

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2024 年 4 月 4 日 (04.04.2024)



(10) 国际公布号
WO 2024/066496 A1

(51) 国际专利分类号:
G06F 9/44 (2018.01) **G06F 12/14 (2006.01)**

(21) 国际申请号: PCT/CN2023/100809

(22) 国际申请日: 2023 年 6 月 16 日 (16.06.2023)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
202211213882.0 2022年9月30日 (30.09.2022) CN

(71) 申请人: 华为技术有限公司 (HUAWEI TECHNOLOGIES CO., LTD.) [CN/CN]; 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。

(72) 发明人: 钟刊 (ZHONG, Kan); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。崔文林 (CUI, Wenlin); 中国广东省深圳市龙岗区坂田华为总部办公楼, Guangdong 518129 (CN)。

(74) 代理人: 北京亿腾知识产权代理事务所 (普通合伙) (E-TONE INTELLECTUAL PROPERTY FIRM (GENERAL PARTNERSHIP)); 中国北京市海淀区中关村紫金数码园 3 号楼 707, Beijing 100190 (CN)。

(81) 指定国(除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ,

(54) Title: DATA ACCESS METHOD AND APPARATUS

(54) 发明名称: 一种数据的访问方法及装置

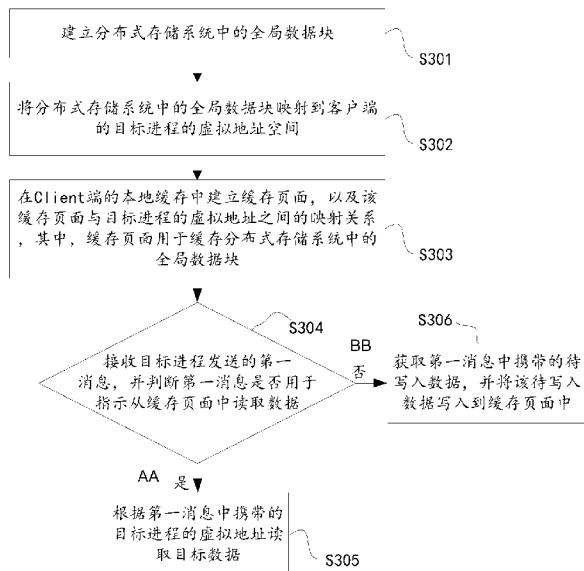


图 3

(57) Abstract: A data access method, comprising: in a running procedure of a target process, acquiring a first virtual address of target data in the target process; querying a first address mapping relationship table on the basis of the first virtual address, and determining whether the target data is stored in a local cache; if the target data is stored in the local cache, querying the first address mapping relationship table on the basis of the first virtual address, so as to obtain a physical address of the target data in the local cache; and acquiring the target data according to a physical address of the target data in the local cache. After a file in a distributed storage system is mapped by using distributed mmap in the distributed storage system, a distributed virtual address of the file in the distributed storage system is locally recorded. In a running procedure of a target process, if a local cache is not hit, a remote DVA space is accessed by

LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN,
MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA,
PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD,
SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ,
UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW。

- (84) 指定国(除另有指明, 要求每一种可提供的地区
保护): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ,
NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚
(AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE,
BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR,
HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO,
PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF,
CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN,
TD, TG)。

根据细则4.17的声明:

- 关于申请人有权要求在先申请的优先权(细则
4.17(iii))

本国际公布:

- 包括国际检索报告(条约第21条(3))。

means of a page fault, thereby reducing delay overheads of an end-to-end software stack.

(57) 摘要: 一种数据访问方法, 包括: 在目标进程运行过程中, 获取目标数据在目标进程中的第一虚拟地址;
基于所述第一虚拟地址, 查询第一地址映射关系表, 确定本地缓存中是否存储有目标数据, 若本地缓存中存储
有目标数据, 基于所述第一虚拟地址, 查询所述第一地址映射关系表, 得到所述目标数据在本地缓存中的物理
地址; 根据所述目标数据在本地缓存中的物理地址, 获取所述目标数据。在分布式存储系统中使用Distributed
mmap映射分布式存储系统中的文件之后, 在本地记录该文件在分布式存储系统中的分布式虚拟地址。在目标
进程运行过程中, 若本地缓存不命中, 通过Page Fault访问远端DVA空间, 节省端到端软件栈的时延开销。

一种数据的访问方法及装置

5 本申请要求在 2022 年 9 月 30 日提交中国国家知识产权局、申请号为 202211213882.0, 发明名称为“一种数据的访问方法及装置”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

本发明涉及计算机技术领域, 尤其涉及一种数据的访问方法及装置。

10 背景技术

传统的文件访问路径长, 在文件访问过程中, 需要经过文件虚拟系统(virtual file system, VFS)、文件系统客户端、文件系统服务端以后, 最终访问持久内存(persistent memory, PM)中的文件。在整个的访问过程中, 软件栈的开销较大。其中, 软件堆栈是指应用程序运行所需的程序或组件的集合。

15 mmap 是一种内存映射文件的方法, 即将一个文件或者其他对象映射到进程的地址空间, 建立文件或者其他对象的地址与进程的虚拟地址空间中一段虚拟地址的一对一映射关系。mmap 虽然能够在文件访问过程中, 减少物理空间占用。但是, mmap 局限于单节点内部对文件、对象、块设备等实现内存映射, 没有跨节点映射的能力。

发明内容

20 本申请提供了一种数据的访问方法、装置、计算机可读存储介质、计算机程序产品。针对分布式存储系统, 构建分布式存储系统中的全局共享内存池, 也可以称之为全局内存池。然后, 基于该全局内存池构建全局数据块, 并将该全局数据块映射到客户端的进程地址空间中。然后, 客户端建立用于缓存分布式系统中的全局数据块的本地缓存, 并通过客户端的本地页表, 建立进程虚拟地址到缓存页面的映射。即在分布式存储系统中使用 Distributed mmap 映射远端文件之后, 在本地记录文件的 DVA。当本地缓存不命中, 通过 Page Fault 访问远端 DVA 空间, 节省端到端软件栈的时延开销。

25 第一方面, 本申请提供了一种数据访问方法, 该方法包括: 在目标进程运行过程中, 获取目标数据在目标进程中的第一虚拟地址; 基于第一虚拟地址, 查询第一地址映射关系表, 确定本地缓存中是否存储有目标数据, 第一地址映射关系表中包括: 第一数据集合中的数据在进程中的虚拟地址和第一数据集合中的数据在本地缓存中的物理地址间的对应关系, 其中, 第一数据集合中的数据预先从分布式存储系统中获取得到; 若本地缓存中存储有目标数据, 基于第一虚拟地址, 查询第一地址映射关系表, 得到目标数据在本地缓存中的物理地址; 根据目标数据在本地缓存中的物理地址, 获取目标数据。

30 也就是说, 在分布式存储系统中, 可以基于分布式存储节点构成的全局内存池, 以及基于全局内存池构建分布式存储系统的全局数据块。客户端处理器可以将全局数据块中的部分数据映射到目标进程的虚拟地址空间中, 并生成第一地址映射关系表。以及客户端处理器在客户端的本地内存中建立类似于 35 Page Cache 的缓存页面, 用于缓存全局数据块, 并建立本地缓存页面与目标进程的虚拟地址间的映射关系。然后客户端处理器根据该映射关系更新第一地址映射关系表。在目标进程的运行过程中, 客户端处理器根据目标数据在目标进程中的第一虚拟地址, 确定本地缓存的缓存页面中是否存储有目标数据。当本地缓存的缓存页面中存储有目标数据时, 客户端处理器可以直接从本地缓存的缓存页面中读取目标数据。即在分布式存储系统中。通过建立全局数据块的虚拟地址到目标进程虚拟地址间的映射, 以及进 40 程虚拟地址到缓存页面的映射, 可以实现客户端进程对全局数据块的访问快速访问, 节省了客户端到存储节点软件栈的时延开销。

在一个可能的实现方式中, 第一地址映射关系表中还包括: 第二数据集合中的数据在进程中的虚拟地址和第二数据集合中的数据在分布式存储系统中的虚拟地址间的对应关系, 方法还包括: 若本地缓存中没有存储有目标数据, 基于第一虚拟地址, 查询第一地址映射关系表, 得到目标数据在分布式存储系统中的第二虚拟地址; 基于第二虚拟地址, 查询第二地址映射关系表, 得到目标数据在分布式存储系统中

的物理地址；第二地址映射关系表中包括：在分布式存储系统中存储的第三数据集合中的数据的虚拟地址和第三数据集合中的数据在分布式存储系统中的物理地址间的对应关系；其中，第三数据集合中的数据包括：第一数据集合中的数据和第二数据集合中的数据；基于目标数据在分布式存储系统中的物理地址，从分布式存储系统中获取目标数据。

也就是说，在本地缓存中没有缓存目标数据的情况下。客户端处理器需要从分布式存储系统的存储节点中，将目标数据读取到缓存页面中。具体地，客户端处理器可以根据目标数据的第一虚拟地址查询第一地址映射关系表得到目标数据在分布式存储系统中的全局虚拟地址，即第二虚拟地址。然后客户端处理器可以根据第二虚拟地址，查询分布式存储系统中的第二虚拟地址表，得到目标数据在分布式存储系统中的物理地址，并根据该物理地址从分布式存储系统中获取目标数据。其中，第二虚拟地址表是分布式存储系统构建全局数据时，得到的地址映射表。

在一个可能的实现方式中，在基于目标数据在分布式存储系统中的物理地址，从分布式存储系统中获取目标数据以后，该方法还包括：将目标数据存储到本地缓存中；建立目标数据在目标进程中的第一虚拟地址和目标数据在本地缓存中的物理地址间的第一对应关系；用第一对应关系替换第一地址映射关系表中的第二对应关系，第二对应关系为目标数据在目标进程中的虚拟地址和目标数据在分布式存储系统中的虚拟地址间的对应关系。

也就是说，客户端处理器在从分布式存储系统中获取到目标数据以后，还需要将该目标数据缓存在本地缓存中，并建立该目标数据在本地缓存中的物理地址与该目标数据在目标进程中的虚拟地址间的映射关系。然后，客户端处理器将该映射关系保存在第一地址映射关系表中。使得当目标进程需要再次获取目标数据时，可以直接从本地缓存中进行获取。节省了目标进程获取目标数据的时延开销。

在一个可能的实现方式中，该方法还包括：在目标进程运行过程中，获取待写入数据在目标进程中的第三虚拟地址；基于第三虚拟地址，查询第一地址映射关系表，确定本地缓存中是否存储有与第三虚拟地址对应的数据；若本地缓存中存储有与第三虚拟地址对应的数据，将本地缓存中与第三虚拟地址对应的数据替换为待写入数据。

也就是说，客户端进程需要向分布式存储系统中的存储节点写入数据时，客户端进程可以直接将需要写入的数据缓存在客户端的缓存页面中。然后，客户端处理器可以将存储有客户端进程写入数据的缓存页面标记为脏页面，并周期性的将脏页面写回到存储节点中。简化了客户端进程对存储节点的写入过程。

第二方面，本申请提供了一种数据处理装置，包括：

获取模块，用于在目标进程运行过程中，获取目标数据在目标进程中的第一虚拟地址；

处理模块，用于根据第一虚拟地址，查询第一地址映射关系表，确定本地缓存中是否存储有目标数据，第一地址映射关系表中包括：第一数据集合中的数据在进程中的虚拟地址和第一数据集合中的数据在本地缓存中的物理地址间的对应关系，其中，第一数据集合中的数据预先从分布式存储系统中获取得到；若本地缓存中存储有目标数据，基于第一虚拟地址，查询第一地址映射关系表，得到目标数据在本地缓存中的物理地址；

获取模块，还用于根据目标数据在本地缓存中的物理地址，获取目标数据。

在一个可能的实现方式中，第一地址映射关系表中还包括：第二数据集合中的数据在进程中的虚拟地址和第二数据集合中的数据在分布式存储系统中的虚拟地址间的对应关系，处理模块还用于：

若本地缓存中没有存储有目标数据，基于第一虚拟地址，查询第一地址映射关系表，得到目标数据在分布式存储系统中的第二虚拟地址；根据第二虚拟地址，查询第二地址映射关系表，得到目标数据在分布式存储系统中的物理地址；第二地址映射关系表中包括：在分布式存储系统中存储的第三数据集合中的数据的虚拟地址和第三数据集合中的数据在分布式存储系统中的物理地址间的对应关系；其中，第三数据集合中的数据包括：第一数据集合中的数据和第二数据集合中的数据；

获取模块还用于，根据目标数据在分布式存储系统中的物理地址，从分布式存储系统中获取目标数据。

在一个可能的实现方式中，处理模块还用于：

将目标数据存储到本地缓存中；

建立目标数据在目标进程中的第一虚拟地址和目标数据在本地缓存中的物理地址间的第一对应关

系；

用第一对应关系替换第一地址映射关系表中的第二对应关系，第二对应关系为目标数据在目标进程中的虚拟地址和目标数据在分布式存储系统中的虚拟地址间的对应关系。

在一个可能的实现方式中，获取模块还用于，在目标进程运行过程中，获取待写入数据在目标进程中的第三虚拟地址；

处理模块还用于，根据第三虚拟地址，查询第一地址映射关系表，确定本地缓存中是否存储有与第三虚拟地址对应的数据；若本地缓存中存储有与第三虚拟地址对应的数据，将本地缓存中与第三虚拟地址对应的数据替换为待写入数据。

第三方面，本申请提供了一种计算机存储介质，计算机存储介质中存储有指令，当指令在计算机上运行时，使得计算机执行第一方面所提供的方法。

第四方面，本申请提供了一种包含指令的计算机程序产品，当指令在计算机上运行时，使得计算机执行第一方面所提供的方法。

可以理解的是，上述第二方面至第四方面的有益效果可以参见上述第一方面中的相关描述，在此不再赘述。

附图说明

图1为本申请实施例提供的一种基于mmap的数据访问示意图；

图2为本申请实施例提供的一种应用场景示意图；

图3为本申请实施例提供的一种数据访问方法的流程示意图；

图4为本申请实施例提供的一种访问分布式存储系统中的数据的过程示意图；

图5为本申请实施例提供的又一种访问分布式存储系统中的数据的过程示意图；

图6为本申请实施例提供的又一种访问分布式存储系统中的数据的过程示意图；

图7为本申请实施例提供的一种传统文件访问路径与全局mmap访问路径的比较示意图；

图8为本申请实施例提供的一种数据处理装置的结构示意图；

图9为本申请实施例提供的一种计算设备的结构示意图。

具体实施方式

为了使本申请实施例的目的、技术方案和优点更加清楚，下面将结合附图，对本申请实施例中的技术方案进行描述。

在本申请实施例中的描述中，“示例性的”、“例如”或者“举例来说”的任何实施例或设计方案不应该被理解为比其他实施例或设计方案更优选或更具优势。确切而言，使用“示例性的”、“例如”或者“举例来说”等词旨在以具体方式呈现相关概念。

此外，术语“第一”、“第二”仅用于描述目的，而不能理解为指示或暗示相对重要性或者隐含指明所指示的技术特征。由此，限定有“第一”、“第二”的特征可以明示或者隐含地包括一个或者更多个该特征。术语“包括”、“包含”、“具有”及它们的变形都意味着“包括但不限于”，除非是以其他方式另外特别强调。

在介绍本发明申请的方案之前，首先对本发明申请实施例中需要用到的关键术语进行解释。

1、mmap是一种内存映射文件的方法，即将文件磁盘中的文件或者其他对象映射到应用程序的进程的地址空间（用户空间），实现文件磁盘地址和进程虚拟地址空间中一段虚拟地址的一一对映关系。实现这样的关系后，进程就可以采用指针的方式读写操作这一段内存，而系统会自动回写脏页面到对应的文件磁盘上，即完成了对文件的操作而不必再调用read, write等系统调用函数。相反，内核空间对这段区域的修改也可以直接反映到用户空间，从而可以实现不同进程间的文件共享。

2、Bypass，指可以通过特定的触发状态（断电或死机）让两个网络不通过网络安全设备的系统，而直接物理上导通。所以有了Bypass后，当网络安全设备故障以后，还可以让连接在这台设备上的网络相互导通，此时，网络设备不会再对网络中的封包做处理。

3、全局内存池，是指将在物理上分离的多个内存池，在逻辑上组合为一个内存池。

接着对本申请涉及的技术方案进行介绍。

示例性的，图1示出了一种基于mmap的数据访问示意图。如图1所示，电子设备上的操作系统为应用程序的每一个进程分配独立的虚拟地址空间。然后，目标进程(可以是应用程序中的任意一个进程)在用户空间中调用库函数mmap，在目标进程的虚拟地址空间中，寻找一段空闲的满足要求的连续的虚拟地址。然后，将寻找到的目标进程中的连续的虚拟地址与存储在文件空间(磁盘)中的文件相关联。其中，在调用mmap函数时，需要标明被映射的文件的标识(file)、被映射的文件在文件空间(磁盘)上的起点(offset)以及被映射的文件的长度(len)。在建立目标进程的虚拟地址与用户空间中的文件的映射关系以后，操作系统通过内存管理单元(Memory Management Unit, MMU)建立进程的虚拟地址与电子设备内存中的物理地址(Page Cache中的Page)的映射关系，并将该映射关系存储在Page Table中。其中，Page Cache中的Page可以用于对文件空间中的文件内容进行缓存。可以理解的是，每一个进程都会有自己的Page Table，Page Table中存储了该进程中虚拟地址到物理地址的映射关系。

当应用程序的进程需要读取该进程地址空间中的数据时，进程通过电子设备的CPU向MMU发送该需要读取的数据的虚拟地址。MMU接收到CPU发送的虚拟地址之后，开始查询该进程的Page Table，确定该进程的Page Table中是否存在映射，以及该进程的读写权限是否正常。在一个可能的示例中，当Page Table中不存在与需要读取的数据的虚拟地址对应的物理地址(即本地缓存不命中)，CPU产生缺页中断(Page Fault)。此时，进程会从用户态切换到内核态，并将Page Fault交给内核的Page Fault Handler进行处理。具体地，CPU可以通过read/prefetch从文件空间中读取数据，并将读取的数据存储到Page Cache中的Page中。然后，CPU触发MMU建立需要读取的数据的虚拟地址与Page Cache中的物理地址之间的映射关系，并将该映射关系保存在Page Table中。在另一个可能的示例中，当Page Cache中存在与需要读取的数据的虚拟地址对应的物理地址(即本地缓存命中)，CPU可以直接读取Page Cache中的数据。

当应用程序的进程在执行过程中，需要修改文件数据时，一般来说修改的文件数据不会马上同步到磁盘(或者说文件空间)，修改后的数据会先缓存在内存的Page Cache中。此时，可以把这种和磁盘数据不一致的页称为脏页，脏页会在合适的时机同步到磁盘。

虽然，基于mmap的数据访问，使得多进程可以通过mmap访问同一个文件，而被访问的文件在Page Cache中只存放一份，减少了物理空间的占用。以及基于mmap的数据访问能够实现用户空间和内核空间的数据直接交互而省去空间不同，数据不同的繁琐过程。但是，mmap局限于单节点内部对文件、对象、块设备等实现内存映射，没有跨节点映射能力。即基于图1所示的mmap数据访问方式，不能实现分布式存储系统中的全局数据访问。

有鉴于此，本申请实施例提供了一种数据访问方法，可以应用于分布式存储系统。利用分布式存储系统中各个存储节点的物理存储器，组成全局地址空间，并对外提供分布式物理地址空间(distributed physical address space, DPAS)，分布式物理地址通过分布式页表(distributed page table, DPT)映射到分布式虚拟地址(distributed virtual address, DVA)。分布式存储系统中的全局数据块可以基于DVA进行构建。其中，全局数据块包括：用户文件或者对象。进一步地，分布式存储系统中的客户(Client)端中的应用程序可以通过distributed mmap将全局数据块映射到本地进程的地址空间中，并通过load/store对其进行访问。使得分布式存储系统中的Client端的应用程序更快的访问分布式存储节点中的数据。

示例性的，图2示出了本发明申请实施例提供的一种应用场景示意图。如图2所示，包括分布式存储系统和Client端。分布式存储系统中包括多个存储节点(Node)。存储节点可以用于存储数据。在一些可能的示例中，存储节点中可以包括：动态随机存取存储器(dynamic random access memory, DRAM)、PM等物理存储介质。多个存储节点中的物理存储介质，组成了分布式存储系统的全局地址空间，并对外提供分布式物理地址空间。分布式物理地址空间中的分布式物理地址(distributed physical address, DPA)可以通过DPT映射到DVA。基于分布式存储系统的DVA可以构建分布式存储系统中的全局数据块。其中，DPA可以通过全局内存总线访问存储节点中的数据，以及Bypass存储节点的CPU。

Client端上安装有应用程序。当应用程序运行时，应用程序可以通过调用Distributedmmap函数，将分布式存储系统中基于DVA构建的全局数据块中的数据映射到Client端的本地进程的地址空间中，

并通过 load/store 对映射到本地进程的地址空间中的数据进行读写操作。

示例性的,本申请实施例中提供了一种数据的访问方法的流程示意图。该方法可以应用于图 2 所示的分布式存储系统。参见图 3,该方法包括:S301-S306。

5 S301,建立分布式存储系统中的全局数据块。

在本实施例中,由于针对的是分布式存储系统中存储的数据。所以,在对分布式存储系统中的数据进行访问之前,分布式存储系统中的管理软件可以先构建分布式存储系统的全局数据块。

10 在一个可能的示例中,如图 4 所示,分布式存储系统中的多个存储节点(Storage node)中的每一个存储节点中的存储介质(比如,AEP、DRAM、XL-Flash)构成了分布式存储系统的全局地址空间,对外提供分布式物理地址地址空间。其中,分布式存储系统的分布式物理地址空间可以通过 DPA 进行访问。即获取到 DPA 以后,根据获取的 DPA 可以直接通过全局内存数据总线直接访问到存储节点中的数据。

15 分布式存储系统中的管理软件在获取到分布分布式存储系统中的 DPA 以后,为了让使用该分布式存储系统的用户都可以共享整个分布式存储系统的存储空间,还需要将分布式存储系统的 DPA 映射为 DVA。在将分布式存储系统的 DPA 映射为 DVA 时,需要在分布式存储系统中生成 DPT。该 DPT 可以存储在分布式存储系统的任意一个存储节点,且该 DPT 用于存放分布存储系统的 DVA 到 DPT 的映射关系。

在得到分布式存储系统的 DVA 以后,分布式存储系统中的管理软件可以基于得到的分布式存储系统的 DVA 构建分布式存储系统的全局数据块。在一个可能的示例中,分布式存储系统的全局数据块可以包括:用户文件或者对象。

S302,将分布式存储系统的全局数据块映射到客户端的目标进程的虚拟地址空间。

20 在本实施例中,Client 端的操作系统会为每一个进程都分配一个固定大小的虚拟地址空间。其中,进程的虚拟地址空间由多个虚拟内存区域构成。虚拟内存区域是进程的虚拟地址空间中的一个同质区间,即具有同样特性的连续地址范围。在一个可能的示例中,如图 4 所示,图 4 中的 Client 端的进程的虚拟地址空间中包括:栈区(stack)、文件映射区、堆(heap)、bss 数据段、初始数据段、text 数据段。其中,文件映射区为堆、栈之间的空余部分。

25 在启动映射的过程中,目标进程可以在用户空间调用库函数 mmap。库函数 mmap 的原型为:

void *mmap(void *start, size_t length, int prot, int flags, int fd, off_t offset)。

其中,参数 start 指向欲映射的内存的起始地址;参数 length 代表需要将分布式存储系统中的全局数据块中的多大部分映射到内存;参数 port 表示映射区域的保护方式;参数 flags 表示影响映射区域的各种特性;参数 fd 表示要映射到内存中的全局数据块的文件描述符;参数 offset 表示全局数据块映射时的偏移量,通常设置为 0,代表被映射的全局数据块从整个数据块的最前方开始对应,其中,offset 必须是分页大小的整数倍。

30 当目标进程调用库函数 mmap 将存储在分布式存储节点中的全局数据块,映射到目标进程的虚拟地址空间以后,Client 端的 CPU 还需要将全局数据块的 DVA,与全局数据块在目标进程的虚拟地址空间中的虚拟地址(virt)之间的映射关系,保存到 Client 端的 MMU 维护的本地页表中,比如图 4 中的 Page Table 中保存的映射关系。以及 Client 端的 CPU 还需要将全局数据块的 DVA,与全局数据块在目标进程的虚拟地址空间中的虚拟地址(virt)之间的映射关系,保存到 mmap 管理软件中。

35 S303,在 Client 端的本地缓存中建立缓存页面,以及该缓存页面与目标进程的虚拟地址之间的映射关系,其中,缓存页面用于缓存分布式存储系统中的全局数据块。

40 在本实施例中,Client 端的 CPU 可以将 Client 端的物理内存作为本地缓存。Client 端的操作系统在为进程分配虚拟的地址空间以后,还需要通过 MMU 将该虚拟空间中的虚拟地址映射到 client 端的本地缓存中。其中,虚拟地址可以通过页表映射到本地缓存,页表由 Client 端的操作系统维护,并被处理器引用。

在一个可能的示例中,进程的虚拟地址空间可以根据一个固定的大小进行划分,本地缓存也可以设置成不同的页面大小,通常情况下,本地缓存中的缓存页面的大小和虚拟页的大小是一样的。

45 在 Client 端建立了本地缓存页面以后,Client 端的目标进程可以触发 CPU 将分布存储节点中存储的数据读取到 Client 端的本地缓存页面。具体地,Client 端的 CPU 获取到待读取数据在目标进程中的虚拟地址以后,Client 端的 CPU 可以根据待读取数据在目标进程中的虚拟地址,查找本地页表得到待读

取数据在分布式存储系统中的 DVA，然后 Client 端的 CPU 向分布式存储系统发送读取数据请求命令，该命令中携带待读取数据的 DVA。分布式存储系统中的 CPU 获取到该 DVA 以后，根据该 DVA 查找分布式存储系统中的 DPT，得到待读取数据的 DPA。然后，分布式存储系统中的 CPU 根据该 DPT 获取到待读取数据，并将该待读取数据发送给 Client 端的 CPU。Client 端的 CPU 接收到该待读取数据以后，将该待读取数据存储在本地缓存页面中。然后，Client 端的 CPU 还需要，建立该缓存页面与待读取数据在进程中的虚拟地址间的映射关系，并用该映射关系替换本地页表中的待读取数据在进程中的虚拟地址与待读取数据的 DVA 间的映射关系。

在一个可能的示例中，在 Client 端可以部署有数据处理装置（data processing unit, DPU）。DPU 中可以预先缓存部分分布式存储系统中的 DPT。当 CPU 需要将分布式存储系统中的全局数据块读取到本地缓存页面中时，CPU 可以根据 MMU 维护的本地页表获取需要被读取到本地缓存页面中的全局数据块的 DVA，即目标 DVA。然后，CPU 将该目标 DVA 发送给 DPU。由 DPU 根据接收的目标 DVA 从分布式存储系统中获取与该目标 DVA 对应的数据。可以理解的是，DPU 可以是插在 Client 端的 PCIE 设备。DPU 可以具有自己的操作系统、CPU、存储介质和网络接口等。

在一个可能的示例中，DPU 中预先缓存有部分由分布式存储系统维护的 DPT。当 DPU 接收到目标 DVA 后，DPU 可以在 DPU 中缓存的 DPT 中查找是否有与该目标 DVA 对应的 DPA。若 DPU 能够在 DPU 缓存的 DPT 中找到与该目标 DVA 对应的 DPA，DPU 可以直接根据该 DPA 从分布式存储系统的存储节点中读取存储在该 DPA 处的数据，并将该数据发送给 Client 端的 CPU。若 DPU 中缓存的 DPT 中不存在与该目标 DVA 对应的 DPA，DPU 需要从分布式存储系统维护的 DPT 中获取与该目标 DVA 对应的 DPA。当 DPU 从分布式存储系统维护的 DPT 中获取到与目标 DVA 对应的 DPA 后，DPU 可以先将该目标 DVA 与 DPA 的对应关系保存到 DPU 缓存的 DPT 中。然后，DPU 可以根据获取到的与该目标 DVA 对应的 DPA，从分布式存储系统的存储节点中读取存储在该 DPA 处的数据，并将该数据发送给 Client 端的 CPU。Client 端的 CPU 接收到 DPU 发送的与目标 DVA 对应的数据以后，CPU 将该数据缓存到 Client 侧的本地缓存的缓存页面中，并触发 MMU 更新本地页表。其中，MMU 更新本地页表是指，将本地页表中保存的目标 DVA 与目标进程的目标虚拟地址的映射关系，更新为目标虚拟地址与本地内存中的物理地址的映射关系。

在本实施例中，页表也叫做映射表。操作系统想要把目标进程的虚拟地址映射到内存的物理地址，最直观的方法就是建立一张映射表。这个映射表能够实现目标进程的虚拟地址里边的页，到物理内存里边的页的映射。

S304，接收目标进程发送的第一消息，当第一消息用于指示从缓存页面中读数据时，执行 S305，当第一消息用于指示向缓存页面中写入数据时，执行 S306。

在本实施例中，当目标进程在运行过程中，需要读取目标进程虚拟地址空间中的某一段数据时，目标进程会向 CPU 发起一个读请求。该读请求携带了需要读取的目标数据块的虚拟地址。当目标进程在运行过程中，需要修改目标进程虚拟地址空间中的某一段数据时，目标进程会向 CPU 发起一个写请求。该写请求中携带了需要修改的目标数据块的虚拟地址。

S305，根据第一消息中携带的目标进程的虚拟地址读取目标数据。

在本实施例中，当目标进程发送的第一消息用于指示从缓存页面中读取数据时，CPU 需要将第一消息中携带的需要读取的目标数据的虚拟地址发送给 MMU。MMU 接收到目标数据块的虚拟地址以后，MMU 需要确定本地页表中是否保存有与该目标数据的虚拟地址对应的本地缓存中的物理地址。当 MMU 确定本地页表中保存有与该目标数据的虚拟地址对应的本地缓存中的物理地址时，MMU 可以触发 CPU 根据该目标数据的虚拟地址对应的物理地址，直接从本地缓存中读取目标数据。当 MMU 确定本地页表中不存在与目标数据块的虚拟地址对应的本地缓存中的物理地址时，MMU 会触发 Client 端的 CPU 产生一个缺页错误，即 Page Fault。响应于该缺页错误，CPU 从分布式存储系统的存储节点中，获取目标数据块，并缓存到本地缓存页面中。

在一个可能的示例中，如图 5 所示，在 Client 端，将 Client 端的内存作为本地缓存，用于缓存分布式存储系统中的全局数据块。Client 端还包括：client 端的 CPU 以及 DPU。当目标程序运行过程中产生了 Page Fault 以后，目标进程会从用户态切换到内核态，并将 Page Fault 交给内核中的内核终端处理器（Page Fault Handler）进行处理。在 Page Fault Handler 处理 Page Fault 的过程中，DPU 可以进行 Page Fault 加速。

参照图 5, MMU 确定本地页表中不存在与目标数据块的虚拟地址对应的本地缓存中的物理地址时, MMU 触发 CPU 产生 Page Fault。响应于该 Page Fault, Client 侧的 CPU 中的 Page Fault Handler 从本地页表中获取目标数据块的虚拟地址对应的 DVA, 即目标 DVA。然后 Page fault Handler 将获取的目标 DVA 发送给 DPU。当 Client 侧的 Page Fault Handler 将目标 DVA 发送给 DPU 以后, Client 端的操作系统需要进行上下文切换, 让出 CPU, 由 DPU 进行后续的 Page Fault 处理。可以理解的是, 在本实施例中, “目标数据块的虚拟地址”指的是目标数据块在目标进程的虚拟地址空间中所具有的虚拟地址。

DPU 接收到 Page fault Handler 发送的目标 DVA 以后, DPU 中的 CPU 可以通过 DPU 内部的内存管理单元 (DPU-MMU) 查找 DPU 内部缓存的 DPT 中, 是否有与目标 DVA 对应的 DPA。当 DPU 中的 CPU 确定 DPU 缓存的 DPT 中存在有与目标 DVA 对应的 DPA 时, DPU 内部的 CPU 可以直接通过 DPU 的网络接口控制器 (network interface controller, NIC), 从分布式存储节点中读取 DPA 处的数据。当 DPU 中的 CPU 确定 DPU 缓存的 DPT 中不存在与目标 DVA 对应的 DPA 时, DPU 内部的 CPU 需要通过 DPU 的 NIC 从分布式存储节点上的 DPT 中获取与目标 DVA 对应的 DPA。DPU 内部的 CPU 获取到与目标 DVA 对应的 DPA 以后, DPU 内部的 CPU 需要触发 DPU-MMU 在 DPU 缓存的 DPT 中增加目标 DVA 与 DPA 的映射关系。以及 DPU 内部的 CPU 获取到与目标 DVA 对应的 DPA 以后, DPU 可以通过 DPU 的 NIC, 从分布式存储节点中读取 DPA 处的数据。

DPU 获取到目标 DVA 对应的数据以后, DPU 将该数据发送给 Page Fault Handler。Page Fault Handler 将目标 DVA 对应的数据存储在 Client 端的本地缓存中的缓存页面上。然后, Page Fault Handler 需要建立该缓存页面, 与目标 DVA 对应的目标进程的虚拟地址之间的映射关系。此时, Page Fault Handler 对 Client 端产生的 Page Fault 处理完成。目标进程可以正常读取 client 本地缓存中的数据。

在一个可能实施例中, 如图 6 所示, 在 Client 端, Client 端的 CPU 可以产生 Page Fault。但是 Client 端 CPU 产生的 Page Fault 会由 DPU 进程处理此时, 可以使用 DPU 的内存作为页面缓存 (即本地缓存页面), 用于缓存分布式存储系统中的全局数据块。

参照图 6, 当 Client 侧的 CPU 获取到目标进程需要读取的数据的虚拟地址以后, Client 侧的 CPU 通过 MMU 从本地页表中获取目标数据块的虚拟地址对应的 DVA, 即目标 DVA。然后 MMU 使用获取的目标地址直接访问 DPU。

DPU 中的 DPU-MMU 接收到 MMU 发送的目标 DVA 以后, DPU-MMU 查询 DPU 中的本地缓存页面, 并进行地址翻译, 获取目标 DVA 对应的物理地址 (即缓存页面中的页面地址)。当目标 DVA 对应的缓存页面在 DPU 的本地缓存中时, DPU 中的 CPU 可以直接从目标 DVA 对应的缓存页面中读取数据。当目标 DVA 对应的缓存页面不在 DPU 的本地缓存中时, DPU-MMU 触发 DPU 内部的 CPU 产生 Page Fault。响应于该 Page Fault, DPU 内部的 CPU 或者 DPU 中的其他硬件单元需要通知 Client 端的操作系统需要进行上下文切换, 让出系统 CPU。然后, DPU 中的 Page Fault Handler 可以通过 DPU 内部的内存管理单元 (DPU-MMU) 查找 DPU 内部缓存的 DPT 中, 是否有与目标 DVA 对应的 DPA。当 DPU 中的 Page Fault Handler 确定 DPU 缓存的 DPT 中存在有与目标 DVA 对应的 DPA 时, DPU 中的 Page Fault Handler 可以直接通过 DPU 的 NIC, 从分布式存储节点中读取 DPA 处的数据。当 DPU 中的 Page Fault Handler 确定 DPU 缓存的 DPT 中不存在与目标 DVA 对应的 DPA 时, DPU 中的 Page Fault Handler 需要通过 DPU 的 NIC 从分布式存储节点上的 DPT 中获取与目标 DVA 对应的 DPA。Page Fault Handler 获取到与目标 DVA 对应的 DPA 以后, DPU 中的 Page Fault Handler 需要触发 DPU-MMU 在 DPU 缓存的 DPT 中增加目标 DVA 与 DPA 的映射关系。以及 DPU 中的 Page Fault Handler 获取到与目标 DVA 对应的 DPA 以后, DPU 中的 Page Fault Handler 可以通过 DPU 的 NIC, 从分布式存储节点中读取 DPA 处的数据。

DPU 中的 Page Fault Handler 读取到目标 DVA 对应的数据以后, 还需要将该目标 DVA 对应的数据写入到 DPU 内部的本地缓存中的缓存页面上。然后, DPU 向 MMU 发送消息, 触发 MMU 建立 DPU 中的缓存页面, 与目标 DVA 对应的目标进程的虚拟地址之间的映射关系。此时, DPU 内部产生的 Page Fault 处理完成。目标进程可以正常读取 GPU 内部本地缓存中的数据。

S306, 获取第一消息中携带的待写入数据, 并将该待写入数据写入到缓存页面中。

在本实施例中, 将待写入数据写入到缓存页面中, 指的是覆盖性写入, 即新写入的数据需要覆盖原始数据。

在一个可能的示例中, 针对图 5 所示的分布式存储系统。当目标进程向 Client 端的 CPU 发送写请

求以后，Client 端的 CPU 获取待写入数据的虚拟地址。该虚拟地址是指“目标进程地址空间中的虚拟地址”。然后，MMU 根据待写入数据的虚拟地址确定本地页表中是否保存有与该数据的虚拟地址本地缓存中的物理地址。当 MMU 确定本地页表中保存有与待写入数据的虚拟地址对应的本地缓存中的物理地址时，MMU 可以触发 Client 端的 CPU 根据该待写入数据的虚拟地址对应的物理地址，将待写入数据写入到本地缓存的缓存页面中。当 MMU 确定本地页表中不存在与目标数据块的虚拟地址对应的本地缓存中的物理地址时，MMU 会触发 Client 端的 CPU 产生一个缺页错误，即 Page Fault。响应于该 Page Fault，DPU 中的 CPU 从分布式存储系统的存储节点中，获取目标数据块，并缓存到本地缓存页面中。然后，Client 端 CPU 建立待写入数据的虚拟地址与本地缓存之间的映射关系，并根据该映射关系将待写入数据写入到本地缓存页面中。其中，DPU 中的 CPU 响应于 Page Fault 从分布式存储系统的存储节点中，获取目标数据块，并缓存到本地缓存页面的过程可以参照 S306 中关于图 5 的描述，在此不再赘述。

Client 端的 CPU 将待写入数据写入到 Client 端的本地缓存中以后，Client 端的 CPU 会将写入数据的缓存页面标记为脏页面（dirty page），并将其加入到 dirty list 中。然后，Client 端的 mmap 管理软件周期性的将 dirty list 中的 page，写回到分布式存储节点中，使得分布式存储节点中的数据和 Client 端的内存中缓存的数据一致。可以理解的是，mmap 管理软件中保存有全局数据块的 DVA 与全局数据块在目标进程的虚拟地址空间中的虚拟地址（virt）之间的映射关系。因此，mmap 管理软件在将 dirty list 中的 page，写回到分布式存储节点中时，mmap 管理软件可以通过本地页表获取 dirty list 中的 page 对应的目标进程中的虚拟地址。然后，mmap 管理软件可以根据 dirty list 中的 page 对应的目标进程中的虚拟地址，以及 mmap 管理软件中保存的映射关系，将 dirty list 中的 page 中存储的数据写回到分布式存储系统中的存储节点上。

在一个可能的示例中，client 端的 mmap 管理软件获取本地缓存中的脏页面，并调用 mcommit() 发起脏页面同步到分布式存储系统中的存储节点的流程。当脏页面同步到分布式存储系统中的存储节点以后，mcommit() 返回成功。

在另一个可能的示例中，client 端的 mmap 管理软件获取本地缓存中的脏页面，通过调用刷新变化函数 msync()，将脏页面写回到分布式存储系统中的存储节点上。

在一个可能的示例中，针对图 6 所示的分布式存储系统。目标进程向 Client 端的 CPU 发送写请求以后，Client 端的 CPU 需要将该写请求发送给 DPU。DPU 中的 DPU-MMU 根据待写入数据的虚拟地址确定本地页表中是否保存有与该数据的虚拟地址本地缓存中的物理地址。待写入数据的虚拟地址是指“目标进程地址空间中的虚拟地址”。若 DPU-MMU 确定 DPU 中的页表中保存有，与待写入数据的虚拟地址对应的 DPU 的本地缓存中的物理地址。MMU 可以触发 DPU 中的 CPU 根据该待写入数据的虚拟地址对应的物理地址，将待写入数据写入到 DPU 中的本地缓存中的缓存页面中。若 DPU-MMU 确定 DPU 中的页表中未保存与待写入数据的虚拟地址对应的 DPU 的本地缓存中的物理地址，DPU-MMU 触发 DPU 内部的 CPU 产生 Page Fault。响应于该 Page Fault，DPU 中的 Page Fault Handler 从分布式存储系统的存储节点中，获取目标数据块，并缓存到 DPU 的缓存页面中。然后，DPU 中的 CPU 建立待写入数据的虚拟地址与 DPU 中的本地缓存中的缓存页面的映射关系，并根据该映射关系将待写入数据写入到本地缓存页面中。其中，DPU 中的 Page Fault Handler 响应于 Page Fault 从分布式存储系统的存储节点中，获取目标数据块，并缓存到 DPU 的本地缓存页面的过程，可以参照 S306 中关于图 6 的描述，在此不再赘述。

DPU 中的 CPU 将目标进程需要写的的数据写入到 DPU 中的本地缓存中的缓存页面以后，DPU 中的 CPU 会将写入数据的缓存页面标记为脏页面（dirty page），并将其加入到 dirty list 中。然后，GPU 中的 mmap 管理软件会周期性的将 dirty list 中的 page 写回到分布式存储节点中，使得分布式存储节点中的数据和 Client 端的内存中缓存的数据一致。在一个可能的示例中，DPU 中的 mmap 管理软件获取本地缓存中的脏页面，并调用 mcommit() 发起脏页面同步到分布式存储系统中的存储节点的流程。当脏页面同步到分布式存储系统中的存储节点以后，mcommit() 返回成功。在另一个可能的示例中，DPU 中的 mmap 管理软件获取本地缓存中的脏页面，通过调用刷新变化函数 msync()，将脏页面写回到分布式存储系统中的存储节点上。

针对上述实施例例中的方法，本申请实施例还提供了一种传统文件访问路径与全局 mmap 访问路径的比较示意图。参照图 7 可知，传统文件访问路径长，需要经过 VFS，文件系统客户端，文件系统服务端，最终访问 PM，软件栈开销大。而在本申请实施例中，在分布式存储系统中，使用 Distributed mmap

映射远端文件之后，在本地（Client 端）记录文件的 DVA（可记录在进程页表中），本地缓存不命中，通过 Page Fault 访问远端 DVA 空间。NIC 直通访问 PM 介质，Bypass Server 端的软件栈开销，端到端软件栈时延从 150us 降 20us。

在本申请实施例中，针对分布式存储系统，构建分布式存储系统中的全局共享内存池。然后，基于该全局内存池构建全局数据块，并将该全局数据块映射到客户端的进程地址空间中。然后，客户端建立用于缓存分布式系统中的全局数据块的本地缓存，并通过客户端的本地页表，建立进程虚拟地址到缓存页面的映射。及在分布式存储系统中，对存储节点中存储的文件进行 mmap 映射，使得客户端的应用程序可以直接通过 load/store 访问，存储节点中的数据。在分布式存储系统中构建基于全局内存池的全局数据块，使得属于同一个分布式存储系统中的多个客户端可 mmap 同一个全局数据块，实现全局数据块在各个客户端之间的共享访问。以及在分布式存储系统中使用 Distributed mmap 映射远端文件之后，在本地记录文件的 DVA（可记录在进程页表中）。当本地缓存不命中，通过 Page Fault 访问远端 DVA 空间，节省端到端软件栈的时延开销。

示例性的，本申请实施例还提供了一种数据处理装置，请参阅图 8。图 8 是本申请实施例提供的一种数据处理装置的结构示意图。如图 8 所示，该数据处理装置中包括：获取模块 810、处理模块 820、存储模块 830。

存储模块 830 用于存储目标进程运行过程中所需要的部分数据，以及存储模块 830 中还存储有本地页表也可以称之为第一地址映射关系表。本地页表中存储有分布式存储系统中的全局数据块的 DVA 与目标进程中的虚拟地址间的映射关系、以及目标进程中的虚拟地址与数据处理装置中的本地缓存的物理地址间的映射关系。

获取模块 810 用于在目标进程过程中，获取目标数据在目标进程中的虚拟地址，也可以称之为第一虚拟地址。

处理模块 820 用于根据获取的目标数据的虚拟地址，查询本地页表确定本地页表中是否存储有与该目标数据的虚拟地址对应的本地缓存的物理地址。当处理模块 820 确定本地页表中是存储有与该目标数据的虚拟地址对应的本地缓存的物理地址时，处理模块 820 获取该物理地址。

获取模块 810 根据处理模块 820 获取的物理地址，在本地缓存中获取目标数据。

在一个可能的示例中，当处理模块 820 确定本地页表中是存储有与该目标数据的虚拟地址对应的本地缓存的物理地址时，处理模块 820 根据目标数据的虚拟地址查询本地页表，得到目标数据的 DVA。然后，处理模块 810 根据目标数据的 DVA 确定目标数据的 DPA。获取模块根据处理模块得到的目标数据 DPA 从分布式存储系统中获取目标数据。

其中，在数据处理装置中，访问目标进程所需要的目标数据的具体过程，可以参照 S302-S306。在此，不再赘述。

示例性的，本申请实施例中还提供了一种计算设备，该计算设备至少包括处理器和存储器，存储器上存储有程序，处理器该程序时，可以实现图 3 所示的方法中的各个步骤的单元或模块。

图 9 为本申请实施例提供的一种计算设备的结构示意图。如图 9 所示，所述计算设备 900 包括至少一个处理器 910、存储器 920 和通信接口 930。其中，处理器 910、存储器 920 和通信接口 930 通信连接，可以通过有线（例如总线）的方式实现通信连接，也可以通过无线的方式实现通信连接。该通信接口 930 用于接收其他设备发送的数据（例如 DUT 对应的 IR）；存储器 920 存储有计算机指令，处理器 910 执行该计算机指令，执行前述方法实施例中的方法。

应理解，在本申请实施例中，该处理器 910 可以是中央处理单元 CPU，该处理器 910 还可以是其他通用处理器、数字信号处理器（digital signal processor，DSP）、专用集成电路（application specific integrated circuit，ASIC）、现场可编程门阵列（field programmable gate array，FPGA）或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。通用处理器可以是微处理器或者是任何常规的处理器等。

该存储器 920 可以包括只读存储器和随机存取存储器，并向处理器 910 提供指令和数据。存储器 920 还可以包括非易失性随机存取存储器。

该存储器 920 可以是易失性存储器或非易失性存储器，或可包括易失性和非易失性存储器两者。其

中，非易失性存储器可以是只读存储器（read-only memory, ROM）、可编程只读存储器（programmable ROM, PROM）、可擦除可编程只读存储器（erasable PROM, EPROM）、电可擦除可编程只读存储器（electrically EPROM, EEPROM）或闪存。易失性存储器可以是随机存取存储器（random access memory, RAM），其用作外部高速缓存。通过示例性但不是限制性说明，许多形式的 RAM 可用，例如静态随机存取存储器（static RAM, SRAM）、动态随机存取存储器（DRAM）、同步动态随机存取存储器（synchronous DRAM, SDRAM）、双倍数据速率同步动态随机存取存储器（double data rate SDRAM, DDR SDRAM）、增强型同步动态随机存取存储器（enhanced SDRAM, ESDRAM）、同步连接动态随机存取存储器（synchlink DRAM, SLDRAM）和直接内存总线随机存取存储器（direct rambus RAM, DR RAM）。

应理解，根据本申请实施例的计算设备 900 可以执行实现本申请实施例中图 3 所示方法，该方法实现的详细描述参见上文，为了简洁，在此不再赘述。

本申请的实施例提供了一种计算机可读存储介质，其上存储有计算机程序，当所述计算机指令在被处理器执行时，使得上文提及的方法被实现。

本申请的实施例提供了一种芯片，该芯片包括至少一个处理器和接口，所述至少一个处理器通过所述接口确定程序指令或者数据；该至少一个处理器用于执行所述程序指令，以实现上文提及的方法。

本申请的实施例提供了一种计算机程序或计算机程序产品，该计算机程序或计算机程序产品包括指令，当该指令执行时，令计算机执行上文提及的方法。

本申请的实施例中的方法步骤可以通过硬件的方式来实现，也可以由处理器执行软件指令的方式来实现。软件指令可以由相应的软件模块组成，软件模块可以被存放于随机存取存储器（random access memory, RAM）、闪存、只读存储器（read-only memory, ROM）、可编程只读存储器（programmable rom, PROM）、可擦除可编程只读存储器（erasable PROM, EPROM）、电可擦除可编程只读存储器（electrically EPROM, EEPROM）、寄存器、硬盘、移动硬盘、CD-ROM 或者本领域熟知的任何其它形式的存储介质中。一种示例性的存储介质耦合至处理器，从而使处理器能够从该存储介质读取信息，且可向该存储介质写入信息。当然，存储介质也可以是处理器的组成部分。处理器和存储介质可以位于 ASIC 中。

在上述实施例中，可以全部或部分地通过软件、硬件、固件或者其任意组合来实现。当使用软件实现时，可以全部或部分地以计算机程序产品的形式实现。所述计算机程序产品包括一个或多个计算机指令。在计算机上加载和执行所述计算机程序指令时，全部或部分地产生按照本申请实施例所述的流程或功能。所述计算机可以是通用计算机、专用计算机、计算机网络、或者其他可编程装置。所述计算机指令可以存储在计算机可读存储介质中，或者通过所述计算机可读存储介质进行传输。所述计算机指令可以从一个网站站点、计算机、服务器或数据中心通过有线（例如同轴电缆、光纤、数字用户线（DSL））或无线（例如红外、无线、微波等）方式向另一个网站站点、计算机、服务器或数据中心进行传输。所述计算机可读存储介质可以是计算机能够存取的任何可用介质或者是包含一个或多个可用介质集成的服务器、数据中心等数据存储设备。所述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（solid state disk, SSD））等。

可以理解的是，在本申请的实施例中涉及的各种数字编号仅为描述方便进行的区分，并不用来限制本申请的实施例的范围。

权 利 要 求 书

1. 一种数据访问方法，其特征在于，所述方法包括：

在目标进程运行过程中，获取目标数据在所述目标进程中的第一虚拟地址；

基于所述第一虚拟地址，查询第一地址映射关系表，确定本地缓存中是否存储有所述目标数据，所述第一地址映射关系表中包括：第一数据集合中的数据在进程中的虚拟地址和所述第一数据集合中的数据在本地缓存中的物理地址间的对应关系，其中，所述第一数据集合中的数据预先从分布式存储系统中获取得到；

若本地缓存中存储有所述目标数据，基于所述第一虚拟地址，查询所述第一地址映射关系表，得到所述目标数据在本地缓存中的物理地址；

根据所述目标数据在本地缓存中的物理地址，获取所述目标数据。

2. 根据权利要求1所述的方法，其特征在于，所述第一地址映射关系表中还包括：第二数据集合中的数据在进程中的虚拟地址和所述第二数据集合中的数据在所述分布式存储系统中的虚拟地址间的对应关系，所述方法还包括：

若本地缓存中没有存储有所述目标数据，基于所述第一虚拟地址，查询所述第一地址映射关系表，得到所述目标数据在所述分布式存储系统中的第二虚拟地址；

基于所述第二虚拟地址，查询第二地址映射关系表，得到所述目标数据在所述分布式存储系统中的物理地址；所述第二地址映射关系表中包括：在分布式存储系统中存储的第三数据集合中的数据的虚拟地址和所述第三数据集合中的数据在所述分布式存储系统中的物理地址间的对应关系；其中，所述第三数据集合中的数据包括：所述第一数据集合中的数据和所述第二数据集合中的数据；

基于所述目标数据在所述分布式存储系统中的物理地址，从所述分布式存储系统中获取所述目标数据。

3. 根据权利要求2所述的方法，其特征在于，在基于所述目标数据在所述分布式存储系统中的物理地址，从所述分布式存储系统中获取所述目标数据以后，所述方法还包括：

将所述目标数据存储到本地缓存中；

建立所述目标数据在所述目标进程中的第一虚拟地址和所述目标数据在本地缓存中的物理地址间的第一对应关系；

用所述第一对应关系替换所述第一地址映射关系表中的第二对应关系，所述第二对应关系为所述目标数据在所述目标进程中的虚拟地址和所述目标数据在所述分布式存储系统中的虚拟地址间的对应关系。

4. 根据权利要求1所述的方法，其特征在于，所述方法还包括：

在所述目标进程运行过程中，获取待写入数据在所述目标进程中的第三虚拟地址；

基于所述第三虚拟地址，查询第一地址映射关系表，确定本地缓存中是否存储有与第三虚拟地址对应的数据；

若所述本地缓存中存储有与所述第三虚拟地址对应的数据，将所述本地缓存中与所述第三虚拟地址对应的数据替换为所述待写入数据。

5. 一种数据处理装置，其特征在于，包括：

获取模块，用于在目标进程运行过程中，获取目标数据在所述目标进程中的第一虚拟地址；

处理模块，用于根据所述第一虚拟地址，查询第一地址映射关系表，确定本地缓存中是否存储有所述目标数据，所述第一地址映射关系表中包括：第一数据集合中的数据在进程中的虚拟地址和所述第一数据集合中的数据在本地缓存中的物理地址间的对应关系，其中，所述第一数据集合中的数据预先从分布式存储系统中获取得到；若本地缓存中存储有所述目标数据，基于所述第一虚拟地址，查询所述第一地址映射关系表，得到所述目标数据在本地缓存中的物理地址；

所述获取模块，还用于根据所述目标数据在本地缓存中的物理地址，获取所述目标数据。

6. 根据权利要求5所述的装置，其特征在于，所述第一地址映射关系表中还包括：第二数据

集合中的数据在进程中的虚拟地址和所述第二数据集合中的数据在所述分布式存储系统中的虚拟地址间的对应关系，所述处理模块还用于：

若本地缓存中没有存储有所述目标数据，基于所述第一虚拟地址，查询所述第一地址映射关系表，得到所述目标数据在所述分布式存储系统中的第二虚拟地址；根据所述第二虚拟地址，查询第二地址映射关系表，得到所述目标数据在所述分布式存储系统中的物理地址；所述第二地址映射关系表中包括：在分布式存储系统中存储的第三数据集合中的数据的虚拟地址和所述第三数据集合中的数据在所述分布式存储系统中的物理地址间的对应关系；其中，所述第三数据集合中的数据包括：所述第一数据集合中的数据和所述第二数据集合中的数据；

所述获取模块还用于，根据所述目标数据在所述分布式存储系统中的物理地址，从所述分布式存储系统中获取所述目标数据。

7. 根据权利要求6所述的装置，其特征在于，所述处理模块还用于：

将所述目标数据存储到本地缓存中；

建立所述目标数据在所述目标进程中的第一虚拟地址和所述目标数据在本地缓存中的物理地址间的第一对应关系；

用所述第一对应关系替换所述第一地址映射关系表中的第二对应关系，所述第二对应关系为所述目标数据在所述目标进程中的虚拟地址和所述目标数据在所述分布式存储系统中的虚拟地址间的对应关系。

8. 根据权利要求5所述的装置，其特征在于，所述获取模块还用于，在目标进程运行过程中，获取待写入数据在所述目标进程中的第三虚拟地址；

所述处理模块还用于，根据所述第三虚拟地址，查询第一地址映射关系表，确定本地缓存中是否存储有与第三虚拟地址对应的数据；若所述本地缓存中存储有与所述第三虚拟地址对应的数据，将所述本地缓存中与所述第三虚拟地址对应的数据替换为所述待写入数据。

9. 一种计算机可读介质，所述计算机存储介质中存储有指令，当所述指令在计算机上运行时，使得计算机执行如权利要求1-4任一所述的方法。

10. 一种包含指令的计算机程序产品，当所述指令在计算机上运行时，使得所述计算机执行如权利要求1-4任一所述的方法。

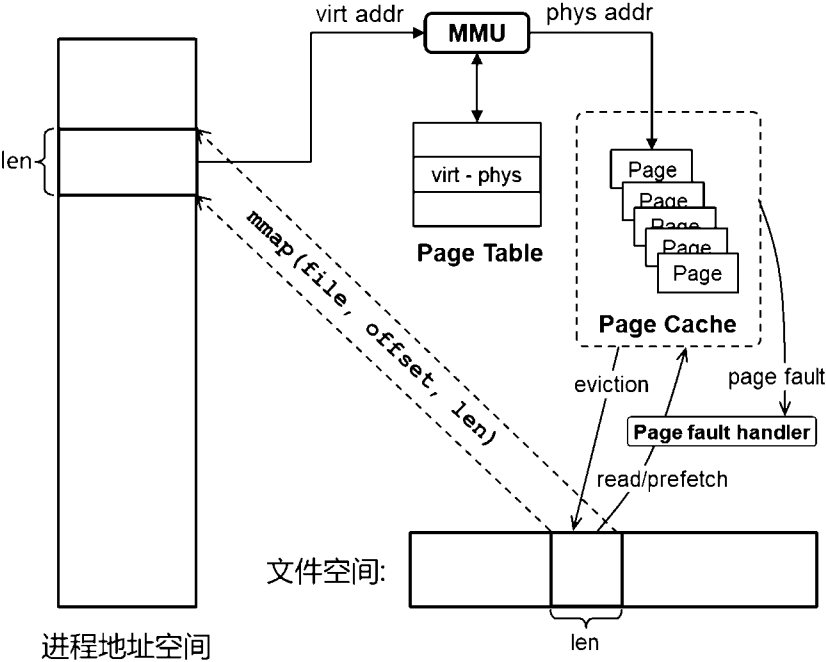


图 1

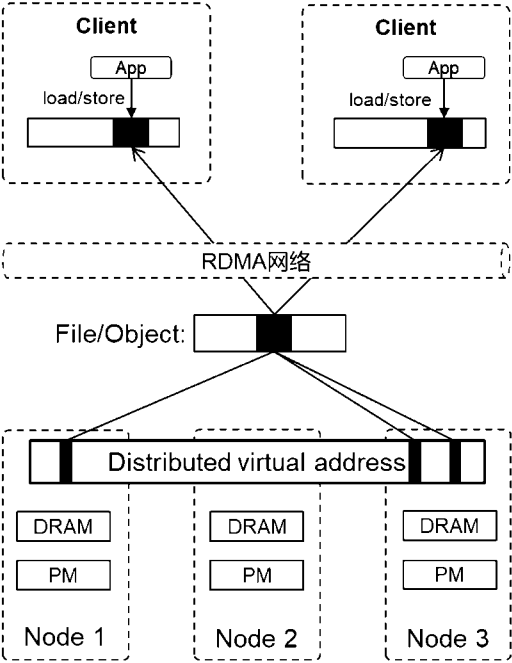


图 2

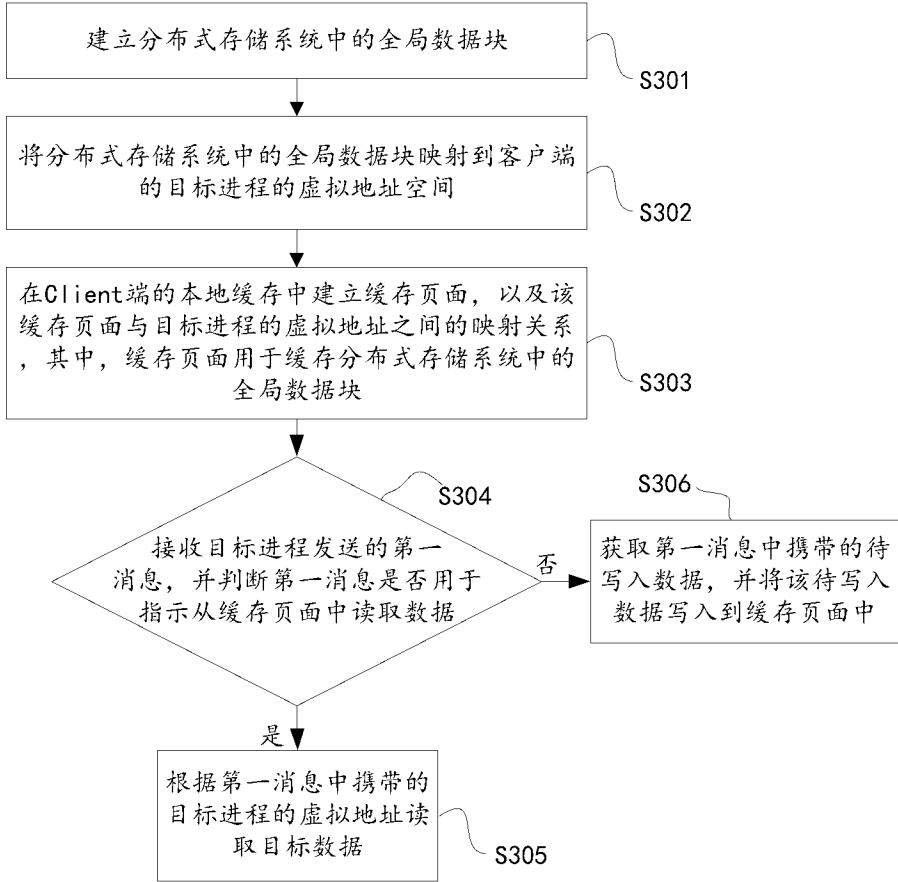


图 3

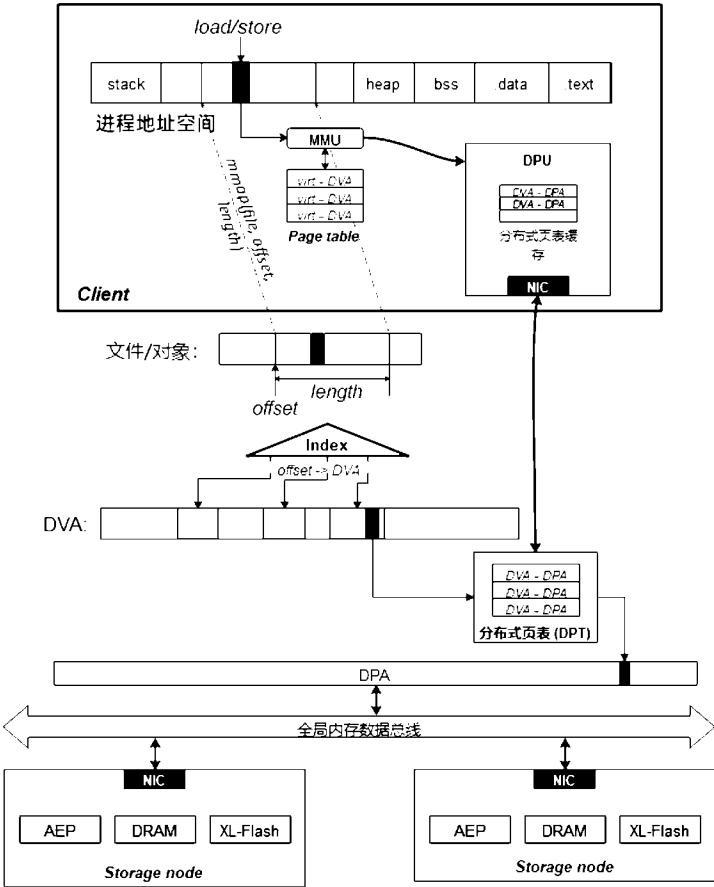


图 4

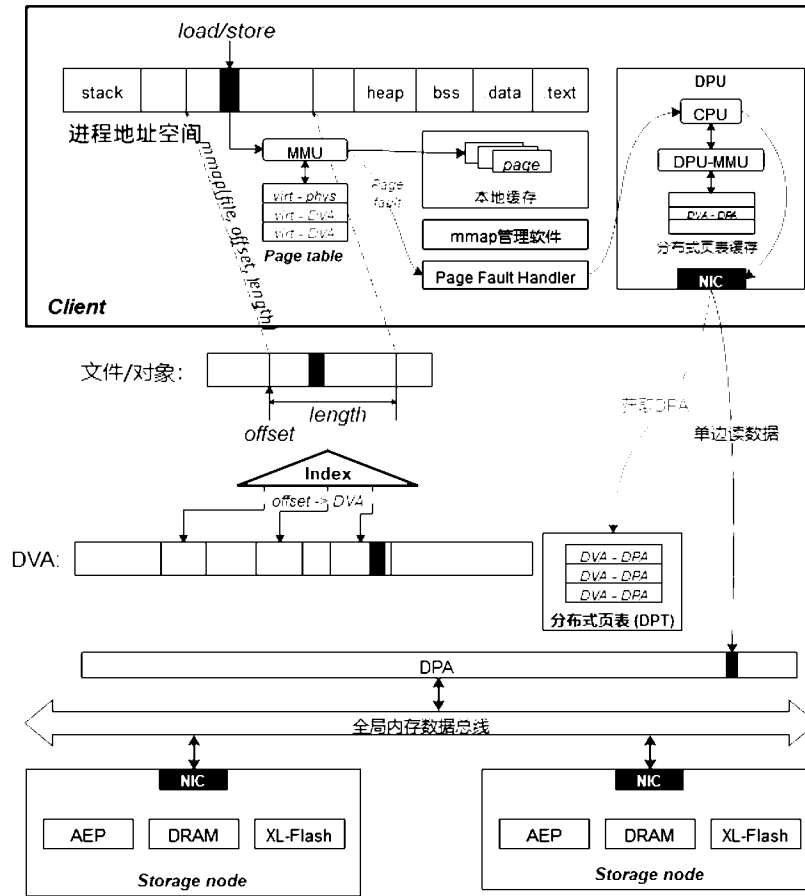


图 5

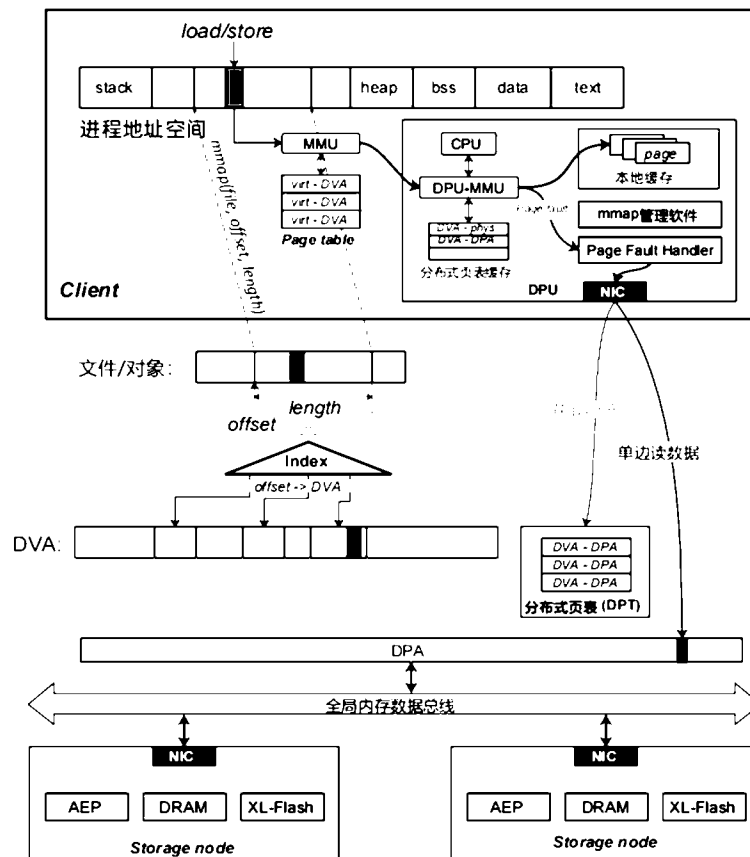


图 6

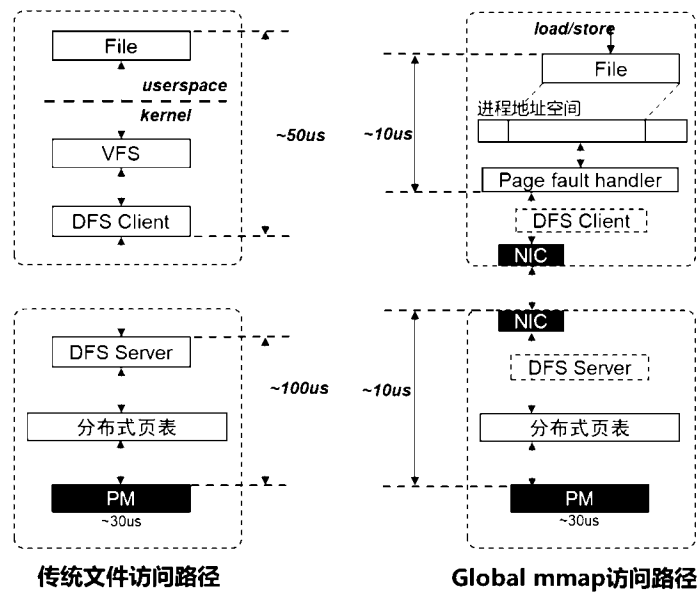


图 7

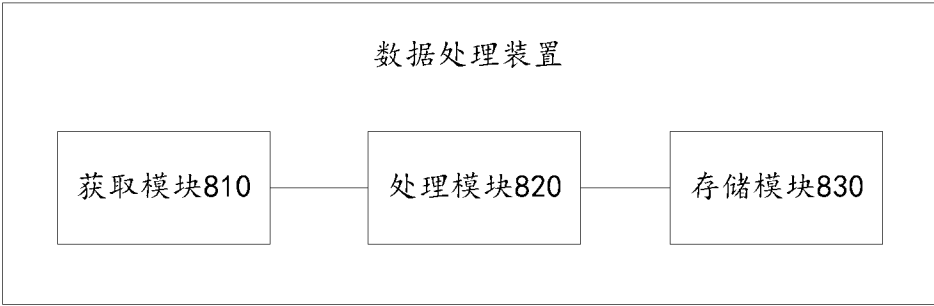


图 8

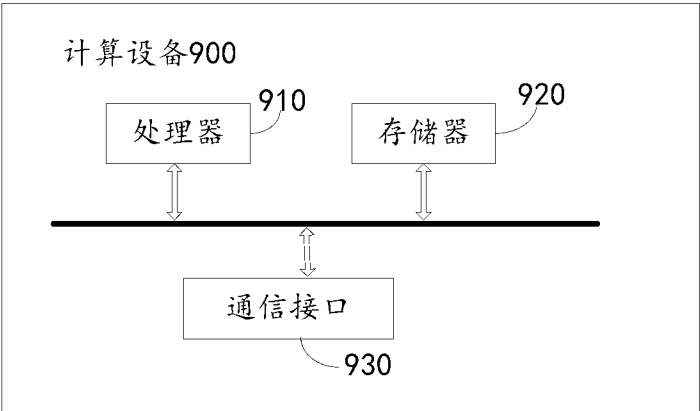


图 9

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2023/100809

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/44(2018.01)i; G06F 12/14(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC:G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNTXT, VEN, WPI, USTXT, EPTXT, JPTXT, WOTXT, IEEE, CNKI: 本地, 缓存, 读取, 访问, 分布式, 进程, 数据, 物理地址, 虚拟地址, DPA, DPT, DVA, mmap, local, cache, read, access, distribut+, data, PA, VA

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US 2018232320 A1 (ADVANCED MICRO DEVICES, INC.) 16 August 2018 (2018-08-16) description, paragraphs 0026-0066, and figures 1-8	1, 4-5, 8-10
Y	CN 114968539 A (HUAWEI TECHNOLOGIES CO., LTD. et al.) 30 August 2022 (2022-08-30) description, paragraphs 0052-0059, and figures 1-10	1, 4-5, 8-10
A	CN 109032963 A (ARM LIMITED) 18 December 2018 (2018-12-18) entire document	1-10
A	CN 113986137 A (INNOGRIT TECHNOLOGY (SHANGHAI) CO., LTD.) 28 January 2022 (2022-01-28) entire document	1-10



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“D” document cited by the applicant in the international application

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search

12 September 2023

Date of mailing of the international search report

20 September 2023

Name and mailing address of the ISA/CN

China National Intellectual Property Administration (ISA/
CN)
China No. 6, Xitucheng Road, Jimenqiao, Haidian District,
Beijing 100088

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2023/100809

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
US	2018232320	A1	16 August 2018	None			
CN	114968539	A	30 August 2022	WO	2022174705	A1	25 August 2022
CN	109032963	A	18 December 2018	GB	2565188	A	06 February 2019
				US	2018357178	A1	13 December 2018
CN	113986137	A	28 January 2022	US	2023137668	A1	04 May 2023

国际检索报告

国际申请号

PCT/CN2023/100809

A. 主题的分类

G06F 9/44(2018.01)i; G06F 12/14(2006.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

IPC:G06F

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称, 和使用的检索词(如使用))

CNTXT, VEN, WPI, USTXT, EPTXT, JPTXT, WOTXT, IEEE, CNKI:本地, 缓存, 读取, 访问, 分布式, 进程, 数据, 物理地址, 虚拟地址, DPA, DPT, DVA, mmap, local, cache, read, access, distribut+, data, PA, VA

C. 相关文件

类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求
Y	US 2018232320 A1 (ADVANCED MICRO DEVICES, INC.) 2018年8月16日 (2018 - 08 - 16) 说明书第0026-0066段, 图1-8	1, 4-5, 8-10
Y	CN 114968539 A (华为技术有限公司 等) 2022年8月30日 (2022 - 08 - 30) 说明书第0052-0059段, 图1-10	1, 4-5, 8-10
A	CN 109032963 A (ARM有限公司) 2018年12月18日 (2018 - 12 - 18) 全文	1-10
A	CN 113986137 A (英韧科技(上海)有限公司) 2022年1月28日 (2022 - 01 - 28) 全文	1-10

☐ 其余文件在C栏的续页中列出。☒ 见同族专利附件。

* 引用文件的具体类型:

“A” 认为不特别相关的表示了现有技术一般状态的文件

“D” 申请人在国际申请中引证的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件(如具体说明的)

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期

2023年9月12日

国际检索报告邮寄日期

2023年9月20日

ISA/CN的名称和邮寄地址

中国国家知识产权局

中国北京市海淀区蓟门桥西土城路6号 100088

授权官员

王平

电话号码 (+86) 010-53961372

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2023/100809

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
US	2018232320	A1	2018年8月16日	无			
CN	114968539	A	2022年8月30日	WO	2022174705	A1	2022年8月25日
CN	109032963	A	2018年12月18日	GB	2565188	A	2019年2月6日
				US	2018357178	A1	2018年12月13日
CN	113986137	A	2022年1月28日	US	2023137668	A1	2023年5月4日