

数据访问方法及计算设备

本申请要求于 2021 年 11 月 30 日提交中国专利局、申请号为 202111441693.4、申请名称为“数据访问方法及计算设备”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

本申请涉及数据存储领域，尤其涉及一种数据访问方法及计算设备。

背景技术

随着数据存储技术的不断创新、完善、加速演进，数据存储的实现机制也越来越繁重。数据存储通常由主机侧中央处理器（central processing unit, CPU）访问主机侧存储器来执行。其执行过程持续占用并消耗处理器资源。可以理解的是，花费高昂成本购买的处理器的工作时间，在用于承担高价值的计算任务时效益更大，使处理器承担数据存储的实现工作显然并不是一个优选的方案。

基于此，现有技术提出增设连接主机侧处理器的数据处理单元（data processing unit, DPU），用于实现访问主机侧存储器来执行原本由主机侧处理器完成的数据存储的工作，这样就可以让处理器将尽可能多的算力投入到其上运行的虚拟机等计算业务中。然而，数据处理单元通常设置在主机外部，需要对硬件进行改进使得数据处理单元能够与主机侧的存储器进行数据交换。现有技术提出通过硬件线缆连接的方式，将主机侧的硬件存储器连接到数据处理单元，使得数据处理单元可以访问主机的硬件存储器。但设置硬件线缆会增加数据处理单元的硬件设计复杂度。

发明内容

有鉴于此，提出了一种数据访问方法及计算设备，本申请实施例的数据访问方法由计算设备的数据处理单元执行，根据本申请实施例的数据访问方法，在保证数据处理单元可以直接访问主机侧的硬件存储器的同时，简化硬件连接方式，降低硬件设计的复杂度。

第一方面，本申请的实施例提供了一种数据访问方法，所述方法由连接处理器的数据处理单元来执行，所述方法包括：将访问请求写入所述处理器的第一缓存队列，其中，所述访问请求包括待访问的存储器的标识，其中，所述存储器连接所述处理器且不直接连接所述数据处理单元；通过高速外围组件互联（PCIe）点对点传输方式向所述存储器的标识指向的所述存储器发送第一指令，其中，所述第一指令包括所述访问请求在所述第一缓存队列中的位置信息，所述第一指令用于指示所述存储器从所述第一缓存队列获取所述访问请求并执行所述访问请求。

根据本申请实施例的数据访问方法，先通过数据处理单元将包括待访问的存储器的标识的访问请求写入处理器的第一缓存队列，使得访问请求可以存储在处理器中的第一缓存队列中；再通过 PCIe 点对点传输方式向存储器的标识指向的存储器发送第一指令，使得待访问的存储器能够接收到第一指令；第一指令包括访问请求在第一缓存队列中的位置信息，第一指

令用于指示存储器从第一缓存队列获取访问请求并执行访问请求，使得存储器接收到第一指令后，能够按照第一指令的指示，从第一缓存队列中的相应位置处获取访问请求，并执行访问请求。通过这种方式，在存储器连接处理器且不直接连接数据处理单元时，数据处理单元无需通过主机的处理器对存储器进行访问，也能够实现对与处理器连接的存储器的直接访问，从而不会占用主机的资源。并且，这种访问方式可以简化硬件连接方式，降低硬件设计的复杂度。

根据第一方面，在所述数据访问方法的第一种可能的实现方式中，所述访问请求还包括所述访问请求的类型以及访问内存地址，所述访问内存地址为所述处理器为所述数据处理单元的内存分配的地址，所述访问请求的类型用于指示所述访问请求为读请求或写请求，所述访问内存地址为所述存储器可访问的地址。

通过这种方式，使得存储器从第一缓存队列获取访问请求时，可以确定访问请求的类型以及访问对象的地址信息，从而能根据访问请求的类型确定执行的是读取操作还是写入操作，并在执行访问请求时，根据访问对象的地址信息，完成对应操作类型下数据处理单元对于存储器的数据访问。

根据第一方面或第一方面的第一种可能的实现方式，在所述数据访问方法的第二种可能的实现方式中，所述方法还包括：从所述处理器的第二缓存队列中获取所述访问请求的执行结果。

由于数据处理单元连接处理器，使得数据处理单元能够从处理器中的第二缓存队列处获取访问请求的执行结果，从而确定访问请求的执行情况。数据处理单元不需对访问请求的执行结果进行缓存，可以降低对数据处理单元的存储空间占用。

根据第一方面或第一方面的第一种可能的实现方式，在所述数据访问方法的第三种可能的实现方式中，所述方法还包括：接收所述存储器通过 PCIE 点对点传输方式传输的所述访问请求的执行结果。

通过这种方式，使得访问请求的执行结果不需由存储器经过处理器传输给数据处理单元，而是由存储器直接传输给数据处理单元，可以提升访问请求的执行结果的传输效率。

根据第一方面的第一种至第三种可能的实现方式中的任意一种可能的实现方式，在所述数据访问方法的第四种可能的实现方式中，所述方法还包括：将数据处理单元的内存的地址传输至所述处理器；接收来自所述处理器的所述访问内存地址。

通过这种方式，使得数据处理单元可以获取到存储器可以访问的访问内存地址，从而能使用该访问内存地址作为数据处理单元通过 PCIE 点对点传输方式进行数据传输的地址，使得数据处理单元和存储器在不直接连接的情况下传输数据成为可能。

根据第一方面，以及以上第一方面的任意一种可能的实现方式，在所述数据访问方法的第五种可能的实现方式中，所述方法还包括：接收所述处理器发送的第一缓存队列和第二缓存队列的地址，所述第一缓存队列的地址用于将访问请求写入所述处理器的第一缓存队列，所述第二缓存队列的地址用于从所述第二缓存队列中获取所述访问请求的执行结果。

通过这种方式，使得数据处理单元能够根据接收到的第一缓存队列的地址找到第一缓存队列，数据处理单元将访问请求写入第一缓存队列时，可以直接将访问请求写入；使得数据处理单元能够根据接收到的第二缓存队列的地址找到第二缓存队列，数据处理单元从第二缓存队列获取访问请求的执行结果时，能够直接获取访问请求的执行结果。从而保证写入访问

请求和获取访问请求结果的实时性和准确度。

第二方面，本申请的实施例提供了一种数据访问方法，所述方法应用于计算设备，所述计算设备包括处理器、连接所述处理器的存储器、以及连接所述处理器且与所述存储器不直接连接的数据处理单元，所述方法包括：所述数据处理单元将访问请求写入所述处理器的第一缓存队列，所述访问请求包括待访问的所述存储器的标识；所述数据处理单元通过高速外围组件互联(PCIe)点对点传输方式向所述存储器的标识指向的所述存储器发送第一指令，所述第一指令指示所述访问请求在所述第一缓存队列中的位置；所述存储器根据所述第一指令从所述第一缓存队列中获取所述访问请求并执行所述访问请求。

根据本申请实施例的数据访问方法，先使用数据处理单元将包括待访问的存储器的标识的访问请求写入处理器的第一缓存队列，使得访问请求可以存储在处理器中的第一缓存队列中；再使用数据处理单元通过 PCIe 点对点传输方式向存储器的标识指向的存储器发送第一指令，使得待访问的存储器能够接收到第一指令；第一指令包括访问请求在第一缓存队列中的位置信息，所述第一指令用于指示存储器从第一缓存队列获取访问请求并执行访问请求，使得存储器接收到第一指令后，能够按照第一指令的指示，从第一缓存队列中的相应位置处获取访问请求，并执行访问请求。通过这种方式，使得计算设备中，在存储器连接处理器且不直接连接数据处理单元时，计算设备无需通过主机的处理器对存储器进行访问，也能够实现对与处理器连接的存储器的直接访问，从而不会占用主机的资源。并且，这种访问方式可以简化硬件连接方式，降低硬件设计的复杂度。

根据第二方面，在所述数据访问方法的第一种可能的实现方式中，所述访问请求还包括访问内存地址，所述访问内存地址为所述处理器为所述数据处理单元的内存分配的地址，所述访问内存地址为所述存储器可访问的地址；所述执行所述访问请求包括：所述存储器根据所述访问内存地址，通过 PCIe 点对点传输方式将数据写入所述数据处理单元的内存或从所述数据处理单元的内存中读取数据。

通过这种方式，使得存储器可以根据访问内存地址找到访问对象也即数据处理单元的内存，从而能执行访问请求写入数据到数据处理单元的内存，或者，从数据处理单元的内存读取数据，完成数据处理单元的内存对存储器的访问。

根据第二方面的第一种可能的实现方式，在所述数据访问方法的第二种可能的实现方式中，所述访问请求还包括所述访问请求的类型，所述访问请求的类型用于指示所述访问请求为读请求或写请求；当所述访问请求为读请求时，所述执行所述访问请求包括：根据所述访问请求从所述存储器中读取数据，并根据所述访问内存地址，将所述存储器中读取的数据通过 PCIe 点对点传输方式写入所述数据处理单元的内存；当所述访问请求为写请求时，所述执行所述访问请求包括：根据所述访问内存地址，通过 PCIe 点对点传输方式从所述数据处理单元的内存中获取待写入所述存储器的数据，并将获取的数据写入所述存储器。

访问请求为读请求时，存储器能够写入数据到数据处理单元的内存，相当于数据处理单元的内存读存储器；访问请求为写请求时，存储器能够从数据处理单元的内存读取数据，相当于数据处理单元的内存写存储器。通过这种方式，使得对应于访问请求的不同类型，存储器能够以符合访问请求的需求的不同的操作进行响应。

根据第二方面的第一种或第二种可能的实现方式，在所述数据访问方法的第三种可能的实现方式中，所述方法还包括：所述数据处理单元向所述处理器发送所述数据处理单元的内

存的地址；所述处理器根据所述数据处理单元的内存的地址，为所述数据处理单元的内存分配所述访问内存地址；所述处理器向所述数据处理单元发送所述访问内存地址。

通过这种方式，使得处理器可以为数据处理单元的内存分配访问内存地址，从而能使用该访问内存地址作为数据处理单元通过 PCIE 点对点传输方式进行数据传输的地址，使得数据处理单元和存储器在不直接连接的情况下传输数据成为可能。为数据处理单元的内存分配访问内存地址并发送访问内存地址可以在配置阶段进行，使得数据处理单元中的软件产生访问请求之后，直接使用该访问内存地址即可，可以提升计算设备执行访问请求的效率。

根据第二方面，以及以上第二方面的任意一种可能的实现方式，在所述数据访问方法的第四种可能的实现方式中，所述方法还包括：所述存储器将所述访问请求的执行结果写入所述处理器的第二缓存队列；所述数据处理单元从所述处理器的第二缓存队列中获取所述访问请求的执行结果。

根据第二方面，以及第二方面的第一种至第三种可能的实现方式中的任意一种可能的实现方式，在所述数据访问方法的第五种可能的实现方式中，所述方法还包括：所述数据处理单元接收所述存储器通过 PCIE 点对点传输方式传输的所述访问请求的执行结果。

根据第二方面，以及以上第二方面的任意一种可能的实现方式，在所述数据访问方法的第六种可能的实现方式中，所述方法还包括：所述数据处理单元接收所述处理器发送的第一缓存队列和第二缓存队列的地址，所述第一缓存队列的地址用于将访问请求写入所述处理器的第一缓存队列，所述第二缓存队列的地址用于从所述第二缓存队列中获取所述访问请求的执行结果。

通过这种方式，使得数据处理单元能够根据接收到的第一缓存队列的地址找到第一缓存队列，以使得数据处理单元可以准确、迅速地执行将访问请求写入第一缓存队列、从第二缓存队列获取访问请求的执行结果的操作。第一缓存队列和第二缓存队列的创建，以及第一缓存队列和第二缓存队列的地址的发送可以在配置阶段进行，使得数据处理单元中的软件产生访问请求之后，直接使用第一缓存队列和第二缓存队列的地址即可，可以提升计算设备执行访问请求的效率。

第三方面，本申请的实施例提供了一种计算设备，包括：处理器；存储器，连接所述处理器；以及数据处理单元，连接所述处理器且与所述存储器不直接连接，所述数据处理单元用于执行上述第一方面或者第一方面的多种可能的实现方式中的一种或几种的数据访问方法。

第四方面，本申请的实施例提供了一种非易失性计算机可读存储介质，其上存储有计算机程序指令，所述计算机程序指令被处理器执行时实现上述第一方面或者第一方面的多种可能的实现方式中的一种或几种的数据访问方法。

第五方面，本申请的实施例提供了一种计算机程序产品，包括计算机可读代码，或者承载有计算机可读代码的非易失性计算机可读存储介质，当所述计算机可读代码在电子设备中运行时，所述电子设备中的处理器执行上述第一方面或者第一方面的多种可能的实现方式中的一种或几种的数据访问方法。

本申请的这些和其他方面在以下（多个）实施例的描述中会更加简明易懂。

附图说明

包含在说明书中并且构成说明书的一部分的附图与说明书一起示出了本申请的示例性实

施例、特征和方面，并且用于解释本申请的原理。

图 1 示出现有技术中计算设备的数据存储功能的一种实现方式。

图 2 示出现有技术中计算设备的数据存储功能的另一种实现方式。

图 3 示出根据本申请实施例的计算设备的示例性应用场景。

图 4 示出根据本申请实施例的计算设备的一种示例性工作流程。

图 5 示出根据本申请实施例的数据处理单元将数据处理单元的内存的地址发送至处理器的一个示例。

图 6 示出根据本申请实施例的处理器得到访问内存地址并发送访问内存地址的一个示例。

图 7 示出根据本申请实施例的处理器创建缓存队列并发送缓存队列地址的一个示例。

图 8 示出根据本申请实施例的数据处理单元将访问请求写入处理器的第一缓存队列的一个示例。

图 9 示出根据本申请实施例的数据处理单元产生第一指令并向存储器发送第一指令的一个示例。

图 10 示出根据本申请实施例的存储器获取访问请求并执行访问请求的一个示例。

图 11 示出根据本申请实施例的存储器将访问请求的执行结果写入第二缓存队列的一个示例。

图 12 示出根据本申请实施例的数据处理单元获取访问请求的执行结果的一个示例。

图 13 示出根据本申请实施例的计算设备的另一种示例性工作流程。

图 14 示出根据本申请实施例的存储器将访问请求的执行结果写入数据处理单元的一个示例。

图 15 示出根据本申请实施例的数据访问方法的示例性工作流程的示意图。

图 16 示出根据本申请实施例的数据访问方法的示例性工作流程的示意图。

图 17 示出根据本申请实施例的计算设备的示例性结构示意图。

具体实施方式

以下将参考附图详细说明本申请的各种示例性实施例、特征和方面。附图中相同的附图标记表示功能相同或相似的元件。尽管在附图中示出了实施例的各种方面，但是除非特别指出，不必按比例绘制附图。

在这里专用的词“示例性”意为“用作例子、实施例或说明性”。这里作为“示例性”所说明的任何实施例不必解释为优于或好于其它实施例。

另外，为了更好的说明本申请，在下文的具体实施方式中给出了众多的具体细节。本领域技术人员应当理解，没有某些具体细节，本申请同样可以实施。在一些实例中，对于本领域技术人员熟知的方法、手段、元件和电路未作详细描述，以便于凸显本申请的主旨。

图 1 示出现有技术中计算设备的数据存储功能的一种实现方式。图 1 所示的计算设备是以超融合基础架构下的计算设备为例。超融合基础架构(hyper converged infrastructure, HCI)是一种将虚拟化计算和存储整合到同一个系统平台的技术，可以将存储和计算功能集成到一个单一计算设备(或计算设备集群，每个计算设备都提供计算和存储功能)上。

如图 1 所示，对于超融合基础架构下的某一计算设备，其主机侧包括处理器(例如中央

处理器)、SAS 扩展器、SAS 磁盘阵列芯片以及存储器,存储器可例如包括串行连接 SCSI 接口标准硬盘驱动器/固态驱动器(serial attached SCSI hard disk drive/ solid state drive, SAS HDD/SSD)和/或非易失性内存主机控制器接口标准固态驱动器(non-volatile memory express, NVMe SSD)等。其中,处理器和存储器 NVMe SSD 通过高速外围组件互联(peripheral component interconnect express, PCIE)线缆连接,处理器还通过 PCIE 线缆连接 SAS 磁盘阵列芯片, SAS 磁盘阵列芯片还通过 SAS 扩展器连接存储器 HDD/SSD。

处理器运行存储虚拟机以及多个用户虚拟机,其中存储虚拟机还运行分布式存储软件,每个用户虚拟机还运行各自的应用软件,分布式存储软件用于发出针对主机侧存储器的访问请求,处理器根据访问请求可以通过 PCIE 方式直接访问主机侧存储器,以为用户虚拟机中的应用软件提供存储服务。

这种方案的缺点是访问存储器实现数据存储功能由处理器来完成,会大量占用处理器资源,导致处理器对于用户虚拟机的支持能力下降。

为释放主机侧处理器的处理资源和内存,提升用户虚拟机的密度,节省系统授权费用,现有技术如图 1 的方案基础上提出了一种改进技术方案,图 2 示出现有技术中计算设备的数据存储功能的另一种实现方式。

如图 2 所示,数据处理单元通过 PCIE 线缆连接主机侧处理器(例如中央处理器),使用数据处理单元来代替主机侧处理器运行分布式存储软件,以向主机中的用户虚拟机中的应用软件提供数据存储服务,这样就可以让尽可能多的处理器计算能力来运行用户虚拟机的应用软件。在此情况下,需要解决数据处理单元如何访问主机侧存储器的问题。

现有技术提出通过硬件线缆连接的方式,将数据处理单元连接到主机侧存储器,使得数据处理单元能够访问主机侧存储器。其中,数据处理单元与存储器 NVMe SSD 可以通过 PCIE 线缆连接,数据处理单元与存储器 HDD/SSD 可以通过 SAS 线缆连接。然而,这种方案对于数据处理单元的硬件设施有如下的要求:

首先, SAS 磁盘阵列芯片通常设计为插接在处理器上,因此,数据处理单元需要集成一个新的 SAS 磁盘阵列芯片来连接 SAS 扩展器。其次,要求数据处理单元具备足够的 PCIE 端口,或者需要集成 PCIE Switch 芯片(未示出),以通过 PCIE 端口或 PCIE Switch 芯片连接多个存储器 NVMe SSD。

线缆和/或芯片的增加为数据处理单元带来额外的功耗、散热、空间布局、硬件成本等问题,使得数据处理单元的硬件连接方式和硬件设计复杂度大大增加。

有鉴于此,提出了一种数据访问方法及计算设备,本申请实施例的数据访问方法由计算设备的数据处理单元执行,根据本申请实施例的数据访问方法,在保证数据处理单元可以直接访问主机侧的硬件存储器的同时,简化硬件连接方式,降低硬件设计的复杂度。

图 3 示出根据本申请实施例的计算设备的示例性应用场景。

如图 3 所示,本申请实施例的计算设备主机侧包括处理器(例如中央处理器)以及多个存储器(可包括存储器 NVMe SSD 以及存储器 HDD/SSD),数据处理单元通过 PCIE 线缆连接处理器,且不连接各存储器。其中,数据处理单元不设置在计算设备的主机侧。

在主机中,处理器可以运行至少一个用户虚拟机,每个用户虚拟机还分别运行各自的应用软件。应用软件具有数据存储需求。处理器和存储器 NVMe SSD 通过 PCIE 线缆连接,处理器还通过 PCIE 线缆连接 SAS 磁盘阵列芯片, SAS 磁盘阵列芯片还通过 SAS 扩展器连接存储器

HDD/SSD。

数据处理单元可以运行分布式存储软件，在分布式存储软件产生访问请求时，或者数据处理单元运行的其他软件（未示出）产生访问请求时，数据处理单元执行本申请实施例的数据访问方法，通过 PCIE 点对点传输方式访问主机侧的存储器，以向主机侧处理器的用户虚拟机中的应用软件提供数据存储服务。

图 3 中的计算设备可例如应用于超融合基础架构中，也可以应用于其他数据处理单元连接主机侧处理器且不连接主机侧存储器时、数据处理单元访问主机侧存储器的应用场景，本申请实施例对计算设备的具体应用场景不作限制。

图 4 示出根据本申请实施例的计算设备的一种示例性工作流程。下面结合图 4 对本申请实施例实现数据处理单元访问主机侧的硬件存储器的示例性方法进行描述。

本申请实施例的计算设备在被用户使用之前可以先进行配置，使得计算设备中，数据处理单元可以直接响应访问请求，而不必在产生访问请求时再配置计算设备中的各装置或模块来为响应访问请求做准备。为了尽可能提升用户体验，配置可例如在计算设备出厂前完成。用户使用计算设备时，可以对处理器运行的应用软件进行各种操作。一些操作可能涉及到数据的读写，此时应用软件可以产生相应的存储需求。根据应用软件的存储需求，或者根据涉及到数据的读写的操作，处理器可以指示其连接的数据处理单元运行的分布式存储软件或其他软件（未示出）产生访问请求。此时数据处理单元可以将访问请求写入处理器中的缓存队列，使得存储器能够从缓存队列获取并执行访问请求，以达到访问主机侧的硬件存储器的目的。

如图 4 所示，根据本申请实施例的计算设备中，在分布式存储软件产生访问请求之前，计算设备可以执行步骤 S1-S3，对数据处理单元、处理器以及存储器进行配置。在数据处理单元、处理器以及存储器配置好后，在分布式存储软件产生访问请求时，计算设备可以执行步骤 S4-S8，实现数据处理单元访问主机侧的硬件存储器。

下面结合图 4 先介绍对数据处理单元、处理器以及存储器进行配置的示例性方法。

步骤 S1，数据处理单元将数据处理单元的内存的地址发送至处理器。图 5 示出根据本申请实施例的数据处理单元将数据处理单元的内存的地址发送至处理器的一个示例。

举例来说，如图 5 所示，数据处理单元包括内存，用于存储数据处理单元在执行任务或指令时所使用的或者生成的数据。数据处理单元的内存中可以包括数据页面区，用于存储数据处理单元可以写出到计算设备中的其他装置或模块（例如存储器）的数据，以及由计算设备中的其他装置或模块（例如存储器）写入的数据。数据处理单元的内存的地址，可以是数据页面区的地址。存储器设置在主机侧，因此，对于存储器来说，能够通过 PCIE 点对点传输方式传输数据的对象也应具备主机侧的地址。而数据处理单元不设置在主机侧，因此，数据处理单元的内存的地址，并不是主机侧的地址，不能够作为存储器通过 PCIE 点对点传输方式传输数据的对象的地址。由于数据处理单元连接主机侧的处理器，在步骤 S1 中，数据处理单元可以先将该数据处理单元的内存的地址发送给主机侧的处理器。由处理器先处理得到对应于数据处理单元的内存的地址的主机侧的地址（其示例参见下文的步骤 S2）。

步骤 S2，处理器根据数据处理单元的内存的地址，为数据处理单元的内存分配访问内存地址，并将该访问内存地址发送给数据处理单元。图 6 示出根据本申请实施例的处理器得到访问内存地址并发送访问内存地址的一个示例。

举例来说，如图 6 所示，处理器设置在计算设备的主机侧，因此，由处理器生成的地址是主机侧的地址，是可以作为存储器通过 PCIE 点对点传输方式传输数据的传输对象的地址使用的。基于此，在接收到数据处理单元的内存的地址之后，处理器可以在其存储空间中，为数据处理单元的内存分配一块对应的存储区域，访问内存地址可以是该存储区域的地址。访问内存地址作为存储器通过 PCIE 点对点传输方式传输数据的对象的地址时，使得与该访问内存地址所属的存储区域对应的数据处理单元的内存，可以作为存储器通过 PCIE 点对点传输方式传输数据的传输对象，从而建立起存储器和数据处理单元之间的 PCIE 点对点传输方式的传输通道。访问内存地址可以在数据处理单元与存储器之间通过 PCIE 点对点传输方式的传输数据时（示例参见下文步骤 S6）使用。

步骤 S3，处理器创建对应于存储器的第一缓存队列、第二缓存队列，并将第一缓存队列、第二缓存队列在处理器中的存储地址分别发送给数据处理单元和存储器。图 7 示出根据本申请实施例的处理器创建缓存队列并发送缓存队列地址的一个示例。

举例来说，如图 7 所示，第一缓存队列可以是用于存储访问请求的提交队列（submission queue, SQ），第二缓存队列可以是存储访问请求的执行结果的完成队列（completion queue, CQ）。第一缓存队列、第二缓存队列可以储存在处理器的内存中，处理器同时连接数据处理单元和存储器，因此可以将第一缓存队列、第二缓存队列在处理器中的存储地址分别传输给数据处理单元和存储器。数据处理单元根据第一缓存队列在处理器中的存储地址可以写入访问请求到第一缓存队列（示例参见下文步骤 S4），根据第二缓存队列在处理器中的存储地址可以从第二缓存队列读取访问请求的执行结果（示例参见下文步骤 S8）。存储器根据第一缓存队列在处理器中的存储地址可以从第一缓存队列读取访问请求（示例参见下文步骤 S6），根据第二缓存队列在处理器中的存储地址可以写入访问请求的执行结果到第二缓存队列（示例参见下文步骤 S7）。通过这种方式，可以实现访问请求、访问请求的执行结果在数据处理单元和存储器之间的传输。

步骤 S3 也可以在步骤 S1 或步骤 S2 之前执行，或者，与步骤 S1 或步骤 S2 同时执行，本申请对步骤 S3 和步骤 S1 的执行顺序以及步骤 S3 和步骤 S2 的执行顺序不作限制。

执行步骤 S1-S3 之后，数据处理单元、处理器以及存储器已经配置完成。下面结合图 4 介绍数据处理单元、处理器以及存储器配置好后，计算设备实现数据处理单元访问主机侧的硬件存储器的示例性方法。

步骤 S4，数据处理单元将访问请求写入处理器的第一缓存队列。

举例来说，在步骤 S3 中，数据处理单元接收到第一缓存队列的存储地址，因此，第一缓存队列在处理器中的位置对于数据处理单元来说是已知的。在此情况下，在数据处理单元中，分布式存储软件或其他软件产生访问请求时，数据处理单元可以根据第一缓存队列的地址，将访问请求写入第一缓存队列中。

图 8 示出根据本申请实施例的数据处理单元将访问请求写入处理器的第一缓存队列的一个示例。如图 8 所示，数据处理单元可以将访问请求先转换为提交队列（第一缓存队列）中的条目的形式（submission queue entry, SQE），转换后的条目可以包括操作码（operation code）、分散聚合表（scatter gather list, SGL）、标识等。第一缓存队列可以包括多个条目，将访问请求写入处理器的第一缓存队列时，可以是将根据访问请求转换后的条目通过直接存储器存取（direct memory access, DMA）方式写入到处理器的第一缓存队列的某条目的

位置（例如队列尾部的条目）。

其中，操作码可以指示访问请求的类型，访问请求可以分为不同类型，例如访问请求可以是数据处理单元读存储器的请求（相当于存储器传输数据到数据处理单元），此时访问请求可以是读请求，存储器执行访问请求时是写入操作，即存储器写入数据到数据处理单元（示例参见下文的步骤 S6）。访问请求还可以是数据处理单元写存储器的请求（相当于数据处理单元传输数据到存储器），此时访问请求可以是写请求，存储器执行访问请求时是读取操作，即存储器读取数据处理单元的内存的数据（示例参见下文的步骤 S6）。通过使用操作码指示访问请求的类型，从而能将访问请求的类型写入处理器的第一缓存队列，以使得存储器能够从第一缓存队列中获取到访问请求的类型，确定执行访问请求是读取操作还是写入操作。

分散聚合表可以存储访问内存地址。由上文步骤 S2 的相关描述可知，访问内存地址是可以在存储器通过 PCIE 点对点传输方式传输数据时作为传输对象的地址使用的。通过使用分散聚合表存储访问内存地址，从而能将访问内存地址写入处理器的第一缓存队列，以使得存储器执行访问请求时，能够根据访问内存地址找到读取操作或写入操作的对象（本示例中是数据处理单元的数据页面区）。

标识可以用于识别待访问的存储器。待访问的存储器可以是主机侧的存储器中的任一存储器。对于主机侧的每一存储器，可以分别对应一个第一缓存队列，在此情况下，每一第一缓存队列也可以包括对应于该第一缓存队列的存储器的标识。将访问请求写入处理器的第一缓存队列时，可以根据访问请求包括的标识，在多个第一缓存队列中，找到具有相同标识的第一缓存队列，再将访问请求写入第一缓存队列。

本领域技术人员应理解，访问请求转换后的条目可以包括更多内容，例如访问请求的长度等等，本申请对于访问请求包括的具体信息不作限制。

步骤 S5，数据处理单元产生包括访问请求在第一缓存队列中的位置信息的第一指令，并通过 PCIE 点对点传输方式向访问请求的标识指向的存储器发送第一指令，第一指令用于指示存储器从第一缓存队列获取访问请求并执行访问请求。

高速外围组件互联点对点传输方式（也即 PCIE 点对点传输方式）是现有技术已有的一种数据传输方式，可以使得两个设备在不直接连接的情况下也能进行数据传输。本申请实施例中通过 PCIE 点对点传输方式使得不直接连接的数据处理单元和待访问的存储器之间能够传输第一指令。

作为第一指令的发送对象的待访问的存储器，可以是步骤 S4 中，访问请求包括的标识对应的存储器。第一指令可以包括访问请求在第一缓存队列中的位置信息，第一指令可以用于指示存储器从存储器对应的一个第一缓存队列获取访问请求并执行访问请求。也即，存储器从哪个队列获取访问请求是由第一缓存队列和存储器的对应关系决定的。在此情况下，待访问的存储器接收到第一指令后，可以根据第一指令的指示，从存储器对应的一个第一缓存队列中的对应位置处获取到访问请求，再执行访问请求。

图 9 示出根据本申请实施例的数据处理单元产生第一指令并向存储器发送第一指令的一个示例。如图 9 所示，可以通过指针（doorbell）记录访问请求在第一缓存队列中的位置信息，得到第一缓存队列的指针（即第一缓存队列的 doorbell 地址），第一指令可以包括第一缓存队列的指针。例如，执行步骤 S4 之前，第一缓存队列的指针可以是“3”，表示前次写入的访问请求写入到第一缓存队列的第 3 个条目处，执行步骤 S4 之后，在第一缓存队列的第 4

个条目处新写入了访问请求，在此情况下，第一缓存队列的指针可以更新为“4”，在步骤 S5 中，处理器可以产生第一指令，第一指令包括的第一缓存队列的指针可以是“4”。处理器可以向待访问的存储器发送包括第一缓存队列的指针“4”的第一指令。

步骤 S6，存储器根据第一指令的指示，从第一缓存队列获取访问请求并执行访问请求。

图 10 示出根据本申请实施例的存储器获取访问请求并执行访问请求的一个示例。

举例来说，如图 10 所示，存储器接收到来自处理器的第一指令，第一指令指示存储器从第一缓存队列获取访问请求并执行访问请求。以第一指令包括的第一缓存队列的指针信息是“4”为例，根据第一指令的指示，存储器从第一缓存队列的第 4 个条目的位置处获取访问请求。结合步骤 S5 的相关描述，获取到的可以是由访问请求转换后的条目。

根据转换后的条目中的操作码，存储器可以确定访问请求的类型。根据转换后的条目中的分散聚合表，存储器可以确定 PCIE 点对点传输方式进行数据传输的对象地址（即访问内存地址）。根据访问请求的类型以及访问内存地址，存储器可以执行访问请求。

存储器执行访问请求的方式，可以根据访问内存地址，通过 PCIE 点对点传输方式将数据写入数据处理单元的内存或从数据处理单元的内存中读取数据。其中，访问请求的类型是读请求时，存储器执行访问请求的方式，可以根据访问请求从存储器中读取数据，并根据访问内存地址，将存储器中读取的数据通过 PCIE 点对点传输方式写入数据处理单元的内存，例如，可以写入数据处理单元的内存的数据页面区。访问请求的类型是写请求时，存储器执行访问请求的方式，可以根据访问内存地址，通过 PCIE 点对点传输方式从数据处理单元的内存中获取待写入存储器的数据，例如，可以从数据处理单元的内存的数据页面区获取数据，并将获取的数据写入存储器。

其中，访问请求还可以包括访问存储器的某一存储区域的信息，例如存储器的某一存储区域的标识。访问请求的类型是读请求时，可以根据访问请求从存储器中读取存储在访问请求包括的存储区域标识对应的存储区域中的数据。访问请求的类型是写请求时，可以将获取的数据写入存储器中访问请求包括的存储区域标识对应的存储区域。

步骤 S7，存储器将访问请求的执行结果写入处理器的第二缓存队列。

访问请求是分布式存储软件等数据处理单元运行的软件产生的，因此，对应于访问请求，可以有一个访问请求的执行结果来传输至产生访问请求的软件，以告知产生访问请求的软件其产生的访问请求的执行情况。访问请求的执行结果可以有两种类型，一种是访问请求已执行完成，一种是访问请求尚未执行完成。

图 11 示出根据本申请实施例的存储器将访问请求的执行结果写入第二缓存队列的一个示例。如图 11 所示，存储器可以将访问请求的执行结果转换为完成队列（第二缓存队列）中的条目（completion queue entry, CQE）的形式，转换后的条目可以包括标识等。标识的不同类型可以对应不同的执行结果，例如标识是“1”可以表示具备该标识的条目是表示访问请求已执行完成的条目，标识是“0”可以表示具备该标识的条目是表示访问请求尚未执行完成的条目。第二缓存队列可以包括多个条目，将访问请求的执行结果写入处理器的第二缓存队列时，可以是将根据访问请求的执行结果转换后的条目通过直接存储器存取（direct memory access, DMA）方式写入到处理器的第二缓存队列的某条目的位置（例如队列尾部的条目）。

对于主机侧的每一存储器，在分别对应一个第一缓存队列时，还可以分别对应一个第二缓存队列，在此情况下，每一第二缓存队列也可以包括对应于该第二缓存队列的存储器的标

识。将访问请求的执行结果写入处理器的第二缓存队列时，可以根据访问请求包括的标识，在多个第二缓存队列中，找到具有相同标识的第二缓存队列，再将访问请求的执行结果写入第二缓存队列。

步骤 S8，数据处理单元从处理器的第二缓存队列获取访问请求的执行结果。

图 12 示出根据本申请实施例的数据处理单元获取访问请求的执行结果的一个示例。

举例来说，步骤 S7 中，存储器将访问请求的执行结果写入处理器中的第二缓存队列，在写入时，是没有通知数据处理单元这一写入操作的。因此，对于数据处理单元来说，处理器的第二缓存队列的更新时间是未知的。为了获取访问请求的执行结果，如图 12 所示，可以设置数据处理单元每间隔预设时间段监测处理器中第二缓存队列是否有更新。在监测到第二缓存队列中出现新写入的访问请求的执行结果时，获取该访问请求的执行结果。数据处理单元可以将执行结果传输给产生访问请求的软件。数据处理单元不需要缓存访问请求的执行结果，可以降低数据处理单元的存储空间占用率。

图 13 示出根据本申请实施例的计算设备的另一种示例性工作流程。如图 13 所示，其中步骤 S1-S6 可以参见上文图 4 中步骤 S1-S6 的相关描述，删去与第二缓存队列相关的部分即可，在此不再赘述。

步骤 S6 完成后，计算设备可以执行步骤 S9：存储器通过 PCIE 点对点传输方式将访问请求的执行结果写入数据处理单元。图 14 示出根据本申请实施例的存储器将访问请求的执行结果写入数据处理单元的一个示例。

举例来说，步骤 S6 中，存储器执行访问请求之后，可以产生一个访问请求的执行结果。如图 14 所示，数据处理单元具备缓存访问请求的执行结果的功能时，访问请求的执行结果可以直接通过 PCIE 点对点传输方式写入数据处理单元。例如，可以在数据处理单元的存储空间中分配一块存储区域，用于缓存访问请求的执行结果。可以设置该区域的存储内容更新时产生中断通知数据处理单元，或者设置数据处理单元轮询该区域监测存储内容是否有更新。其中，中断以及轮询可以基于现有技术来实现。在新写入的访问请求的执行结果使得存储内容更新时，数据处理单元可以找到新写入的访问请求的执行结果，并将该执行结果传输给产生访问请求的软件。访问请求的执行结果不需要通过处理器传输，可以提高数据处理单元获取访问请求的执行结果的效率。

图 15 示出根据本申请实施例的数据访问方法的示例性工作流程的示意图。

如图 15 所示，本申请实施例提出一种数据访问方法，所述方法由连接处理器的数据处理单元来执行，所述方法包括步骤 S10-S11：

步骤 S10，将访问请求写入所述处理器的第一缓存队列，其中，所述访问请求包括待访问的存储器的标识，其中，所述存储器连接所述处理器且不直接连接所述数据处理单元。

步骤 S11，通过高速外围组件互联(PCIE)点对点传输方式向所述存储器的标识指向的所述存储器发送第一指令，其中，所述第一指令包括所述访问请求在所述第一缓存队列中的位置信息，所述第一指令用于指示所述存储器从所述第一缓存队列获取所述访问请求并执行所述访问请求。

其中，步骤 S10 的示例性实现方式，可以参照上文以及图 4、图 13 中步骤 S4 的相关描述。步骤 S11 的示例性实现方式，可以参照上文以及图 4、图 13 中步骤 S5 的相关描述。

根据本申请实施例的数据访问方法，先通过数据处理单元将包括待访问的存储器的标识

的访问请求写入处理器的第一缓存队列，使得访问请求可以存储在处理器中的第一缓存队列中；再通过 PCIE 点对点传输方式向存储器的标识指向的存储器发送第一指令，使得待访问的存储器能够接收到第一指令；第一指令包括访问请求在第一缓存队列中的位置信息，第一指令用于指示存储器从第一缓存队列获取访问请求并执行访问请求，使得存储器接收到第一指令后，能够按照第一指令的指示，从第一缓存队列中的相应位置处获取访问请求，并执行访问请求。通过这种方式，在存储器连接处理器且不直接连接数据处理单元时，数据处理单元无需通过主机的处理器对存储器进行访问，也能够实现对与处理器连接的存储器的直接访问，从而不会占用主机的资源。并且，这种访问方式可以简化硬件连接方式，降低硬件设计的复杂度。

在一种可能的实现方式中，所述访问请求还包括所述访问请求的类型以及访问内存地址，所述访问内存地址为所述处理器为所述数据处理单元的内存分配的地址，所述访问请求的类型用于指示所述访问请求为读请求或写请求，所述访问内存地址为所述存储器可访问的地址。访问请求可以参照上文以及图 8 的相关描述中的访问请求的示例。

通过这种方式，使得存储器从第一缓存队列获取访问请求时，可以确定访问请求的类型以及访问对象的地址信息，从而能根据访问请求的类型确定执行的是读取操作还是写入操作，并在执行访问请求时，根据访问对象的地址信息，完成对应操作类型下数据处理单元对于存储器的数据访问。

在一种可能的实现方式中，所述方法还包括：从所述处理器的第二缓存队列中获取所述访问请求的执行结果。其示例性实现方式，可以参照上文以及图 4 中步骤 S8 的相关描述，以及上文及图 12 的相关描述。

由于数据处理单元连接处理器，使得数据处理单元能够从处理器中的第二缓存队列处获取访问请求的执行结果，从而确定访问请求的执行情况。数据处理单元不需对访问请求的执行结果进行缓存，可以降低对数据处理单元的存储空间占用。

在一种可能的实现方式中，所述方法还包括：接收所述存储器通过 PCIE 点对点传输方式传输的所述访问请求的执行结果。其示例性实现方式，可以参照上文以及图 13 中步骤 S9 的相关描述，以及上文及图 14 的相关描述。

通过这种方式，使得访问请求的执行结果不需由存储器经过处理器传输给数据处理单元，而是由存储器直接传输给数据处理单元，可以提升访问请求的执行结果的传输效率。

在一种可能的实现方式中，所述方法还包括：将数据处理单元的内存的地址传输至所述处理器；接收来自所述处理器的所述访问内存地址。其中，将数据处理单元的内存的地址传输至所述处理器的示例性实现方式，可以参照上文以及图 4、图 13 中步骤 S1 的相关描述，以及上文及图 5 的相关描述。接收来自所述处理器的所述访问内存地址的示例性实现方式，可以参照上文以及图 4、图 13 中步骤 S2 的相关描述，以及上文及图 6 的相关描述。

通过这种方式，使得数据处理单元可以获取到存储器可以访问的访问内存地址，从而能使用该访问内存地址作为数据处理单元通过 PCIE 点对点传输方式进行数据传输的地址，使得数据处理单元和存储器在不直接连接的情况下传输数据成为可能。

在一种可能的实现方式中，所述方法还包括：接收所述处理器发送的第一缓存队列和第二缓存队列的地址，所述第一缓存队列的地址用于将访问请求写入所述处理器的第一缓存队列，所述第二缓存队列的地址用于从所述第二缓存队列中获取所述访问请求的执行结果。其

示例性实现方式,可以参照上文以及图4中步骤S3的相关描述,以及上文及图7的相关描述。

通过这种方式,使得数据处理单元能够根据接收到的第一缓存队列的地址找到第一缓存队列,数据处理单元将访问请求写入第一缓存队列时,可以直接将访问请求写入;使得数据处理单元能够根据接收到的第二缓存队列的地址找到第二缓存队列,数据处理单元从第二缓存队列获取访问请求的执行结果时,能够直接获取访问请求的执行结果。从而保证写入访问请求和获取访问请求结果的实时性和准确度。

图16示出根据本申请实施例的数据访问方法的示例性工作流程的示意图。

如图16所示,本申请提出一种数据访问方法,所述方法应用于计算设备,所述计算设备包括处理器、连接所述处理器的存储器、以及连接所述处理器且与所述存储器不直接连接的数据处理单元,所述方法包括步骤S12-S14:

步骤S12,所述数据处理单元将访问请求写入所述处理器的第一缓存队列,所述访问请求包括待访问的所述存储器的标识;

步骤S13,所述数据处理单元通过高速外围组件互联(PCIe)点对点传输方式向所述存储器的标识指向的所述存储器发送第一指令,所述第一指令指示所述访问请求在所述第一缓存队列中的位置;

步骤S14,所述存储器根据所述第一指令从所述第一缓存队列中获取所述访问请求并执行所述访问请求。

其中,步骤S12的示例性实现方式,可以参照上文以及图4、图13中步骤S4的相关描述。步骤S13的示例性实现方式,可以参照上文以及图4、图13中步骤S5的相关描述。步骤S14的示例性实现方式,可以参照上文以及图4、图13中步骤S6的相关描述。

根据本申请实施例的数据访问方法,先使用数据处理单元将包括待访问的存储器的标识的访问请求写入处理器的第一缓存队列,使得访问请求可以存储在处理器中的第一缓存队列中;再使用数据处理单元通过PCIe点对点传输方式向存储器的标识指向的存储器发送第一指令,使得待访问的存储器能够接收到第一指令;第一指令包括访问请求在第一缓存队列中的位置信息,所述第一指令用于指示存储器从第一缓存队列获取访问请求并执行访问请求,使得存储器接收到第一指令后,能够按照第一指令的指示,从第一缓存队列中的相应位置处获取访问请求,并执行访问请求。通过这种方式,使得计算设备中,在存储器连接处理器且不直接连接数据处理单元时,计算设备无需通过主机的处理器对存储器进行访问,也能够实现对与处理器连接的存储器的直接访问,从而不会占用主机的资源。并且,这种访问方式可以简化硬件连接方式,降低硬件设计的复杂度。

在一种可能的实现方式中,所述访问请求还包括访问内存地址,所述访问内存地址为所述处理器为所述数据处理单元的内存分配的地址,所述访问内存地址为所述存储器可访问的地址;步骤S14中,所述执行所述访问请求包括:所述存储器根据所述访问内存地址,通过PCIe点对点传输方式将数据写入所述数据处理单元的内存或从所述数据处理单元的内存中读取数据。

其中,访问内存地址可以参见上文以及图6的相关描述中的访问内存地址的示例。存储器将数据写入所述数据处理单元的内存或从所述数据处理单元的内存中读取数据的示例性实现方式,可以参照上文以及图4、图13中步骤S6的相关描述。

通过这种方式,使得存储器可以根据访问内存地址找到访问对象也即数据处理单元的内

存，从而能执行访问请求写入数据到数据处理单元的内存，或者，从数据处理单元的内存读取数据，完成数据处理单元的内存对存储器的访问。

在一种可能的实现方式中，所述访问请求还包括所述访问请求的类型，所述访问请求的类型用于指示所述访问请求为读请求或写请求；当所述访问请求为读请求时，所述执行所述访问请求包括：根据所述访问请求从所述存储器中读取数据，并根据所述访问内存地址，将所述存储器中读取的数据通过 PCIE 点对点传输方式写入所述数据处理单元的内存；当所述访问请求为写请求时，所述执行所述访问请求包括：根据所述访问内存地址，通过 PCIE 点对点传输方式从所述数据处理单元的内存中获取待写入所述存储器的数据，并将获取的数据写入所述存储器。

其中，访问请求的类型可以参见上文以及图 8 的相关描述中的访问请求的类型的示例。访问请求为读请求或写请求时，执行所述访问请求的示例性实现方式，可以参照上文以及图 4、图 13 中步骤 S6 的相关描述。

访问请求为读请求时，存储器能够写入数据到数据处理单元的内存，相当于数据处理单元的内存读存储器；访问请求为写请求时，存储器能够从数据处理单元的内存读取数据，相当于数据处理单元的内存写存储器。通过这种方式，使得对应于访问请求的不同类型，存储器能够以符合访问请求的需求的不同的操作进行响应。

在一种可能的实现方式中，所述方法还包括：所述数据处理单元向所述处理器发送所述数据处理单元的内存的地址；所述处理器根据所述数据处理单元的内存的地址，为所述数据处理单元的内存分配所述访问内存地址；所述处理器向所述数据处理单元发送所述访问内存地址。其示例性实现方式，可以参照上文以及图 4、图 13 中步骤 S1、S2 的相关描述。

通过这种方式，使得处理器可以为数据处理单元的内存分配访问内存地址，从而能使用该访问内存地址作为数据处理单元通过 PCIE 点对点传输方式进行数据传输的地址，使得数据处理单元和存储器在不直接连接的情况下传输数据成为可能。为数据处理单元的内存分配访问内存地址并发送访问内存地址可以在配置阶段进行，使得数据处理单元中的软件产生访问请求之后，直接使用该访问内存地址即可，可以提升计算设备执行访问请求的效率。

在一种可能的实现方式中，所述方法还包括：所述存储器将所述访问请求的执行结果写入所述处理器的第二缓存队列；所述数据处理单元从所述处理器的第二缓存队列中获取所述访问请求的执行结果。其示例性实现方式，可以参照上文以及图 4 中步骤 S7、S8 的相关描述。

在一种可能的实现方式中，所述方法还包括：所述数据处理单元接收所述存储器通过 PCIE 点对点传输方式传输的所述访问请求的执行结果。其示例性实现方式，可以参照上文以及图 13 中步骤 S9 的相关描述。

在一种可能的实现方式中，所述方法还包括：接收所述处理器发送的第一缓存队列和第二缓存队列的地址，所述第一缓存队列的地址用于将访问请求写入所述处理器的第一缓存队列，所述第二缓存队列的地址用于从所述第二缓存队列中获取所述访问请求的执行结果。其示例性实现方式，可以参照上文以及图 4 中步骤 S3 的相关描述，以及上文及图 7 的相关描述。

通过这种方式，使得数据处理单元能够根据接收到的第一缓存队列的地址找到第一缓存队列，以使得数据处理单元可以准确、迅速地执行将访问请求写入第一缓存队列、从第二缓存队列获取访问请求的执行结果的操作。第一缓存队列和第二缓存队列的创建，以及第一缓存队列和第二缓存队列的地址的发送可以在配置阶段进行，使得数据处理单元中的软件产生

访问请求之后，直接使用第一缓存队列和第二缓存队列的地址即可，可以提升计算设备执行访问请求的效率。

图 17 示出根据本申请实施例的计算设备的示例性结构示意图。

如图 17 所示，本申请实施例提供一种计算设备，包括：

处理器 101；

存储器 102，连接所述处理器 101；以及

数据处理单元 103，连接所述处理器 101 且与所述存储器 102 不直接连接，用于：将访问请求写入所述处理器 101 的第一缓存队列，所述访问请求包括待访问的所述存储器 102 的标识；通过高速外围组件互联(PCIe)点对点传输方式向所述存储器的标识指向的所述存储器 102 发送第一指令，所述第一指令指示所述访问请求在所述第一缓存队列中的位置；其示例性实现方式，可以参照上文及图 4、图 13 中步骤 S4、S5 的相关描述，以及图 8、图 9 的相关描述。

所述存储器 102，用于根据所述第一指令从所述第一缓存队列中获取所述访问请求，并执行所述访问请求。其示例性实现方式，可以参照上文及图 4、图 13 中步骤 S6 的相关描述，以及图 10 的相关描述。

其中处理器 101 可以参见图 3 中的中央处理器的示例，存储器 102 可以参见图 3 中的存储器 HDD/SSD、NVMe SSD 的示例，数据处理单元 103 可以参见图 3 中的数据处理单元的示例。

在一种可能的实现方式中，所述访问请求还包括访问内存地址，所述访问内存地址为所述处理器 101 为所述数据处理单元 103 的内存分配的地址，所述访问内存地址为所述存储器 102 可访问的地址；

所述存储器 102 用于根据所述访问内存地址，通过 PCIe 点对点传输方式将数据写入所述数据处理单元 103 的内存或从所述数据处理单元 103 的内存中读取数据。

在一种可能的实现方式中，所述访问请求还包括所述访问请求的类型，所述访问请求的类型用于指示所述访问请求为读请求或写请求；

当所述访问请求为读请求时，所述存储器 102 用于：根据所述访问请求从所述存储器 102 中读取数据，并根据所述访问内存地址，将所述存储器 102 中读取的数据通过 PCIe 点对点传输方式写入所述数据处理单元 103 的内存；

当所述访问请求为写请求时，所述存储器 102 用于：根据所述访问内存地址，通过 PCIe 点对点传输方式从所述数据处理单元 103 的内存中获取所述待写入所述存储器 102 的数据，并将获取的数据写入所述存储器 102。

在一种可能的实现方式中，所述处理器 101 用于：

接收所述数据处理单元 103 发送的所述数据处理单元 103 的内存的地址；

根据所述数据处理单元 103 的内存的地址，为所述数据处理单元 103 的内存分配所述访问内存地址；

向所述数据处理单元 103 发送所述访问内存地址。

在一种可能的实现方式中，所述数据处理单元 103 用于：从所述处理器 101 的第二缓存队列中获取所述访问请求的执行结果。

在一种可能的实现方式中，所述数据处理单元 103 用于：接收所述存储器 102 通过 PCIe 点对点传输方式传输的所述访问请求的执行结果。

计算设备可以包括桌面型计算机、膝上型计算机、手持计算机、笔记本电脑、超级移动个人计算机 (ultra-mobile personal computer, UMPC)、上网本、个人数字助理 (personal digital assistant, PDA)、增强现实 (augmented reality, AR) 设备、虚拟现实 (virtual reality, VR) 设备、人工智能 (artificial intelligence, AI) 设备、可穿戴式设备、车载设备、智能家居设备、或智慧城市设备、服务器设备中的至少一种。本申请实施例对该计算设备的具体类型不作特殊限制。

计算设备可以包括处理器 101, 存储器 102, 数据处理单元 103。可以理解的是, 本申请实施例示意的结构并不构成对计算设备的具体限定。在本申请另一些实施例中, 计算设备可以包括比图示更多或更少的部件, 或者组合某些部件, 或者拆分某些部件, 或者不同的部件布置。图示的部件可以以硬件, 软件或软件和硬件的组合实现。

本申请的实施例提供了一种非易失性计算机可读存储介质, 其上存储有计算机程序指令, 所述计算机程序指令被处理器执行时实现上述方法。

本申请的实施例提供了一种计算机程序产品, 包括计算机可读代码, 或者承载有计算机可读代码的非易失性计算机可读存储介质, 当所述计算机可读代码在电子设备的处理器中运行时, 所述电子设备中的处理器执行上述方法。

计算机可读存储介质可以是可以保持和存储由指令执行设备使用的指令的有形设备。计算机可读存储介质例如可以是一——但不限于——电存储设备、磁存储设备、光存储设备、电磁存储设备、半导体存储设备或者上述的任意合适的组合。计算机可读存储介质的更具体的例子 (非穷举的列表) 包括: 便携式计算机盘、硬盘、随机存取存储器 (Random Access Memory, RAM)、只读存储器 (Read Only Memory, ROM)、可擦式可编程只读存储器 (Electrically Programmable Read-Only-Memory, EPROM 或闪存)、静态随机存取存储器 (Static Random-Access Memory, SRAM)、便携式压缩盘只读存储器 (Compact Disc Read-Only Memory, CD-ROM)、数字多功能盘 (Digital Video Disc, DVD)、记忆棒、软盘、机械编码设备、例如其上存储有指令的打孔卡或凹槽内凸起结构、以及上述的任意合适的组合。

这里所描述的计算机可读程序指令或代码可以从计算机可读存储介质下载到各个计算/处理设备, 或者通过网络、例如因特网、局域网、广域网和/或无线网下载到外部计算机或外部存储设备。网络可以包括铜传输电缆、光纤传输、无线传输、路由器、防火墙、交换机、网关计算机和/或边缘服务器。每个计算/处理设备中的网络适配卡或者网络接口从网络接收计算机可读程序指令, 并转发该计算机可读程序指令, 以供存储在各个计算/处理设备中的计算机可读存储介质中。

用于执行本申请操作的计算机程序指令可以是汇编指令、指令集架构 (Instruction Set Architecture, ISA) 指令、机器指令、机器相关指令、微代码、固件指令、状态设置数据、或者以一种或多种编程语言的任意组合编写的源代码或目标代码, 所述编程语言包括面向对象的编程语言—诸如 Smalltalk、C++等, 以及常规的过程式编程语言—诸如 “C” 语言或类似的编程语言。计算机可读程序指令可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中, 远程计算机可以通过任意种类的网络—包括局域网 (Local Area Network, LAN) 或广域网 (Wide Area Network, WAN)—一连接到用户计算机, 或者, 可以连接到外部计算机 (例如利用因特网服务提供商来通过因

特网连接)。在一些实施例中,通过利用计算机可读程序指令的状态信息来个性化定制电子电路,例如可编程逻辑电路、现场可编程门阵列(Field-Programmable Gate Array, FPGA)或可编程逻辑阵列(Programmable Logic Array, PLA),该电子电路可以执行计算机可读程序指令,从而实现本申请的各个方面。

这里参照根据本申请实施例的方法、装置(系统)和计算机程序产品的流程图和/或框图描述了本申请的各个方面。应当理解,流程图和/或框图的每个方框以及流程图和/或框图中各方框的组合,都可以由计算机可读程序指令实现。

附图中的流程图和框图显示了根据本申请的多个实施例的装置、系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上,流程图或框图中的每个方框可以代表一个模块、程序段或指令的一部分,所述模块、程序段或指令的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。在有些作为替换的实现中,方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如,两个连续的方框实际上可以基本并行地执行,它们有时也可以按相反的顺序执行,这依所涉及的功能而定。

也要注意的,框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合,可以用执行相应的功能或动作的硬件(例如电路或ASIC(Application Specific Integrated Circuit, 专用集成电路))来实现,或者可以用硬件和软件的组合,如固件等来实现。

尽管在此结合各实施例对本发明进行了描述,然而,在实施所要求保护的本发明过程中,本领域技术人员通过查看所述附图、公开内容、以及所附权利要求书,可理解并实现所述公开实施例的其它变化。在权利要求中,“包括”(comprising)一词不排除其他组成部分或步骤,“一”或“一个”不排除多个的情况。单个处理器或其它单元可以实现权利要求中列举的若干项功能。相互不同的从属权利要求中记载了某些措施,但这并不表示这些措施不能组合起来产生良好的效果。

以上已经描述了本申请的各实施例,上述说明是示例性的,并非穷尽性的,并且也不限于所披露的各实施例。在不偏离所说明的各实施例的范围的情况下,对于本技术领域的普通技术人员来说许多修改和变更都是显而易见的。本文中所用术语的选择,旨在最好地解释各实施例的原理、实际应用或对市场中的技术的改进,或者使本技术领域的其它普通技术人员能理解本文披露的各实施例。

权 利 要 求 书

1. 一种数据访问方法，其特征在于，所述方法由连接处理器的数据处理单元来执行，所述方法包括：

将访问请求写入所述处理器的第一缓存队列，其中，所述访问请求包括待访问的存储器的标识，其中，所述存储器连接所述处理器且不直接连接所述数据处理单元；

通过高速外围组件互联(PCIe)点对点传输方式向所述存储器的标识指向的所述存储器发送第一指令，其中，所述第一指令包括所述访问请求在所述第一缓存队列中的位置信息，所述第一指令用于指示所述存储器从所述第一缓存队列获取所述访问请求并执行所述访问请求。

2. 根据权利要求1所述的方法，其特征在于，所述访问请求还包括所述访问请求的类型以及访问内存地址，所述访问内存地址为所述处理器为所述数据处理单元的内存分配的地址，所述访问请求的类型用于指示所述访问请求为读请求或写请求，所述访问内存地址为所述存储器可访问的地址。

3. 根据权利要求1或2所述的方法，其特征在于，所述方法还包括：

从所述处理器的第二缓存队列中获取所述访问请求的执行结果。

4. 根据权利要求1或2所述的方法，其特征在于，所述方法还包括：

接收所述存储器通过PCIe点对点传输方式传输的所述访问请求的执行结果。

5. 根据权利要求2-4中任一项所述的方法，其特征在于，所述方法还包括：

将数据处理单元的内存的地址传输至所述处理器；

接收来自所述处理器的所述访问内存地址。

6. 一种数据访问方法，其特征在于，所述方法应用于计算设备，所述计算设备包括处理器、连接所述处理器的存储器、以及连接所述处理器且与所述存储器不直接连接的数据处理单元，所述方法包括：

所述数据处理单元将访问请求写入所述处理器的第一缓存队列，所述访问请求包括待访问的所述存储器的标识；

所述数据处理单元通过高速外围组件互联(PCIe)点对点传输方式向所述存储器的标识指向的所述存储器发送第一指令，所述第一指令指示所述访问请求在所述第一缓存队列中的位置；

所述存储器根据所述第一指令从所述第一缓存队列中获取所述访问请求并执行所述访问请求。

7. 根据权利要求 6 所述的方法，其特征在于，所述访问请求还包括访问内存地址，所述访问内存地址为所述处理器为所述数据处理单元的内存分配的地址，所述访问内存地址为所述存储器可访问的地址；

所述执行所述访问请求包括：所述存储器根据所述访问内存地址，通过 PCIE 点对点传输方式将数据写入所述数据处理单元的内存或从所述数据处理单元的内存中读取数据。

8. 根据权利要求 7 所述的方法，其特征在于，所述访问请求还包括所述访问请求的类型，所述访问请求的类型用于指示所述访问请求为读请求或写请求；

当所述访问请求为读请求时，所述执行所述访问请求包括：根据所述访问请求从所述存储器中读取数据，并根据所述访问内存地址，将所述存储器中读取的数据通过 PCIE 点对点传输方式写入所述数据处理单元的内存；

当所述访问请求为写请求时，所述执行所述访问请求包括：根据所述访问内存地址，通过 PCIE 点对点传输方式从所述数据处理单元的内存中获取待写入所述存储器的数据，并将获取的数据写入所述存储器。

9. 根据权利要求 7 或 8 所述的方法，其特征在于，所述方法还包括：

所述数据处理单元向所述处理器发送所述数据处理单元的内存的地址；

所述处理器根据所述数据处理单元的内存的地址，为所述数据处理单元的内存分配所述访问内存地址；

所述处理器向所述数据处理单元发送所述访问内存地址。

10. 根据权利要求 6-9 中任一项所述的方法，其特征在于，所述方法还包括：

所述存储器将所述访问请求的执行结果写入所述处理器的第二缓存队列；

所述数据处理单元从所述处理器的第二缓存队列中获取所述访问请求的执行结果。

11. 根据权利要求 6-9 中任一项所述的方法，其特征在于，所述方法还包括：

所述数据处理单元接收所述存储器通过 PCIE 点对点传输方式传输的所述访问请求的执行结果。

12. 一种计算设备，其特征在于，包括：

处理器；

存储器，连接所述处理器；以及

数据处理单元，连接所述处理器且与所述存储器不直接连接，用于：

将访问请求写入所述处理器的第一缓存队列，所述访问请求包括待访问的所述存储器的标识；

通过高速外围组件互联(PCIe)点对点传输方式向所述存储器的标识指向的所述存储器发送第一指令，所述第一指令指示所述访问请求在所述第一缓存队列中的位置；

所述存储器，用于根据所述第一指令从所述第一缓存队列中获取所述访问请求并执行所述访问请求。

13. 根据权利要求 12 所述的设备，其特征在于，所述访问请求还包括访问内存地址，所述访问内存地址为所述处理器为所述数据处理单元的内存分配的地址，所述访问内存地址为所述存储器可访问的地址；

所述存储器用于根据所述访问内存地址，通过 PCIe 点对点传输方式将数据写入所述数据处理单元的内存或从所述数据处理单元的内存中读取数据。

14. 根据权利要求 13 所述的设备，其特征在于，所述访问请求还包括所述访问请求的类型，所述访问请求的类型用于指示所述访问请求为读请求或写请求；

当所述访问请求为读请求时，所述存储器用于：根据所述访问请求从所述存储器中读取数据，并根据所述访问内存地址，将所述存储器中读取的数据通过 PCIe 点对点传输方式写入所述数据处理单元的内存；

当所述访问请求为写请求时，所述存储器用于：根据所述访问内存地址，通过 PCIe 点对点传输方式从所述数据处理单元的内存中获取所述待写入所述存储器的数据，并将获取的数据写入所述存储器。

15. 根据权利要求 13 或 14 所述的设备，其特征在于，所述处理器用于：

接收所述数据处理单元发送的所述数据处理单元的内存的地址；

根据所述数据处理单元的内存的地址，为所述数据处理单元的内存分配所述访问内存地址；

向所述数据处理单元发送所述访问内存地址。

16. 根据权利要求 12-15 中任一项所述的设备，其特征在于，所述数据处理单元用于：

从所述处理器的第二缓存队列中获取所述访问请求的执行结果。

17. 根据权利要求 12-15 中任一项所述的设备，其特征在于，所述数据处理单元用于：

接收所述存储器通过 PCIe 点对点传输方式传输的所述访问请求的执行结果。