基于半虚拟化设备的数据处理方法、装置和系统

本申请要求于 2022 年 02 月 18 日提交中国专利局、申请号为 202210153414.2、申请名称为“基于半虚拟化设备的数据处理方法、装置和系统”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本申请涉及虚拟化技术领域，具体而言，涉及一种基于半虚拟化设备的数据处理方法、装置和系统。

背景技术

目前，软硬结合的 virtio 设备虚拟化实现中，主机和设备之间通过 PCIe（Peripheral Component Interconnect express,高速串行计算机扩展总线标准）连接。根据 virtio 设备规范，当设备收到数据后，需要经过多个步骤提交给 CPU（Central Processing Unit，中央处理器），每个步骤均是通过 DMA（Direct Memory Access，直接存储器访问）的方式写主机内存。但是，当每个步骤过于频繁时，CPU PCIe 子系统出现瓶颈，设备侧会看到 PCIe 接口反压，DMA 性能下降。

针对上述的问题，目前尚未提出有效的解决方案。

发明内容

本申请实施例提供了一种基于半虚拟化设备的数据处理方法、装置和系统，以至少解决相关技术中半虚拟化设备与主机之前交互频繁，导致 DMA 性能下降的技术问题。

根据本申请实施例的一个方面，提供了一种基于半虚拟化设备的数据处理方法，包括：获取半虚拟化设备的完成队列中存储的多个初始数据，其中，多个初始数据用于表征半虚拟化设备已经处理完成，但未提交至主机的原始数据的描述信息；确定多个初始数据中满足预设条件的多个第一数据；对多个第一数据进行聚合操作，生成第一聚合结果；发送携带有第一聚合结果的直接存储器访问请求至主机的内存。

根据本申请实施例的另一方面，还提供了一种基于半虚拟化设备的数据处理装置，包括：数据获取模块，用于获取半虚拟化设备的完成队列中存储的多个初始数据，其中，多个初始数据用于表征半虚拟化设备已经处理完成，但未提交至主机的原始数据的描述信息；数据确定模块，用于确定多个初始数据中满足预设条件的多个第一数据；聚合模块，用于对多个第一数据进行聚合操作，生成第一聚合结果；发送模块，用于发送携带有第一聚合结果的直接存储器访问请求至主机的内存。

根据本申请实施例的另一方面，还提供了一种基于半虚拟化设备的数据处理系统，包括：主机，包括：内存和完成队列；半虚拟化设备，与主机连接，用于获取完成队列中存储的多个初始数据，其中，多个初始数据用于表征半虚拟化设备已经处理完成，但未提交至主机的原始数据的描述信息；确定多个初始数据中满足预设条件的多个第一数据；对多

个第一数据进行聚合操作，生成第一聚合结果；发送携带有第一聚合结果的直接存储器访问请求至主机的内存。

根据本申请实施例的另一方面，还提供了一种计算机可读存储介质，计算机可读存储介质包括存储的程序，其中，在程序运行时控制计算机可读存储介质所在设备执行上述的基于半虚拟化设备的数据处理方法。

根据本申请实施例的另一方面，还提供了一种计算机终端，包括：存储器和处理器，处理器用于运行存储器中存储的程序，其中，程序运行时执行上述的基于半虚拟化设备的数据处理方法。

在本申请实施例中，当需要更新 used ring 时，可以获取 used 中存储的多个初始数据，然后从多个初始数据中筛选出符合预设条件的多个第一数据，对多个第一数据进行聚合操作生成第一聚合结果，并发送携带有该第一聚合结果的 DMA 请求至内存，达到一次性更新 used ring 的多个队列项的目的。容易注意到的是，通过聚合操作的方式发起一次 DMA 请求，无需针对每个队列项发起一次 DMA 请求，从而达到了减少更新 used ring 所产生的操作次数，避免设备侧看到 PCIe 接口反压，提高 DMA 性能的技术效果，进而解决了相关技术中半虚拟化设备与主机之前交互频繁，导致 DMA 性能下降的技术问题。

附图说明

此处所说明的附图用来提供对本申请的进一步理解，构成本申请的一部分，本申请的示意性实施例及其说明用于解释本申请，并不构成对本申请的不当限定。在附图中：

图 1 是根据现有技术的一种更新 used ring 的示意图；

图 2 是根据本申请实施例的一种用于实现基于半虚拟化设备的数据处理方法的计算机终端（或移动设备）的硬件结构框图；

图 3 是根据本申请实施例的一种基于半虚拟化设备的数据处理方法的流程图；

图 4 是根据本申请实施例的一种可选的软硬结合的 virtio 设备虚拟化实现架构的示意图；

图 5 是根据本申请实施例的一种可选的更新 used ring 的示意图；

图 6 是根据本申请实施例的一种可选的按 queue 调度的示意图；

图 7 是根据本申请实施例的一种基于半虚拟化设备的数据处理装置的示意图；

图 8 是根据本申请实施例的一种基于半虚拟化设备的数据处理系统的示意图；

图 9 是根据本申请实施例的一种计算机终端的结构框图。

具体实施方式

为了使本技术领域的人员更好地理解本申请方案，下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅仅是本申请一部分的实施例，而不是全部的实施例。基于本申请中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都应当属于本申请保护的范围。

需要说明的是，本申请的说明书和权利要求书及上述附图中的术语“第一”、“第二”

等是用于区别类似的对象，而不必用于描述特定的顺序或先后次序。应该理解这样使用的数据在适当情况下可以互换，以便这里描述的本申请的实施例能够以除了在这里图示或描述的那些以外的顺序实施。此外，术语“包括”和“具有”以及他们的任何变形，意图在于覆盖不排他的包含，例如，包含了一系列步骤或单元的过程、方法、系统、产品或设备不必限于清楚地列出的那些步骤或单元，而是可包括没有清楚地列出的或对于这些过程、方法、产品或设备固有的其它步骤或单元。

首先，在对本申请实施例进行描述的过程中出现的部分名词或术语适用于如下解释：

virtio：virtio 是一种 I/O 半虚拟化解决方案，是一套通用 I/O 设备虚拟化的程序，是对半虚拟化 Hypervisor 中的一组通用 I/O 设备的抽象。使用 virtio 协议的设备称为 virtio 设备。

数据 buffer：存储有设备收到的数据。

used ring：是 virtio 设备的完成队列，当设备完成了一个驱动下发的请求后，硬件通过提交 used ring 来通知驱动命令已经完成。used ring 指向数据 buffer 的结构体，仅包含对数据的描述信息（地址、长度等等）。

used ring index：是 used ring 的指针。

目前，当前设备收到数据后，提交给 CPU 需要经过如下三个步骤：写数据 buffer；写 used ring；写 used ring index。例如，以 used ring 的提交为例进行说明，如图 1 所示，used ring 包含有 8 个队列项，已提交的队列项用实心方框所示，未提交的队列项用空心方框表示。现有写 used ring 的步骤如下：将队列项 0 提交给 CPU，used ring index 更新为 1，表明从队列 1 开始未提交；将队列 1 提交给 CPU，used ring index 更新为 2，表明从队列 2 开始未提交；将队列 2 提交给 CPU，used ring index 更新为 3，表明从队列 3 开始未提交；将队列 3 提交给 CPU，used ring index 更新为 4，表明从队列 4 开始未提交。

因此，当需要更新 used ring 中的多个队列项，或者需要多次更新 used ring index 时，需要多次发起 DMA 请求，导致操作次数较多，DMA 性能下降。

为了解决上述问题，本申请提供了一种聚合提交方案，通过将需要发起的多次 DMA 请求聚合成一次，降低了更新的次数，提高了 DMA 性能。

实施例 1

根据本申请实施例，还提供了一种基于半虚拟化设备的数据处理方法，需要说明的是，在附图的流程图示出的步骤可以在诸如一组计算机可执行指令的计算机系统中执行，并且，虽然在流程图中示出了逻辑顺序，但是在某些情况下，可以以不同于此处的顺序执行所示出或描述的步骤。

本申请实施例所提供的方法实施例可以在移动终端、计算机终端或者类似的运算装置中执行。图 2 示出了一种用于实现基于半虚拟化设备的数据处理方法的计算机终端（或移动设备）的硬件结构框图。如图 2 所示，计算机终端 20（或移动设备）可以包括一个或多个（图中采用 202a、202b，……，202n 来示出）处理器（处理器可以包括但不限于微处

理器 MCU 或可编程逻辑器件 FPGA 等的处理装置）、用于存储数据的存储器 204、以及用于通信功能的传输装置 206。除此以外，还可以包括：显示器、输入/输出接口（I/O 接口）、通用串行总线（USB）端口（可以作为 BUS 总线的端口中的一个端口被包括）、网络接口、电源和/或相机。本领域普通技术人员可以理解，图 2 所示的结构仅为示意，其并不对上述电子装置的结构造成限定。例如，计算机终端 20 还可包括比图 2 中所示更多或者更少的组件，或者具有与图 1 所示不同的配置。

应当注意到的是上述一个或多个处理器 202 和/或其他数据处理电路在本文中通常可以被称为“数据处理电路”。该数据处理电路可以全部或部分的体现为软件、硬件、固件或其他任意组合。此外，数据处理电路可为单个独立的处理模块，或全部或部分的结合到计算机终端 20（或移动设备）中的其他元件中的任意一个内。该数据处理电路作为一种处理器控制（例如与接口连接的可变电阻终端路径的选择）。

存储器 204 可用于存储应用软件的软件程序以及模块，如本申请实施例中的基于半虚拟化设备的数据处理方法对应的程序指令/数据存储装置，处理器通过运行存储在存储器 204 内的软件程序以及模块，从而执行各种功能应用以及数据处理，即实现上述的基于半虚拟化设备的数据处理方法。存储器 204 可包括高速随机存储器，还可包括非易失性存储器，如一个或者多个磁性存储装置、闪存、或者其他非易失性固态存储器。在一些实例中，存储器 204 可进一步包括相对于处理器 202 远程设置的存储器，这些远程存储器可以通过网络连接至计算机终端 20。上述网络的实例包括但不限于互联网、企业内部网、局域网、移动通信网及其组合。

传输装置 206 用于经由一个网络接收或者发送数据。上述的网络具体实例可包括计算机终端 20 的通信供应商提供的无线网络。在一个实例中，传输装置 206 包括一个网络适配器（Network Interface Controller，NIC），其可通过基站与其他网络设备相连从而可与互联网进行通讯。在一个实例中，传输装置 206 可以为射频（Radio Frequency，RF）模块，其用于通过无线方式与互联网进行通讯。

显示器可以例如触摸屏式的液晶显示器（LCD），该液晶显示器可使得用户能够与计算机终端 20（或移动设备）的用户界面进行交互。

此处需要说明的是，在一些可选实施例中，上述图 2 所示的计算机设备（或移动设备）可以包括硬件元件（包括电路）、软件元件（包括存储在计算机可读介质上的计算机代码）、或硬件元件和软件元件两者的结合。应当指出的是，图 2 仅为特定具体实例的一个实例，并且旨在示出可存在于上述计算机设备（或移动设备）中的部件的类型。

在上述运行环境下，本申请提供了如图 3 所示的基于半虚拟化设备的数据处理方法。图 3 是根据本申请实施例的一种基于半虚拟化设备的数据处理方法的流程图。如图 3 所示，该方法包括如下步骤：

步骤 S302，获取半虚拟化设备的完成队列中存储的多个初始数据，其中，多个初始数据用于表征半虚拟化设备已经处理完成，但未提交至主机的原始数据的描述信息。

上述步骤中的半虚拟化设备可以安装在主机上，例如，可以是半虚拟化网卡，也可以是半虚拟化存储器，但不仅限于此。上述步骤中的初始数据可以是存储在 used ring 中未提交给 CPU 的数据项，每个数据项中存储有相应原始数据的描述信息，该描述信息可以包括存储有原始数据的数据 buffer 的地址、原始数据的长度等，但不仅限于此。针对不同类型的半虚拟化设备，原始数据的类型不同，例如，对于半虚拟化网卡，原始数据可以是原始报文。

例如，以如图 4 所示的软硬结合的 virtio 设备虚拟化实现架构为例进行说明，设备接收到原始数据之后，首先执行写 buffer 步骤，将原始数据写入不同的 buffer 中，等待主机对原始数据进行处理，在 virtio 设备处理完毕，也即，将原始数据写入不同的 buffer 之后，可以将相应的描述信息存储至 used ring 中，例如，buffer 0 至 buffer 2 中存储的原始数据已经写入相应的 buffer，可以将相应的描述信息存储至 used ring 中，分别对应队列项 0 至队列项 2。然后执行写 used ring 步骤，可以将队列项 0 至队列项 2 中对应的描述信息作为初始数据。

步骤 S304，确定多个初始数据中满足预设条件的多个第一数据。

上述步骤中的预设条件可以是预先实际需要所设定的聚合条件，例如，该条件可以是直接将 used ring 中所有未提交的队列项；该条件也可以是属于同一个网卡传输队列对应的未提交的队列项，但不仅限于此。

例如，仍以如图 4 的软硬结合的 virtio 设备虚拟化实现架构为例进行说明，假设预设条件为将 used ring 中所有未提交的队列项，因此，可以将队列项 0 至队列项 2 作为第一数据。

步骤 S306，对多个第一数据进行聚合操作，生成第一聚合结果。

在一种可选的实施例中，上述的聚合操作可以是将多个第一数据对应的描述信息进行拼接，例如，将地址进行拼接，将长度进行求和，从而得到上述的第一聚合结果，但不仅限于此，也可以采用其他聚合操作。

步骤 S308，发送携带有第一聚合结果的直接存储器访问请求至主机的内存。

在一种可选的实施例中，对于第一聚合结果，为了能够一次性更新 used ring 对应的内存，可以按照 DMA 协议对第一聚合结果进行封装，得到 DMA 请求，并发送 DMA 请求至主机的内存，完成更新 used ring 的目的。

例如，如图 5 所示的 used ring 为例进行说明，used ring 中包含的 8 个队列项均未提交至内存，此时，队列项 0 已经完成提交，可以将队列项 1 至队列项 7 作为初始数据，通过预设条件可以筛选出队列项 1 至队列项 3 作为第一数据，并将三个队列项进行聚合，生成一个 DMA 请求，一次性完成更新 used ring 的第 1 至 3 项，并将队列项 1 至队列项 3 对应的方框更改为实心方框，表明队列项 1 至队列项 3 已完成提交。

根据本申请上述实施例提供的方案，当需要更新 used ring 时，可以获取 used 中存储的多个初始数据，然后从多个初始数据中筛选出符合预设条件的多个第一数据，对多个

第一数据进行聚合操作生成第一聚合结果，并发送携带有该第一聚合结果的DMA请求至内存，达到一次性更新used ring的多个队列项的目的。容易注意到的是，通过聚合操作的方式发起一次DMA请求，无需针对每个队列项发起一次DMA请求，从而达到了减少更新used ring所产生的操作次数，避免设备侧看到PCIe接口反压，提高DMA性能的技术效果，进而解决了相关技术中半虚拟化设备与主机之前交互频繁，导致DMA性能下降的技术问题。

在本申请上述实施例中，确定多个初始数据中满足预设条件的多个第一数据包括如下至少之一：获取目标缓存中存储的数据，得到多个第一数据，其中，多个初始数据依次缓存至目标缓存；在预设定时时间到达的情况下，确定多个初始数据为多个第一数据；在多个初始数据的数量大于或等于预设数量的情况下，确定多个初始数据为多个第一数据。

上述的目标缓存可以是预先设定的自适应缓存，当PCIe性能出现瓶颈时，数据堆积在自适应缓存中。在一种可选的实施例中，used ring中的队列项在提交给CPU之前，都会依次存储在自适应缓存中，因此，可以将自适应缓存中存储的所有数据项作为第一数据。

上述的预设定时时间可以是预先设置的定时器的时间，可以根据实际需要进行设定。在一种可选的实施例中，在预设定时时间到达之前，无需对used ring中存储的队列项进行任何处理，在预设定时时间到达之后，可以将used ring中存储的所有未提交的队列项作为第一数据。

上述的预设数量可以是预先设置的最大聚合数量，可以根据实际需要进行设定。在一种可选的实施例中，在used ring中存储的未提交的队列项的数量未到达预设数量之前，无需对used ring中存储的队列项进行任何处理，在未提交的队列项的数量到达预设数量之后，可以将used ring中存储的所有未提交的队列项作为第第一数据。

需要说明的是，上述的三个条件可以单独使用，也可以任意组合使用，例如，将目标缓存和预设定时时间进行组合，目标缓存和预设数量进行组合，预设定时时间和预设数量进行组合，以及目标缓存、预设定时时间和预设数量进行组合。具体组合方式可以根据实际需要进行确定，本申请对此不做具体限定。

例如，仍以如图4的软硬结合的virtio设备虚拟化实现架构为例进行说明，预设条件可以包括自适应缓存、定时器和最大聚合数量，可以将上述三个方法综合使用，确定出第一数据，并对第一数据进行聚合操作。

在本申请上述实施例中，对多个第一数据进行聚合操作，生成第一聚合结果包括：确定每个第一数据对应的第一传输队列，其中，第一传输队列用于传输原始数据；获取同一个第一传输队列对应的第一数据，得到第一聚合结果。

由于virtio设备支持多队列，因此，上述的第一传输队列可以是virtio设备支持的队列queue，virtio设备可以使用多个queue传输原始数据给主机，每个queue都对应一个used ring。

在一种可选的实施例中，针对每个队列项，可以确定相应的原始数据发往的queue。由于聚合操作之前，不同queue对应的队列项往往交织在一起，无法直接进行聚合操作，

因此，可以采用按 queue 调度的方式，将相同 queue 对应的队列项排列在一起，进而可以将属于同一个 queue 对应的队列项进行聚合，得到第一聚合结果。

例如，仍以如图 4 的软硬结合的 virtio 设备虚拟化实现架构为例进行说明，如图 6 所示，填充不同图案的方框代表不同 queue 对应的队列项，不同的数字代表 queue 队列中的序号。在调度之前，不同 queue 对应的队列项交织在一起，经过调度，同一个 queue 对应的队列项相邻，因此，可以将同一个 queue 对应的队列项进行聚合，一次性提交给内存。

在本申请上述实施例中，获取同一个第一传输队列对应的第一数据，得到第一聚合结果包括：确定多个第一数据中第一个第一数据对应的第一传输队列，得到目标传输队列；获取多个第一数据中目标传输队列对应的第一数据，得到第一聚合结果。

在一种可选的实施例中，由于 used ring 中存储位置越前的队列项表示半虚拟化设备处理的时间越早，为了降低发送原始数据的设备的等待时间，可以将第一数据中存储位置最前的队列项对应的 queue 作为目标 queue，并将目标 queue 对应的所有第一数据进行聚合操作，得到第一聚合结果，而其他 queue 对应的第一数据需要等待目标 queue 对应的第一数据提交完成之后，再进行处理。

在本申请上述实施例中，该方法还包括：基于完成队列对应的初始队列标识，确定多个初始数据中的多个第二数据，其中，初始队列标识用于表征已经提交完成的数据的标识信息，多个第二数据用于表征当前提交至主机的数据；对多个第二数据进行聚合操作，得到第二聚合结果；基于第二聚合结果对初始队列标识进行更新。

上述的初始队列标识可以是至 used ring index，指向了 used ring 中第一个未提交的队列项。上述的更新可以是对 used ring index 的数值进行更新。

在一种可选的实施例中，在更新 used ring 之后，需要对 used ring index 进行更新，由于更新了 used ring 中的多个队列项，因此，需要对 used ring index 进行多次更新。为了避免多次更新 used ring index 产生的操作次数多，可以确定出最新提交的队列项作为第二数据，并将多次 DMA 请求合并成一次，也即，对第二数据进行聚合操作，得到第二聚合结果，并对 used ring index 进行一次更新。

例如，仍以如图 4 的软硬结合的 virtio 设备虚拟化实现架构为例进行说明，在执行写 used ring 步骤之后，可以执行写 used ring index 步骤，对队列项 0 至队列项 2 进行聚合操作，基于第二聚合结果对 used ring index 进行更新，将取值更新为 3。

例如，如图 5 所示的 used ring 为例进行说明，used ring 中包含的 8 个队列项均未提交至内存，此时，队列项 0 已经完成提交，此时，used ring index 的取值为 1，也即，初始队列标识是 1。并且，一次性完成更新 used ring 的第 1 至 3 项，因此，可以直接将 used ring index 的取值更为 4。

需要说明的是，不同版本 virtio 设备的完成队列的队列结构不同，对于新版本的 virtio 设备，如果不需要对 used ring index 进行更新，则无需执行上述步骤。

在本申请上述实施例中，对多个第二数据进行聚合操作，得到第二聚合结果包括：确

定每个第二数据对应的第二传输队列；获取同一个第二传输队列对应的第二数据，得到第二聚合结果。

上述的第二传输队列也可以是 virtio 设备支持的队列 queue，virtio 设备可以使用多个 queue 传输原始数据给主机，每个 queue 都对应一个 used ring。

在一种可选的实施例中，与 used ring 中队列项的聚合操作类似，可以采用按 queue 调度的方式，将相同 queue 对应的队列项排列在一起，进而可以将属于同一个 queue 对应的队列项进行聚合，得到第二聚合结果。

例如，仍以如图 4 的软硬结合的 virtio 设备虚拟化实现架构为例进行说明，如图 6 所示，填充不同图案的方框代表不同 queue 对应的队列项，在调度之前，不同 queue 对应的队列项交织在一起，经过调度，同一个 queue 对应的队列项相邻，因此，可以将同一个 queue 对应的队列项进行聚合，一次性更新 used ring index。

需要说明的是，按 queue 调度的方式可以只进行一次，也即，如果在更新 used ring 的过程中已经采用按 queue 调度的方式进行聚合操作，则更新 used ring index 的时候无需采用按 queue 调度的方式进行聚合操作；如果在更新 used ring 的过程中未采用按 queue 调度的方式进行聚合操作，则更新 used ring index 的时候采用按 queue 调度的方式进行聚合操作。

在本申请上述实施例中，该方法还包括：获取多个原始数据；确定每个原始数据对应的第三传输队列；按照第三传输队列对多个原始数据进行排序，得到排序后的数据，其中，同一个第三传输队列对应的原始数据相邻；依次将排序后的数据写入主机的数据缓冲区中。

上述的第三传输队列也可以是 virtio 设备支持的队列 queue，virtio 设备可以使用多个 queue 传输原始数据给主机。

在一种可选的实施例中，由于聚合操作之前，不同 queue 对应的原始数据往往交织在一起，无法直接进行聚合操作，因此，可以采用按 queue 调度的方式，将相同 queue 对应的原始数据排列在一起，进而按照排序后的数据依次写入到数据 buffer 中。

例如，仍以如图 4 的软硬结合的 virtio 设备虚拟化实现架构为例进行说明，如图 6 所示，填充不同图案的方框代表不同 queue 对应的队列项，在调度之前，不同 queue 对应的数据交织在一起，经过调度，同一个 queue 对应的数据相邻，因此，可以依次将数据存储至数据 buffer，从而确保 used ring 中存储的队列项中，属于同一个 queue 对应的队列项相邻。

需要说明的是，按 queue 调度的方式可以只进行一次，也即，如果在写数据 buffer 之前，采用按 queue 调度的方式对原始数据进行缓存调度，则后续更新 used ring 和更新 used ring index 的时候无需采用按 queue 调度的方式进行聚合操作；如果在写数据 buffer 之前，未采用按 queue 调度的方式对原始数据进行缓存调度，则在更新 used ring 的过程中如果采用按 queue 调度的方式进行聚合操作，则更新 used ring index 的时候无需采用按 queue 调度的方式进行聚合操作；如果在更新 used ring 的过程中未采用按 queue 调度

的方式进行聚合操作，则更新 used ring index 的时候采用按 queue 调度的方式进行聚合操作。

需要说明的是，对于前述的各方法实施例，为了简单描述，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，本申请并不受所描述的动作顺序的限制，因为依据本申请，某些步骤可以采用其他顺序或者同时进行。其次，本领域技术人员也应该知悉，说明书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是本申请所必须的。

通过以上的实施方式的描述，本领域的技术人员可以清楚地了解到根据上述实施例的方法可借助软件加必需的通用硬件平台的方式来实现，当然也可以通过硬件，但很多情况下前者是更佳的实施方式。基于这样的理解，本申请的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质（如 ROM/RAM、磁碟、光盘）中，包括若干指令用以使得一台终端设备（可以是手机，计算机，服务器，或者网络设备等）执行本申请各个实施例所述的方法。

实施例 2

根据本申请实施例，还提供了一种用于实施上述基于半虚拟化设备的数据处理方法的基于半虚拟化设备的数据处理装置，如图 7 所示，该装置 700 包括：数据获取模块 702、数据确定模块 704、聚合模块 706 和发送模块 708。

其中，数据获取模块 702 用于获取半虚拟化设备的完成队列中存储的多个初始数据，其中，多个初始数据用于表征半虚拟化设备已经处理完成，但未提交至主机的原始数据的描述信息；数据确定模块 704 用于确定多个初始数据中满足预设条件的多个第一数据；聚合模块 706 用于对多个第一数据进行聚合操作，生成第一聚合结果；发送模块 708 用于发送携带有第一聚合结果的直接存储器访问请求至主机的内存。

此处需要说明的是，上述数据获取模块 702、数据确定模块 704、聚合模块 706 和发送模块 708 对应于实施例 1 中的步骤 S302 至步骤 S308，四个模块与对应的步骤所实现的实例和应用场景相同，但不限于上述实施例 1 所公开的内容。需要说明的是，上述模块作为装置的一部分可以运行在实施例 1 提供的计算机终端 10 中。

在本申请上述实施例中，数据确定模块包括如下至少之一：数据获取单元、第一数据确定单元和第二数据确定单元。

其中，数据获取单元用于获取目标缓存中存储的数据，得到多个第一数据，其中，多个初始数据依次缓存至目标缓存；第一确定单元用于在预设定时时间到达的情况下，确定多个初始数据为多个第一数据；第二确定单元用于在多个初始数据的数量大于或等于预设数量的情况下，确定多个初始数据为多个第一数据。

在本申请上述实施例中，聚合模块包括：队列确定单元和结果获取单元。

其中，队列确定单元用于确定每个第一数据对应的第一传输队列，其中，第一传输队列用于传输原始数据；结果获取单元用于获取同一个第一传输队列对应的第一数据，得到

第一聚合结果。

在本申请上述实施例中，结果获取单元还用于确定多个第一数据中第一个第一数据对应的第一传输队列，得到目标传输队列，并获取多个第一数据中目标传输队列对应的第一数据，得到第一聚合结果。

在本申请上述实施例中，该装置还包括：更新模块。

其中，数据确定模块还用于基于完成队列对应的初始队列标识，确定多个初始数据中的多个第二数据，其中，初始队列标识用于表征已经提交完成的数据的标识信息，多个第二数据用于表征当前提交至主机的数据；聚合模块还用于对多个第二数据进行聚合操作，得到第二聚合结果；更新模块用于基于第二聚合结果对初始队列标识进行更新。

在本申请上述实施例中，聚合模块包括：队列确定单元和结果获取单元。

其中，队列确定单元用于确定每个第二数据对应的第二传输队列；结果获取单元用于获取同一个第二传输队列对应的第二数据，得到第二聚合结果。

在本申请上述实施例中，该装置还包括：数据获取模块、队列确定模块、排序模块和写入模块。

其中，数据获取模块用于获取多个原始数据；队列确定模块用于确定每个原始数据对应的第三传输队列；排序模块用于按照第三传输队列对多个原始数据进行排序，得到排序后的数据，其中，同一个第三传输队列对应的原始数据相邻；写入模块用于依次将排序后的数据写入主机的数据缓冲区中。

需要说明的是，本申请上述实施例中涉及到的优选实施方案与实施例 1 提供的方案以及应用场景、实施过程相同，但不仅限于实施例 1 所提供的方案。

实施例 3

根据本申请实施例，还提供了一种用于实施上述基于半虚拟化设备的数据处理方法的基于半虚拟化设备的数据处理系统，如图 8 所示，该系统包括：

主机 82，包括：内存和完成队列。

半虚拟化设备 84，与主机连接，用于获取完成队列中存储的多个初始数据，其中，多个初始数据用于表征半虚拟化设备已经处理完成，但未提交至主机的原始数据的描述信息；确定多个初始数据中满足预设条件的多个第一数据；对多个第一数据进行聚合操作，生成第一聚合结果；发送携带有第一聚合结果的直接存储器访问请求至主机的内存。

在本申请上述实施例中，半虚拟化设备还用于执行如下至少之一步骤：获取目标缓存中存储的数据，得到多个第一数据，其中，多个初始数据依次缓存至目标缓存；在预设定时时间到达的情况下，确定多个初始数据为多个第一数据；在多个初始数据的数量大于或等于预设数量的情况下，确定多个初始数据为多个第一数据。

在本申请上述实施例中，半虚拟化设备还用于确定每个第一数据对应的第一传输队列，并获取同一个第一传输队列对应的第一数据，得到第一聚合结果，其中，第一传输队列用于传输原始数据。

在本申请上述实施例中，半虚拟化设备还用于确定多个第一数据中第一个第一数据对应的第一传输队列，得到目标传输队列，并获取多个第一数据中目标传输队列对应的第一数据，得到第一聚合结果。

在本申请上述实施例中，半虚拟化设备还用于基于完成队列对应的初始队列标识，确定多个初始数据中的多个第二数据，其中，初始队列标识用于表征已经提交完成的数据的标识信息，多个第二数据用于表征当前提交至主机的数据；对多个第二数据进行聚合操作，得到第二聚合结果；基于第二聚合结果对初始队列标识进行更新。

在本申请上述实施例中，半虚拟化设备还用于确定每个第二数据对应的第二传输队列；结果获取单元用于获取同一个第二传输队列对应的第二数据，得到第二聚合结果。

在本申请上述实施例中，主机还包括：数据缓冲区；半虚拟化设备还用于获取多个原始数据；确定每个原始数据对应的第三传输队列；按照第三传输队列对多个原始数据进行排序，得到排序后的数据，其中，同一个第三传输队列对应的原始数据相邻；依次将排序后的数据写入数据缓冲区中。

需要说明的是，本申请上述实施例中涉及到的优选实施方案与实施例 1 提供的方案以及应用场景、实施过程相同，但不仅限于实施例 1 所提供的方案。

实施例 4

本申请的实施例可以提供一种计算机终端，该计算机终端可以是计算机终端群中的任意一个计算机终端设备。可选地，在本实施例中，上述计算机终端也可以替换为移动终端等终端设备。

可选地，在本实施例中，上述计算机终端可以位于计算机网络的多个网络设备中的至少一个网络设备。

在本实施例中，上述计算机终端可以执行基于半虚拟化设备的数据处理方法中以下步骤的程序代码：获取半虚拟化设备的完成队列中存储的多个初始数据，其中，多个初始数据用于表征半虚拟化设备已经处理完成，但未提交至主机的原始数据的描述信息；确定多个初始数据中满足预设条件的多个第一数据；对多个第一数据进行聚合操作，生成第一聚合结果；发送携带有第一聚合结果的直接存储器访问请求至主机的内存。

可选地，图 9 是根据本申请实施例的一种计算机终端的结构框图。如图 9 所示，该计算机终端 A 可以包括：一个或多个（图中仅示出一个）处理器 902、以及存储器 904。

其中，存储器可用于存储软件程序以及模块，如本申请实施例中的基于半虚拟化设备的数据处理方法和装置对应的程序指令/模块，处理器通过运行存储在存储器内的软件程序以及模块，从而执行各种功能应用以及数据处理，即实现上述的基于半虚拟化设备的数据处理方法。存储器可包括高速随机存储器，还可以包括非易失性存储器，如一个或者多个磁性存储装置、闪存、或者其他非易失性固态存储器。在一些实例中，存储器可进一步包括相对于处理器远程设置的存储器，这些远程存储器可以通过网络连接至终端 A。上述网络的实例包括但不限于互联网、企业内部网、局域网、移动通信网及其组合。

处理器可以通过传输装置调用存储器存储的信息及应用程序，以执行下述步骤：获取半虚拟化设备的完成队列中存储的多个初始数据，其中，多个初始数据用于表征半虚拟化设备已经处理完成，但未提交至主机的原始数据的描述信息；确定多个初始数据中满足预设条件的多个第一数据；对多个第一数据进行聚合操作，生成第一聚合结果；发送携带有第一聚合结果的直接存储器访问请求至主机的内存。

可选的，上述处理器还可以执行如下步骤的程序代码：获取目标缓存中存储的数据，得到多个第一数据，其中，多个初始数据依次缓存至目标缓存；和/或，在预设定时时间到达的情况下，确定多个初始数据为多个第一数据；和/或，在多个初始数据的数量大于或等于预设数量的情况下，确定多个初始数据为多个第一数据。

可选的，上述处理器还可以执行如下步骤的程序代码：确定每个第一数据对应的第一传输队列，其中，第一传输队列用于传输原始数据；获取同一个第一传输队列对应的第一数据，得到第一聚合结果。

可选的，上述处理器还可以执行如下步骤的程序代码：确定多个第一数据中第一个第一数据对应的第一传输队列，得到目标传输队列；获取多个第一数据中目标传输队列对应的第一数据，得到第一聚合结果。

可选的，上述处理器还可以执行如下步骤的程序代码：基于完成队列对应的初始队列标识，确定多个初始数据中的多个第二数据，其中，初始队列标识用于表征已经提交完成的数据的标识信息，多个第二数据用于表征当前提交至主机的数据；对多个第二数据进行聚合操作，得到第二聚合结果；基于第二聚合结果对初始队列标识进行更新。

可选的，上述处理器还可以执行如下步骤的程序代码：确定每个第二数据对应的第二传输队列；获取同一个第二传输队列对应的第二数据，得到第二聚合结果。

可选的，上述处理器还可以执行如下步骤的程序代码：获取多个原始数据；确定每个原始数据对应的第三传输队列；按照第三传输队列对多个原始数据进行排序，得到排序后的数据，其中，同一个第三传输队列对应的原始数据相邻；依次将排序后的数据写入主机的数据缓冲区中。

采用本申请实施例，提供了一种基于半虚拟化设备的数据处理的方案。通过聚合操作的方式发起一次 DMA 请求，无需针对每个队列项发起一次 DMA 请求，从而达到了减少更新 used ring 所产生的操作次数，避免设备侧看到 PCIe 接口反压，提高 DMA 性能的技术效果，进而解决了相关技术中半虚拟化设备与主机之前交互频繁，导致 DMA 性能下降的技术问题。

本领域普通技术人员可以理解，图 9 所示的结构仅为示意，计算机终端也可以是智能手机（如 Android 手机、iOS 手机等）、平板电脑、掌上电脑以及移动互联网设备（Mobile Internet Devices，MID）、PAD 等终端设备。图 9 其并不对上述电子装置的结构造成限定。例如，计算机终端 A 还可包括比图 9 中所示更多或者更少的组件（如网络接口、显示装置等），或者具有与图 9 所示不同的配置。

本领域普通技术人员可以理解上述实施例的各种方法中的全部或部分步骤是可以通

过程序来指令终端设备相关的硬件来完成，该程序可以存储于一计算机可读存储介质中，存储介质可以包括：闪存盘、只读存储器（Read-Only Memory，ROM）、随机存取器（Random Access Memory，RAM）、磁盘或光盘等。

实施例 5

本申请的实施例还提供了一种存储介质。可选地，在本实施例中，上述存储介质可以用于保存上述实施例所提供的基于半虚拟化设备的数据处理方法所执行的程序代码。

可选地，在本实施例中，上述存储介质可以位于计算机网络中计算机终端群中的任意一个计算机终端中，或者位于移动终端群中的任意一个移动终端中。

可选地，在本实施例中，存储介质被设置为存储用于执行以下步骤的程序代码：获取半虚拟化设备的完成队列中存储的多个初始数据，其中，多个初始数据用于表征半虚拟化设备已经处理完成，但未提交至主机的原始数据的描述信息；确定多个初始数据中满足预设条件的多个第一数据；对多个第一数据进行聚合操作，生成第一聚合结果；发送携带有第一聚合结果的直接存储器访问请求至主机的内存。

可选的，上述存储介质还被设置为存储用于执行以下步骤的程序代码：获取目标缓存中存储的数据，得到多个第一数据，其中，多个初始数据依次缓存至目标缓存；和/或，在预设定时时间到达的情况下，确定多个初始数据为多个第一数据；和/或，在多个初始数据的数量大于或等于预设数量的情况下，确定多个初始数据为多个第一数据。

可选的，上述存储介质还被设置为存储用于执行以下步骤的程序代码：确定每个第一数据对应的第一传输队列，其中，第一传输队列用于传输原始数据；获取同一个第一传输队列对应的第一数据，得到第一聚合结果。

可选的，上述存储介质还被设置为存储用于执行以下步骤的程序代码：确定多个第一数据中第一个第一数据对应的第一传输队列，得到目标传输队列；获取多个第一数据中目标传输队列对应的第一数据，得到第一聚合结果。

可选的，上述存储介质还被设置为存储用于执行以下步骤的程序代码：基于完成队列对应的初始队列标识，确定多个初始数据中的多个第二数据，其中，初始队列标识用于表征已经提交完成的数据的标识信息，多个第二数据用于表征当前提交至主机的数据；对多个第二数据进行聚合操作，得到第二聚合结果；基于第二聚合结果对初始队列标识进行更新。

可选的，上述存储介质还被设置为存储用于执行以下步骤的程序代码：确定每个第二数据对应的第二传输队列；获取同一个第二传输队列对应的第二数据，得到第二聚合结果。

可选的，上述存储介质还被设置为存储用于执行以下步骤的程序代码：获取多个原始数据；确定每个原始数据对应的第三传输队列；按照第三传输队列对多个原始数据进行排序，得到排序后的数据，其中，同一个第三传输队列对应的原始数据相邻；依次将排序后的数据写入主机的数据缓冲区中。

上述本申请实施例序号仅仅为了描述，不代表实施例的优劣。

在本申请的上述实施例中，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，可以参见其他实施例的相关描述。

在本申请所提供的几个实施例中，应该理解到，所揭露的技术内容，可通过其它的方式实现。其中，以上所描述的装置实施例仅仅是示意性的，例如所述单元的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式，例如多个单元或组件可以结合或者可以集成到另一个系统，或一些特征可以忽略，或不执行。另一点，所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口，单元或模块的间接耦合或通信连接，可以是电性或其它的形式。

所述作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

另外，在本申请各个实施例中的各功能单元可以集成在一个处理单元中，也可以是各个单元单独物理存在，也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现，也可以采用软件功能单元的形式实现。

所述集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用时，可以存储在一个计算机可读取存储介质中。基于这样的理解，本申请的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质中，包括若干指令用以使得一台计算机设备（可为个人计算机、服务器或者网络设备等）执行本申请各个实施例所述方法的全部或部分步骤。而前述的存储介质包括：U 盘、只读存储器（ROM，Read-Only Memory）、随机存取存储器（RAM，Random Access Memory）、移动硬盘、磁碟或者光盘等各种可以存储程序代码的介质。

以上所述仅是本申请的优选实施方式，应当指出，对于本技术领域的普通技术人员来说，在不脱离本申请原理的前提下，还可以做出若干改进和润饰，这些改进和润饰也应视为本申请的保护范围。

权 利 要 求 书

1. 一种基于半虚拟化设备的数据处理方法，其特征在于，包括：

获取所述半虚拟化设备的完成队列中存储的多个初始数据，其中，所述多个初始数据用于表征所述半虚拟化设备已经处理完成，但未提交至主机的原始数据的描述信息；

确定所述多个初始数据中满足预设条件的多个第一数据；

对所述多个第一数据进行聚合操作，生成第一聚合结果；

发送携带有所述第一聚合结果的直接存储器访问请求至所述主机的内存。

2. 根据权利要求 1 所述的方法，其特征在于，确定所述多个初始数据中满足预设条件的多个第一数据包括如下至少之一：

获取目标缓存中存储的数据，得到所述多个第一数据，其中，所述多个初始数据依次缓存至所述目标缓存；

在预设定时时间到达的情况下，确定所述多个初始数据为所述多个第一数据；

在所述多个初始数据的数量大于或等于预设数量的情况下，确定所述多个初始数据为所述多个第一数据。

3. 根据权利要求 1 所述的方法，其特征在于，对所述多个第一数据进行聚合操作，生成第一聚合结果包括：

确定每个第一数据对应的第一传输队列，其中，所述第一传输队列用于传输所述原始数据；

获取同一个第一传输队列对应的第一数据，得到所述第一聚合结果。

4. 根据权利要求 3 所述的方法，其特征在于，获取同一个第一传输队列对应的第一数据，得到所述第一聚合结果包括：

确定所述多个第一数据中第一个第一数据对应的第一传输队列，得到目标传输队列；

获取所述多个第一数据中所述目标传输队列对应的第一数据，得到所述第一聚合结果。

5. 根据权利要求 1 所述的方法，其特征在于，所述方法还包括：

基于所述完成队列对应的初始队列标识，确定所述多个初始数据中的多个第二数据，其中，所述初始队列标识用于表征已经提交完成的数据的标识信息，所述多个第二数据用于表征当前提交至所述主机的数据；

对所述多个第二数据进行聚合操作，得到第二聚合结果；

基于所述第二聚合结果对所述初始队列标识进行更新。

6. 根据权利要求 5 所述的方法，其特征在于，对所述多个第二数据进行聚合操作，得到第二聚合结果包括：

确定每个第二数据对应的第二传输队列；

获取同一个第二传输队列对应的第二数据，得到所述第二聚合结果。

7. 根据权利要求 1 所述的方法，其特征在于，所述方法还包括：

获取多个原始数据；

确定每个原始数据对应的第三传输队列；

按照所述第三传输队列对所述多个原始数据进行排序，得到排序后的数据，其中，同一个第三传输队列对应的原始数据相邻；

依次将所述排序后的数据写入所述主机的数据缓冲区中。

8. 一种基于半虚拟化设备的数据处理装置，其特征在于，包括：

数据获取模块，用于获取所述半虚拟化设备的完成队列中存储的多个初始数据，其中，所述多个初始数据用于表征所述半虚拟化设备已经处理完成，但未提交至主机的原始数据的描述信息；

数据确定模块，用于确定所述多个初始数据中满足预设条件的多个第一数据；

聚合模块，用于对所述多个第一数据进行聚合操作，生成第一聚合结果；

发送模块，用于发送携带有所述第一聚合结果的直接存储器访问请求至所述主机的内存。

9. 一种基于半虚拟化设备的数据处理系统，其特征在于，包括：

主机，包括：内存和完成队列；

所述半虚拟化设备，与所述主机连接，用于获取所述完成队列中存储的多个初始数据，其中，所述多个初始数据用于表征所述半虚拟化设备已经处理完成，但未提交至所述主机的原始数据的描述信息；确定所述多个初始数据中满足预设条件的多个第一数据；对所述多个第一数据进行聚合操作，生成第一聚合结果；发送携带有所述第一聚合结果的直接存储器访问请求至所述主机的内存。

10. 一种计算机可读存储介质，其特征在于，所述计算机可读存储介质包括存储的程序，其中，在所述程序运行时控制所述计算机可读存储介质所在设备执行权利要求 1 至 7 中任意一项所述的基于半虚拟化设备的数据处理方法。

11. 一种计算机终端，其特征在于，包括：存储器和处理器，所述处理器用于运行所述存储器中存储的程序，其中，所述程序运行时执行权利要求 1 至 7 中任意一项所述的基于半虚拟化设备的数据处理方法。