息至虚拟机管理器，实现不依赖于 IOMMU 的 I/O 缺页请求接口，缩短 I/O 内存页缺页恢复的处理链路，提高 I/O 内存页缺页恢复的处理效率，降低设备访问内存延迟。

上述为本实施例的一种内存页缺页系统的示意性方案。需要说明的是，该内存页缺页系统的技术方案与上述的内存页缺页系统的技术方案属于同一构思，内存页缺页系统的技术方案未详细描述的细节内容，均可以参见上述内存页缺页方法的技术方案的描述。

图 7 示出了根据本公开一个实施例提供的一种计算设备 700 的结构框图。该计算设备 700 的部件包括但不限于存储器 710 和处理器 720。处理器 720 与存储器 710 通过总线 730 相连接，数据库 750 设置为保存数据。

计算设备 700 还包括接入设备 740，接入设备 740 使得计算设备 700 能够经由一个或多个网络 760 通信。这些网络的示例包括公用交换电话网（PSTN，Public Switched Telephone Network）、局域网（LAN，Local Area Network）、广域网（WAN，Wide Area Network）、个域网（PAN，Personal Area Network）或诸如因特网的通信网络的组合。接入设备 740 可以包括有线或无线的任何类型的网络接口（例如，网络接口卡（NIC，network interface controller））中的一个或多个，诸如 IEEE802.11 无线局域网（WLAN，Wireless Local Area Network）无线接口、全球微波互联接入（Wi-MAX，Worldwide Interoperability for Microwave Access）接口、以太网接口、通用串行总线（USB，Universal Serial Bus）接口、蜂窝网络接口、蓝牙接口、近场通信（NFC，Near Field Communication）。

在本公开的一个实施例中，计算设备 700 的上述部件以及图 7 中未示出的其他部件也可以彼此相连接，例如通过总线。应当理解，图 7 所示的计算设备结构框图是出于示例的目的，而不是对本公开范围的限制。本领域技术人员可以根据需要，增添或替换其他部件。

计算设备 700 可以是任何类型的静止或移动计算设备，包括移动计算机或移动计算设备（例如，平板计算机、个人数字助理、膝上型计算机、笔记本计算机、上网本等）、移动电话（例如，智能手机）、可佩戴的计算设备（例如，智能手表、智能眼镜等）或其他类型的移动设备，或者诸如台式计算机或个人计算机（PC，Personal Computer）的静止计算设备。计算设备 700 还可以是移动式或静止式的服务器。

其中，处理器 720 设置为执行如下计算机可执行指令，该计算机可执行指令被处理器执行时实现上述内存页缺页方法的步骤。

上述为本实施例的一种计算设备的示意性方案。需要说明的是，该计算设备的技术方案与上述的内存页缺页方法的技术方案属于同一构思，计算设备的技术方案未详细描述的细节内容，均可以参见上述内存页缺页方法的技术方案的描述。

本公开一实施例还提供一种计算机可读存储介质，其存储有计算机可执行指令，该计算机可执行指令被处理器执行时实现上述内存页缺页方法的步骤。

上述为本实施例的一种计算机可读存储介质的示意性方案。需要说明的是，该存储介质的技术方案与上述的内存页缺页方法的技术方案属于同一构思，存储介质的技术方案未详细描述的细节内容，均可以参见上述内存页缺页方法的技术方案的描述。

本公开一实施例还提供一种计算机程序，其中，当所述计算机程序在计算机中执行时，令计算机执行上述内存页缺页方法的步骤。

上述为本实施例的一种计算机程序的示意性方案。需要说明的是，该计算机程序的技术方案与上述的内存页缺页方法的技术方案属于同一构思，计算机程序的技术方案未详细描述的细节内容，均可以参见上述内存页缺页方法的技术方案的描述。

上述对本公开特定实施例进行了描述。其它实施例在所附权利要求书的范围内。在一些情况下，在权利要求书中记载的动作或步骤可以按照不同于实施例中的顺序来执行并且仍然可以实现期望的结果。另外，在附图中描绘的过程不一定要求示出的特定顺序或者连续顺序才能实现期望的结果。在某些实施方式中，多任务处理和并行处理也是可以的或者可能是有利的。

所述计算机指令包括计算机程序代码，所述计算机程序代码可以为源代码形式、对象代码形式、可执行文件或某些中间形式等。所述计算机可读介质可以包括：能够携带所述计算机程序代码的任何实体或装置、记录介质、U 盘、移动硬盘、磁碟、光盘、计算机存储器、只读存储器（ROM, Read-Only Memory）、随机存取存储器（RAM, Random Access Memory）、电载波信号、电信信号以及软件分发介质等。需要说明的是，所述计算机可读介质包含的内容可以根据专利实践的要求进行适当的增减，例如在某些地区，根据专利实践，计算机可读介质不包括电载波信号和电信信号。

需要说明的是，对于前述的各方法实施例，为了简便描述，故将其都表述为一系列的动作组合，但是本领域技术人员应该知悉，本公开实施例并不受所描述的动作顺序的限制，因为依据本公开实施例，某些步骤可以采用其它顺序或者同时进行。其次，本领域技术人员也应该知悉，公开实施例中所描述的实施例均属于优选实施例，所涉及的动作和模块并不一定都是本公开实施例所必须的。

在上述实施例中，对各个实施例的描述都各有侧重，某个实施例中没有详述的部分，可以参见其它实施例的相关描述。

以上公开的本公开优选实施例只是用于帮助阐述本公开。可选实施例并没有详尽叙述所有的细节，也不限制该发明为所述的具体实施方式。显然，根据本公开实施例的内容，可作很多的修改和变化。本公开选取并具体描述这些实施例，是为了更好地解释本公开实施例的原理和实际应用，从而使所属技术领域技术人员能很好地理解和利用本公开。本公开受权利要求书及其全部范围和等效物的限制。

工业实用性

本公开实施例提出了一种内存页缺页处理方法以及系统，该内存页缺页处理方法，通过在数据处理设备中引入缺页请求处理设备，在数据处理设备向缺页请求处理设备发送缺页请

求的情况下，缺页请求处理设备根据缺页请求发送缺页消息至虚拟机管理器，实现不依赖于IOMMU 的 I/O 缺页请求接口，缩短 I/O 内存页缺页恢复的处理链路，提高 I/O 内存页缺页恢复的处理效率，降低设备访问内存延迟。

权 利 要 求 书

1.一种内存页缺页处理方法，应用于内存缺页处理系统，所述内存缺页处理系统包括主机、运行在主机中的虚拟机和虚拟机管理器、部署在所述主机中对所述虚拟机的 I/O 访问进行处理的数据处理设备、部署在所述主机中对所述数据处理设备发送的缺页请求进行处理的缺页请求处理设备，其中，所述方法包括：

所述数据处理设备，在确定所述虚拟机发送的 I/O 访问对应的内存页缺失的情况下，向所述缺页请求处理设备发送所述缺页请求；

所述缺页请求处理设备，根据所述缺页请求发送缺页消息至缺页异常队列；

所述虚拟机管理器，获取所述缺页异常队列中的所述缺页消息，并根据所述缺页消息进行缺页处理，获得缺失的内存页，并向所述缺页请求处理设备发送针对所述缺失的内存页的处理完成指令；

所述缺页请求处理设备，根据所述处理完成指令更新状态表，以使所述数据处理设备根据更新后的状态表进行所述 I/O 访问处理，其中，所述状态表中包括所述缺失的内存页的存在状态。

2.根据权利要求 1 所述的内存页缺页处理方法，在所述确定所述虚拟机发送的 I/O 访问对应的内存页缺失的情况下之前，还包括：

所述数据处理设备，接收所述虚拟机发送的 I/O 访问，将所述 I/O 访问添加至 I/O 队列，并确定所述 I/O 访问对应的虚拟地址；

调用所述缺页请求处理设备中的状态表，查找与所述虚拟地址对应的内存页的存在状态；

相应地，所述确定所述虚拟机发送的 I/O 访问对应的内存页缺失，包括：

所述数据处理设备，根据所述内存页的存在状态，确定所述虚拟机发送的 I/O 访问对应的内存页缺失。

3.根据权利要求 2 所述的内存页缺页处理方法，所述数据处理设备包括数据处理组件；

相应地，所述接收所述虚拟机发送的 I/O 访问，并确定所述 I/O 访问对应的虚拟地址之前，还包括：

所述虚拟机，向所述数据处理设备发送、针对所述数据处理组件的 I/O 访问。

4.根据权利要求 2 所述的内存页缺页处理方法，所述确定所述虚拟机发送的 I/O 访问对应的内存页缺失之后，还包括：

所述缺页请求处理设备，中断所述 I/O 队列中所述 I/O 访问的处理、以及所述 I/O 队列中其他 I/O 访问的处理，其中，所述其他 I/O 访问在所述 I/O 队列中、位于所述 I/O 访问之后。

5.根据权利要求 1 所述的内存页缺页处理方法，所述虚拟机管理器包括与所述缺页请求处理设备对应的缺页请求处理驱动；

相应地，所述获取所述缺页异常队列中的所述缺页消息，包括：

所述虚拟机管理器，通过所述缺页请求处理驱动获取所述缺页异常队列中的所述缺页消

6.根据权利要求 1 所述的内存页缺页处理方法，所述根据所述缺页消息进行缺页处理，获得缺失的内存页，包括：

所述虚拟机管理器中的内存管理单元，根据所述缺页消息中携带的缺失内存页对应的虚拟地址，确定与所述虚拟地址对应的内存页。

7.根据权利要求 2 所述的内存页缺页处理方法，所述根据所述处理完成指令更新状态表，以使所述数据处理设备根据更新状态表进行所述 I/O 访问处理，包括：

所述缺页请求处理设备，根据所述处理完成指令更新状态表，并发送针对更新后的状态表的更新完成指令，至所述数据处理设备；

所述数据处理设备，响应于所述更新完成指令，根据更新后的状态表进行所述 I/O 访问处理。

8.根据权利要求 1 所述的内存页缺页处理方法，所述虚拟机为至少两个；

相应地，所述根据所述缺页请求发送缺页消息至缺页异常队列，包括：

所述缺页请求处理设备，根据所述至少两个缺页请求发送至少两个缺页消息至缺页异常队列，其中，所述至少两个缺页请求为所述数据处理设备，在确定所述至少两个虚拟机发送的 I/O 访问对应的内存页均缺失的情况下，向所述缺页请求处理设备发送；

相应地，所述获取所述缺页异常队列中的所述缺页消息，并根据所述缺页消息进行缺页处理，获得缺失的内存页，并向所述缺页请求处理设备发送针对所述缺失的内存页的处理完成指令，包括：

所述虚拟机管理器，获取所述缺页异常队列中的所述至少两个缺页消息，并根据所述至少两个缺页消息进行缺页处理，获得缺失的至少两个内存页，并向所述缺页请求处理设备发送针对所述缺失的至少两个内存页的处理完成指令。

9.根据权利要求 8 所述的内存页缺页处理方法，所述虚拟机为至少两个；

相应地，所述根据所述处理完成指令更新状态表，以使所述数据处理设备根据更新后的状态表进行所述 I/O 访问处理，包括：

所述缺页请求处理设备，根据所述处理完成指令更新至少两个状态表，以使所述数据处理设备根据更新后的至少两个状态表进行所述至少两个 I/O 访问处理，其中，所述状态表中包括所述缺失的至少两个内存页的存在状态。

10.根据权利要求 9 所述的内存页缺页处理方法，所述缺页请求处理设备中包括缺页请求处理接口；

相应的，所述根据所述处理完成指令更新至少两个状态表，以使所述数据处理设备根据更新后的至少两个状态表进行所述至少两个 I/O 访问处理，包括：

所述缺页请求处理设备，根据所述处理完成指令更新至少两个状态表，并通过所述缺页

请求处理接口发送针对所述更新后的至少两个状态表的至少两个更新完成指令，至所述数据处理设备；

所述数据处理设备，响应于所述至少两个更新完成指令，根据更新后的至少两个状态表进行所述 I/O 访问处理。

11.根据权利要求 1 所述的内存页缺页处理方法，所述根据所述缺页请求发送缺页消息至缺页异常队列，包括：

所述缺页请求处理设备，根据所述缺页请求发送缺页消息至所述主机内存中的共享内存的缺页异常队列。

12.一种内存缺页处理系统，包括：

所述数据处理设备，被配置为在确定所述虚拟机发送的 I/O 访问对应的内存页缺失的情况下，向所述缺页请求处理设备发送所述缺页请求；

所述缺页请求处理设备，被配置为根据所述缺页请求发送缺页消息至缺页异常队列；

所述虚拟机管理器，被配置为获取所述缺页异常队列中的所述缺页消息，并根据所述缺页消息进行缺页处理，获得缺失的内存页，并向所述缺页请求处理设备发送针对所述缺失的内存页的处理完成指令；

所述缺页请求处理设备，被配置为根据所述处理完成指令更新状态表，以使所述数据处理设备根据更新后的状态表进行所述 I/O 访问处理，其中，所述状态表中包括缺失的内存页的存在状态。

13.一种缺页请求处理设备，包括缺页消息发送模块以及状态表更新模块，其中：

所述缺页消息发送模块，被配置为根据缺页请求发送缺页消息至缺页异常队列，其中，所述缺页消息为数据处理设备在确定虚拟机发送的 I/O 访问对应的内存页缺失的情况下发送的；

所述状态表更新模块，被配置为根据处理完成指令更新状态表，以使所述数据处理设备根据更新后的状态表进行所述 I/O 访问处理，其中，所述处理完成指令为虚拟机管理器，在获取所述缺页异常队列中的缺页消息，并根据所述缺页消息进行缺页处理，获得缺失的内存页的情况下发送的，所述状态表中包括缺失的内存页的存在状态。

14.一种计算设备，包括：

存储器和处理器；

所述存储器设置为存储计算机可执行指令，所述处理器设置为执行所述计算机可执行指令，

该计算机可执行指令被处理器执行时实现权利要求 1 至 11 任意一项所述目标对象确定方法的步骤。