(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 115981875 B
(45)授权公告日 2023.08.25

(21)申请号 202310272752.2
(22)申请日 2023.03.21
(65)同一申请的已公布的文献号
申请公布号 CN 115981875 A
(43)申请公布日 2023.04.18
(73)专利权人 人工智能与数字经济广东省实验室(广州)
地址 510330 广东省广州市海珠区新港东路2429号首层自编051房
(72)发明人 汤庸 陈万德 袁成哲 汤非易 林荣华 毛承洁
(74)专利代理机构 广州科粤专利商标代理有限公司 44001
专利代理师 劳剑东 邓潮彬
(51)Int.Cl.
G06F 9/50(2006.01)
G06F 12/02(2006.01)

(56)对比文件
US 2009193191 A1,2009.07.30
US 2017024140 A1,2017.01.26
CN 107315746 A,2017.11.03
US 2019026476 A1,2019.01.24
US 2017212680 A1,2017.07.27
US 2021389883 A1,2021.12.16
CN 114647383 A,2022.06.21
US 2018075050 A1,2018.03.15
CN 106844507 A,2017.06.13
CN 113312300 A,2021.08.27
CN 105205053 A,2015.12.30
US 2014281132 A1,2014.09.18
CN 105993013 A,2016.10.05
杨帆;李飞;舒继武.安全持久性内存存储研究综述.计算机研究与发展.2020,(05),全文.

审查员 阮圆

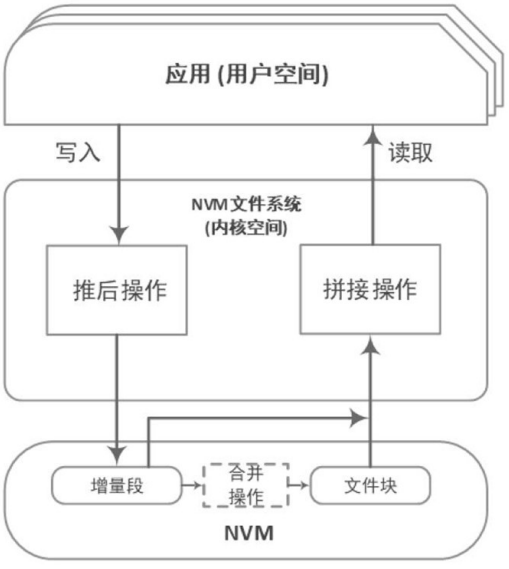
权利要求书3页 说明书11页 附图4页

(54)发明名称

内存存储系统的增量更新方法、装置、设备、介质和产品

(57)摘要

本发明公开了面向非易失性内存存储系统的增量更新方法、装置、设备、介质和产品，包括：根据存储粒度、操作分解、空间分配、数据恢复以及日志结构的考量因素，构建面向非易失性内存存储系统的系统架构；接着接收用户空间的上层应用程序发送的新数据，根据系统架构，对新数据执行推后操作；然后响应于新接收到的读请求，根据系统架构，采用数据聚合操作来对读请求对应的各个数据进行拼接处理，返回读请求对应的请求结果；最后根据系统架构，按照存储顺序对各个增量段进行合并操作，进而消除增量段的重叠。本发明能够减少数据写入放大和更好发挥并发性能，能够缓解存储系统在实时数据分析应用场景的性能瓶颈，可广泛应用于存储系统技术领域。



CN 115981875 B

1.一种面向非易失性内存存储系统的增量更新方法,其特征在于,包括:

根据存储粒度、操作分解、空间分配、数据恢复和日志结构的考量因素,构建面向非易失性内存存储系统的系统架构;

接收用户空间的上层应用程序发送的新数据,根据所述系统架构,对所述新数据执行推后操作;

响应于新接收到的读请求,根据所述系统架构,采用数据聚合操作来对所述读请求对应的各个数据进行拼接处理,返回所述读请求对应的请求结果;

根据所述系统架构,按照存储顺序对各个增量段进行合并操作,进而消除增量段的重叠;

对于所述存储粒度的考量因素,构建好的所述系统架构将每个区块的增量段存储到临时空间;

对于所述操作分解的考量因素,构建好的所述系统架构通过推后操作对增量段进行存储、通过合并操作将增量段进行合并、以及通过拼接操作将增量段与原始数据进行拼接;

对于所述空间分配的考量因素,构建好的所述系统架构按照块粒度和段粒度进行日志维护;

对于所述数据恢复的考量因素,构建好的所述系统架构实时维护一个全局日志,所述全局日志用于记录存储系统的操作过程;

对于所述日志结构的考量因素,构建好的所述系统架构为每个区块维护一个对应的操作日志,每个操作日志建立有对应的索引;

所述接收用户空间的上层应用程序发送的新数据,根据所述系统架构,对所述新数据执行推后操作,具体包括:

接收用户空间的上层应用程序发送的新数据,将所述新数据拆分为逻辑上连续的文件块;

将非易失性内存中的临时存储空间划分为三个不同大小的增量段;

将每个拆分得到的所述文件块存储到对应的增量段中;

为每个增量段创建一个元数据条目,进而对新数据存储的增量段进行更新索引;

所述响应于新接收到的读请求,根据所述系统架构,采用数据聚合操作来对所述读请求对应的各个数据进行拼接处理,返回所述读请求对应的请求结果,具体包括:

响应于新接收到的读请求,通过一个线程读取原始块中的数据;

创建多个线程检索增量段中的数据;

根据操作日志中记录的元数据顺序,将所述原始块和所述增量段中读取到的数据进行移动和拼接,得到一个完整的结果作为所述读请求对应的请求结果;

所述响应于新接收到的读请求,根据所述系统架构,采用数据聚合操作来对所述读请求对应的各个数据进行拼接处理,返回所述读请求对应的请求结果,具体包括:

响应于新接收到的读请求,通过一个线程读取原始块中的数据;

创建多个线程检索增量段中的数据;

根据操作日志中记录的元数据顺序,将所述原始块和所述增量段中读取到的数据进行移动和拼接,得到一个完整的结果作为所述读请求对应的请求结果;

所述根据所述系统架构,按照存储顺序对各个增量段进行合并操作,进而消除增量段

的重叠,包括:

将最近访问的当前块记录到第一优先级列表中;当所述当前块的区段数超过预设最大数量的一半时,将所述当前块记录到第二优先级列表中;

根据所述第一优先级列表和所述第二优先级列表的操作日志,通过存储系统执行合并处理;其中,所述合并处理的执行周期根据数据的读写频率动态调整;

在所述合并处理的过程中,存储系统跳过重叠增量段。

2.根据权利要求1所述的面向非易失性内存存储系统的增量更新方法,其特征在于,所述根据操作日志中记录的元数据顺序,将所述原始块和所述增量段中读取到的数据进行移动和拼接,得到一个完整的结果作为所述读请求对应的请求结果,包括:

根据操作日志中记录的元数据顺序,使用去重叠策略检测和跳过重叠数据的方法,以顺序或反序的方向将所述原始块和所述增量段中读取到的数据进行拼接,得到一个完整的结果作为所述读请求对应的请求结果。

3.一种面向非易失性内存存储系统的增量更新装置,其特征在于,包括:

第一模块,用于根据存储粒度、操作分解、空间分配、数据恢复以及日志结构的考量因素,构建面向非易失性内存存储系统的系统架构;

第二模块,用于接收用户空间的上层应用程序发送的新数据,根据所述系统架构,对所述新数据执行推后操作;

第三模块,用于响应于新接收到的读请求,根据所述系统架构,采用数据聚合操作来对所述读请求对应的各个数据进行拼接处理,返回所述读请求对应的请求结果;以及,

第四模块,用于根据所述系统架构,按照存储顺序对各个增量段进行合并操作,进而消除增量段的重叠;

对于所述存储粒度的考量因素,构建好的所述系统架构将每个区块的增量段存储到临时空间;

对于所述操作分解的考量因素,构建好的所述系统架构通过推后操作对增量段进行存储、通过合并操作将增量段进行合并、以及通过拼接操作将增量段与原始数据进行拼接;

对于所述空间分配的考量因素,构建好的所述系统架构按照块粒度和段粒度进行日志维护;

对于所述数据恢复的考量因素,构建好的所述系统架构实时维护一个全局日志,所述全局日志用于记录存储系统的操作过程;

对于所述日志结构的考量因素,构建好的所述系统架构为每个区块维护一个对应的操作日志,每个操作日志建立有对应的索引;

所述接收用户空间的上层应用程序发送的新数据,根据所述系统架构,对所述新数据执行推后操作,具体包括:

接收用户空间的上层应用程序发送的新数据,将所述新数据拆分为逻辑上连续的文件块;

将非易失性内存中的临时存储空间划分为三个不同大小的增量段;

将每个拆分得到的所述文件块存储到对应的增量段中;

为每个增量段创建一个元数据条目,进而对新数据存储的增量段进行更新索引;

所述响应于新接收到的读请求,根据所述系统架构,采用数据聚合操作来对所述读请

求对应的各个数据进行拼接处理，返回所述读请求对应的请求结果，具体包括：

响应于新接收到的读请求，通过一个线程读取原始块中的数据；

创建多个线程检索增量段中的数据；

根据操作日志中记录的元数据顺序，将所述原始块和所述增量段中读取到的数据进行移动和拼接，得到一个完整的结果作为所述读请求对应的请求结果；

所述响应于新接收到的读请求，根据所述系统架构，采用数据聚合操作来对所述读请求对应的各个数据进行拼接处理，返回所述读请求对应的请求结果，具体包括：

响应于新接收到的读请求，通过一个线程读取原始块中的数据；

创建多个线程检索增量段中的数据；

根据操作日志中记录的元数据顺序，将所述原始块和所述增量段中读取到的数据进行移动和拼接，得到一个完整的结果作为所述读请求对应的请求结果；

所述根据所述系统架构，按照存储顺序对各个增量段进行合并操作，进而消除增量段的重叠，包括：

将最近访问的当前块记录到第一优先级列表中；当所述当前块的区段数超过预设最大数量的一半时，将所述当前块记录到第二优先级列表中；

根据所述第一优先级列表和所述第二优先级列表的操作日志，通过存储系统执行合并处理；其中，所述合并处理的执行周期根据数据的读写频率动态调整；

在所述合并处理的过程中，存储系统跳过重叠增量段。

4.一种电子设备，其特征在于，所述电子设备包括处理器和存储器，所述存储器中存储有至少一条指令或至少一段程序，所述至少一条指令或所述至少一段程序由所述处理器加载并执行，以实现如权利要求1至2任一所述的面向非易失性内存存储系统的增量更新方法。

5.一种计算机可读存储介质，其特征在于，所述存储介质中存储有至少一条指令或至少一段程序，所述至少一条指令或所述至少一段程序由处理器加载并执行以实现权利要求1至2任一所述的面向非易失性内存存储系统的增量更新方法。

内存存储系统的增量更新方法、装置、设备、介质和产品

技术领域

[0001] 本发明涉及存储系统技术领域，尤其是一种内存存储系统的增量更新方法、装置、设备、介质和产品。

背景技术

[0002] 网络技术的迅速发展，大幅度改善了数据传输的带宽和延迟，促进了应用软件生态的繁荣。然而，大量的应用也对边缘服务器的存储性能提出了更高要求，包括存储空间、读写带宽、持久化存储等方面。特别是，人工智能应用中的实时数据分析任务可能会在终端用户、边缘服务器和云端之间产生密集的数据流，需要边缘服务器承担大量数据写入任务。幸运的是，非易失性内存，具有与DRAM同一数量级的性能和类似块设备的持久性存储能力，改变了传统的存储架构，为边缘服务器满足上述的更高要求提供了可能。根据边缘计算研究表明，边缘节点的任务依赖存储系统的持久化效率。更糟糕的是，在非易失性内存设备和DRAM设备之间仍然存在性能差距，尤其在写性能上。

[0003] 考虑通用场景的非易失性内存存储系统如NOVA、SplitFS和Libnvmmio等，使用写时拷贝(Copy-on-Write，CoW)或日志Journaling(也称为logging)机制来更新数据，提供了较好的读取性能。然而，写时拷贝和日志机制都采用了备份数据后再修改的理念，存在重复写入数据的问题，从而限制了设备的写入性能。

[0004] 现有的数据存储系统更新方案，写时拷贝机制和日志机制通过备份数据的方式实现系统意外崩溃后的数据恢复，对读取性能更加友好。而增量更新机制，则将新数据直接存到临时空间或者日志中，有效地减少即时的写入放大，但造成数据的碎片化存储，从而影响读取性能。非易失性内存有读写不对称问题，在读写比例平衡或写比例更高的写密集环境中，更需要系统软件从写入过程减少开销的角度，减少数据的写入放大。因此，在实时数据分析场景中，增量更新的方式更适用于非易失性内存，能够缓解它相对于自身读取性能的写入性能不足问题。

[0005] 然而，现有的增量更新机制基于传统的存储设备而设计，采用了集中式结构和批量优化写入顺序的策略，并不能有效利用非易失性内存的特性，限制了并发写入的性能；同时，增量过多造成的碎片化存储也降低了存储系统的读取性能，传统的读取缓存策略也并不适用于非易失性内存存储系统。

发明内容

[0006] 针对现有技术中的不足，本发明实施例提供一种内存存储系统的增量更新方法、装置、设备、介质和产品、装置、设备和介质，以减少数据写入放大和更好地发挥并发性能。

[0007] 为实现上述目的，本发明可以采用如下技术方案：

[0008] 第一方面，本发明实施例提供了面向非易失性内存存储系统的增量更新方法，包括：

[0009] 根据存储粒度、操作分解、空间分配、数据恢复以及日志结构的考量因素，构建面

向非易失性内存存储系统的系统架构；

[0010] 接收用户空间的上层应用程序发送的新数据，根据所述系统架构，对所述新数据执行推后操作；

[0011] 响应于新接收到的读请求，根据所述系统架构，采用数据聚合操作来对所述读请求对应的各个数据进行拼接处理，返回所述读请求对应的请求结果；

[0012] 根据所述系统架构，按照存储顺序对各个增量段进行合并操作，进而消除增量段的重叠。

[0013] 可选地，对于所述存储粒度的考量因素，构建好的所述系统架构将每个区块的增量段存储到临时空间；

[0014] 对于所述操作分解的考量因素，构建好的所述系统架构通过推后操作对增量段进行存储、通过合并操作将增量段进行合并、以及通过拼接操作将增量段与原始数据进行拼接；

[0015] 对于所述空间分配的考量因素，构建好的所述系统架构按照块粒度和段粒度进行日志维护；

[0016] 对于所述数据恢复的考量因素，构建好的所述系统架构实时维护一个全局日志，所述全局日志用于记录存储系统的操作过程；

[0017] 对于所述日志结构的考量因素，构建好的所述系统架构为每个区块维护一个对应的操作日志，每个操作日志建立有对应的索引。

[0018] 可选地，所述接收用户空间的上层应用程序发送的新数据，根据所述系统架构，对所述新数据执行推后操作，包括：

[0019] 接收用户空间的上层应用程序发送的新数据，将所述新数据拆分为逻辑上连续的文件块；

[0020] 将非易失性内存中的临时存储空间划分为三个不同大小的增量段；

[0021] 将每个拆分得到的所述文件块存储到对应的增量段中；

[0022] 为每个增量段创建一个元数据条目，进而对新数据存储的增量段进行更新索引。

[0023] 可选地，所述响应于新接收到的读请求，根据所述系统架构，采用数据聚合操作来对所述读请求对应的各个数据进行拼接处理，返回所述读请求对应的请求结果，包括：

[0024] 响应于新接收到的读请求，通过一个线程读取原始块中的数据；

[0025] 创建多个线程检索增量段中的数据；

[0026] 根据操作日志中记录的元数据顺序，将所述原始块和所述增量段中读取到的数据进行移动和拼接，得到一个完整的结果作为所述读请求对应的请求结果。

[0027] 可选地，所述根据操作日志中记录的元数据顺序，将所述原始块和所述增量段中读取到的数据进行移动和拼接，得到一个完整的结果作为所述读请求对应的请求结果，包括：

[0028] 根据操作日志中记录的元数据顺序，使用去重叠策略检测和跳过重叠数据的方法，以顺序或反序的方向将所述原始块和所述增量段中读取到的数据进行拼接，得到一个完整的结果作为所述读请求对应的请求结果。

[0029] 可选地，所述根据所述系统架构，按照存储顺序对各个增量段进行合并操作，进而消除增量段的重叠，包括：

[0030] 将最近访问的当前块记录到第一个优先级列表中;当所述当前块的区段数超过预设最大数量的一半时,将所述当前块记录到第二个优先级列表中;

[0031] 根据所述第一优先级列表和所述第二优先级列表的操作日志,通过存储系统执行合并处理;其中,所述合并处理的执行周期根据数据的读写频率动态调整;

[0032] 在所述合并处理的过程中,存储系统跳过重叠增量段。

[0033] 第二方面,本发明实施例还提供了一种面向非易失性内存存储系统的增量更新装置,包括:

[0034] 第一模块,用于根据存储粒度、操作分解、空间分配、数据恢复以及日志结构的考量因素,构建面向非易失性内存存储系统的系统架构;

[0035] 第二模块,用于接收用户空间的上层应用程序发送的新数据,根据所述系统架构,对所述新数据执行推后操作;

[0036] 第三模块,用于响应于新接收到的读请求,根据所述系统架构,采用数据聚合操作来对所述读请求对应的各个数据进行拼接处理,返回所述读请求对应的请求结果;

[0037] 第四模块,用于根据所述系统架构,按照存储顺序对各个增量段进行合并操作,进而消除增量段的重叠。

[0038] 第三方面,本发明实施例还提供了一种电子设备,包括处理器以及存储器;

[0039] 所述存储器用于存储程序;

[0040] 所述处理器执行所述程序实现如前面所述的方法。

[0041] 第四方面,本发明实施例还提供了一种计算机可读存储介质,所述存储介质存储有程序,所述程序被处理器执行实现如前面所述的方法。

[0042] 第五方面,本发明实施例还提供了一种计算机程序产品或计算机程序,该计算机程序产品或计算机程序包括计算机指令,该计算机指令存储在计算机可读存储介质中。计算机设备的处理器可以从计算机可读存储介质读取该计算机指令,处理器执行该计算机指令,使得该计算机设备执行前面的方法。

[0043] 本发明与现有技术相比,其有益效果在于:本发明的实施例首先根据存储粒度、操作分解、空间分配、数据恢复以及日志结构的考量因素,构建面向非易失性内存存储系统的系统架构;接着接收用户空间的上层应用程序发送的新数据,根据所述系统架构,对所述新数据执行推后操作;然后响应于新接收到的读请求,根据所述系统架构,采用数据聚合操作来对所述读请求对应的各个数据进行拼接处理,返回所述读请求对应的请求结果;最后根据所述系统架构,按照存储顺序对各个增量段进行合并操作,进而消除增量段的重叠。本发明提供了一种能够减少数据写入放大和更好发挥并发性能的机制,能够缓解存储系统在实时数据分析应用场景的性能瓶颈。

附图说明

[0044] 为了更清楚地说明本发明实施例中的技术方案,下面将对实施例中所需要使用的附图进行简单的介绍,显而易见地,下面描述中的附图仅仅是本申请的一些实施例,对于本领域普通技术人员来讲,在不付出创造性劳动的前提下,还可以根据这些附图获得其他的附图。

[0045] 图1为本发明实施例提供的PostMerge结构的架构图;

[0046] 图2为本发明实施例提供的推后操作的流程图；

[0047] 图3为本发明实施例提供的拼接操作的流程图；

[0048] 图4为本发明实施例提供的合并操作的流程图；

[0049] 图5为本发明实施例提供的具体应用场景下的实施流程图。

具体实施方式

[0050] 实施例：

[0051] 下面将结合本发明实施例中的附图，对本发明实施例中的技术方案进行清楚、完整的描述，显然，所描述的实施例仅是本申请一部分实施例，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，都属于本申请保护的范围。

[0052] 针对现有技术存在的问题，本发明旨在为非易失性内存存储系统设计一项能够减少数据写入放大和更好发挥并发性能的机制，以缓解存储系统在实时数据分析应用场景的性能瓶颈。具体地，本发明实施例的一方面提供了面向非易失性内存存储系统的增量更新方法，包括：

[0053] 根据存储粒度、操作分解、空间分配、数据恢复以及日志结构的考量因素，构建面向非易失性内存存储系统的系统架构；

[0054] 接收用户空间的上层应用程序发送的新数据，根据所述系统架构，对所述新数据执行推后操作；

[0055] 响应于新接收到的读请求，根据所述系统架构，采用数据聚合操作来对所述读请求对应的各个数据进行拼接处理，返回所述读请求对应的请求结果；

[0056] 根据所述系统架构，按照存储顺序对各个增量段进行合并操作，进而消除增量段的重叠。

[0057] 可选地，对于所述存储粒度的考量因素，构建好的所述系统架构将每个区块的增量段存储到临时空间；

[0058] 对于所述操作分解的考量因素，构建好的所述系统架构通过推后操作对增量段进行存储、通过合并操作将增量段进行合并、以及通过拼接操作将增量段与原始数据进行拼接；

[0059] 对于所述空间分配的考量因素，构建好的所述系统架构按照块粒度和段粒度进行日志维护；

[0060] 对于所述数据恢复的考量因素，构建好的所述系统架构实时维护一个全局日志，所述全局日志用于记录存储系统的操作过程；

[0061] 对于所述日志结构的考量因素，构建好的所述系统架构为每个区块维护一个对应的操作日志，每个操作日志建立有对应的索引。

[0062] 可选地，所述接收用户空间的上层应用程序发送的新数据，根据所述系统架构，对所述新数据执行推后操作，包括：

[0063] 接收用户空间的上层应用程序发送的新数据，将所述新数据拆分为逻辑上连续的文件块；

[0064] 将非易失性内存中的临时存储空间划分为三个不同大小的增量段；

[0065] 将每个拆分得到的所述文件块存储到对应的增量段中；

[0066] 为每个增量段创建一个元数据条目，进而对新数据存储的增量段进行更新索引。

[0067] 可选地，所述响应于新接收到的读请求，根据所述系统架构，采用数据聚合操作来对所述读请求对应的各个数据进行拼接处理，返回所述读请求对应的请求结果，包括：

[0068] 响应于新接收到的读请求，通过一个线程读取原始块中的数据；

[0069] 创建多个线程检索增量段中的数据；

[0070] 根据操作日志中记录的元数据顺序，将所述原始块和所述增量段中读取到的数据进行移动和拼接，得到一个完整的结果作为所述读请求对应的请求结果。

[0071] 可选地，所述根据操作日志中记录的元数据顺序，将所述原始块和所述增量段中读取到的数据进行移动和拼接，得到一个完整的结果作为所述读请求对应的请求结果，包括：

[0072] 根据操作日志中记录的元数据顺序，使用去重叠策略检测和跳过重叠数据的方法，以顺序或反序的方向将所述原始块和所述增量段中读取到的数据进行拼接，得到一个完整的结果作为所述读请求对应的请求结果。

[0073] 可选地，所述根据所述系统架构，按照存储顺序对各个增量段进行合并操作，进而消除增量段的重叠，包括：

[0074] 将最近访问的当前块记录到第一个优先级列表中；当所述当前块的区段数超过预设最大数量的一半时，将所述当前块记录到第二个优先级列表中；

[0075] 根据所述第一优先级列表和所述第二优先级列表的操作日志，通过存储系统执行合并处理；其中，所述合并处理的执行周期根据数据的读写频率动态调整；

[0076] 在所述合并处理的过程中，存储系统跳过重叠增量段。

[0077] 本发明实施例的另一方面还提供了一种面向非易失性内存存储系统的增量更新装置，包括：

[0078] 第一模块，用于根据存储粒度、操作分解、空间分配、数据恢复以及日志结构的考量因素，构建面向非易失性内存存储系统的系统架构；

[0079] 第二模块，用于接收用户空间的上层应用程序发送的新数据，根据所述系统架构，对所述新数据执行推后操作；

[0080] 第三模块，用于响应于新接收到的读请求，根据所述系统架构，采用数据聚合操作来对所述读请求对应的各个数据进行拼接处理，返回所述读请求对应的请求结果；

[0081] 第四模块，用于根据所述系统架构，按照存储顺序对各个增量段进行合并操作，进而消除增量段的重叠。

[0082] 本发明实施例的另一方面还提供了一种电子设备，包括处理器以及存储器；

[0083] 所述存储器用于存储程序；

[0084] 所述处理器执行所述程序实现如前面所述的方法。

[0085] 本发明实施例的另一方面还提供了一种计算机可读存储介质，所述存储介质存储有程序，所述程序被处理器执行实现如前面所述的方法。

[0086] 本发明实施例还公开了一种计算机程序产品或计算机程序，该计算机程序产品或计算机程序包括计算机指令，该计算机指令存储在计算机可读存储介质中。计算机设备的处理器可以从计算机可读存储介质读取该计算机指令，处理器执行该计算机指令，使得该

计算机设备执行前面的方法。

[0087] 下面结合说明书附图,对本发明的具体实施过程进行详细描述:

[0088] 本发明提出面向非易失性内存的增量更新机制PostMerge。作为一种更新机制,PostMerge旨在减少面向非易失性内存的文件系统在写密集型场景下的写放大。PostMerge的执行过程可以分解为推后(postponing)、拼接(splicing)和合并(merging)三个操作,下面为PostMerge设计的详细介绍。

[0089] 存储粒度:随着存储设备性能的大幅提升,科学界和工业界都呼吁使用更大的页面(2MB)来提高文件系统的存储效率。较大的块有利于减少操作的开销,如文件系统元数据管理、数据块寻址和空间回收等。然而,更大的块也放大了有二次写入的负面影响。在PostMerge中,由于一个区块的增量总是该区块的一个片段,本发明把增量称为一个增量段。为了最小化边缘服务器的写入放大,PostMerge只将增量段存储到临时空间,并以更小的粒度动态分配空间。

[0090] 操作分解:写时拷贝和日志都是通过预先备份数据来维持数据的一致性,所以它们更偏向于读操作。相比之下,PostMerge在提高写入性能方面更加积极,在持久化过程中没有数据备份。如图1所示,PostMerge的操作包括推后、拼接和合并。推后操作只存储增量段,然后按优先级定期将它们通过合并操作合并到原始路径中。当未合并的增量段所持有的数据被请求读取时,PostMerge将这些增量段与原始数据块通过拼接操作拼接返回结果。

[0091] 空间分配:PostMerge按照一个块和一个增量段的粒度来维护日志,即BLog和SLog。对于每个数据块,上述日志在非易失性内存中被预先分配,以减少写入性的抖动,提高响应速度。在增量段的空间分配方面,PostMerge根据更新的大小当存在小增量段空间请求时,每次为一个块的更新分配小增量段的空间,避免复的空间分配开销。具体来说,一个增量有四种长度,分别是256B、1KB和4KB。此外,在文件系统初始化过程中,一些元数据需要被保留和格式化,以便在外中断后可以恢复数据。

[0092] 数据恢复:在崩溃重启期间,系统需要恢复到与崩溃前相同的状态,因此有必要在更新机制中确保崩溃的一致性。为了进一步减少数据传输的开销,PostMerge使用直接内存拷贝方式,将数据从非易失性内存(内核空间)复制到DRAM(用户空间)。由于PostMerge的传输不会覆盖原始数据,PostMerge的崩溃恢复很容易实现。换句话说,PostMerge可以简单地通过扫描其操作日志来回滚和重做中断的推迟和合并。为此,本发明为PostMerge维护了一个全局日志,该日志记录了其写操作的整个过程。在空间回收方面,合并在复制了新的数据后释放了增量段空间。由于增量段的长度是在创建时确定的,所以不存在不可回收的碎片。

[0093] 日志结构:为了记录块上的操作,PostMerge为每个块维护一个操作日志,称为BLog。BLog与块节点(blocknode)一一对应,因此BLog可以嵌入blocknode中。除了记录块的元数据之外,BLog还建立了许多SLog索引,这些SLog是为每个增量段维护的日志。有了BLog和SLog,块和增量段的操作都可以被记录下来,从而方便拼接和合并。由于BLog和SLog都只生成一次并采用就地更新的方式,所以PostMerge的元数据更新没有影响并发执行的读/写锁。因此,blocknode、BLog和SLog形成了自上而下的层级结构,从而允许PostMerge以不同的粒度管理元数据。

[0094] 推后操作:虽然非易失性内存在性能的数量级上优于传统固态盘和机械硬盘,但它具有读写不对称问题,并且它的短时延特点导致它对写入开销更为敏感。因此,如果最终

的新数据与原始数据能够推迟合并，将合并工作交给后台进程，先持久化数据到临时空间就返回响应完成，那么系统的写吞吐量将会大大提高，从而为写密集型场景提升整体的性能。如图2所示，新数据来自位于DRAM中用户空间的上层应用程序，然后被拆分为逻辑上连续的文件块。在推后操作期间，新数据将持久化存储为段。在新数据的开始可能有一个块偏移，它的终止位置可能没有块对齐，所以段的长度不是固定的。一个写了多次的块在合并之前会有多个增量段。当一个增量段被持久化到非易失性内存时，相应的SLog和BLog也会被更新。出于快速持久化的目的，在推后操作中，PostMerge将这些新数据临时存储在增量段中。为了存储不同大小的增量，PostMerge将非易失性内存中的临时存储空间划分为三个不同大小的许多增量段(256B、1KB和4KB)。同时，为每个增量段创建一个元数据条目(SLog)。使用三个固定大小增量段是为了对齐地址和简化索引，而不会浪费大量空间。每个增量段的元数据条目是64B，所以条目的更新是细粒度的和原子性的。在一个增量段被成功写入后，PostMerge会更新相应的SLog和BLog。总的来说，推后操作存储新数据是分段的，有轻微的写放大(由元数据更新引起)，但没有像写时拷贝那样备份原始数据的开销。

[0095] 拼接操作：当一个读请求到来时，系统需要返回正确的结果。但是，请求的数据分散在原始块和增量段中。因此，拼接操作使用一个线程读取原始块中的数据，并创建多个线程检索增量段中的数据。最后，将上述得到的数据在DRAM中拼接成一个完整的结果，从而与读取请求相对应。推后操作存储的新数据分散在各增量段中。因此，当读取请求到来时，无法仅通过原始路径读取完整的数据。换句话说，原始路径中的数据可能已经过时。为了读取最新的数据并补偿读取的性能损失，PostMerge采用数据聚合操作来立即响应读取请求。

[0096] 如图3所示，PostMerge使用多线程方法来加速所请求数据的拼接。具体来说，PostMerge使用一个线程来读取原始路径中的原始数据，并创建多个子线程来读取分散在各个增量段中的增量数据。上述线程读取所需的数据后，在适当的位置移动和拼接数据。其中，数据是严格按照BLog和日志中的元数据顺序移动和拼接的，拼接顺序和拼接位置的策略如下所述。

[0097] PostMerge在拼接过程中使用去重叠策略检测和跳过重叠数据。跳过重叠增量段后，可以顺序或反向执行拼接。按照历史写入的顺序拼接增量段是一种简单的方法，但是如果增量段之间有重叠，那么一些增量段的拼接就是多余的。当某一段的大小或所有相关段的数量很大时，重叠现象的写入放大不能被忽略。但是，频繁读取的区段已经在先前的读取操作中被拼接操作拼接到相应的块中，因此区段中的重叠对拼接的影响有限。相反，由于数据重叠的存在，以历史写入的相反顺序拼接区段可能导致过时数据被读取，这导致重叠检测是必要的。在最好的情况下，当请求的数据在最新段时，反向拼接可以最快响应。然而，在最坏的情况下，这使得读取请求相当昂贵，所以PostMerge选择更简单和更有效的顺序拼接。

[0098] 基于混合存储器架构，拼接操作的拼接位置可以是在DRAM或非易失性内存中。如果合并后拼接非易失性内存中的数据，拼接的结果可以暂时保留供下次读取。假设同一条数据被多次读取，非易失性内存中拼接的优势会进一步放大。然而，在非易失性内存中拼接意味着昂贵的写入；因此，读请求不能及时得到响应。相反，DRAM的写性能比傲腾持久性内存高一个数量级，最差情况下的读延迟在可接受的范围内。因此，本实施例设计选择在DRAM中进行拼接操作。

[0099] 在满足一定条件的情况下,系统根据优先级列表在后台进行合并操作。在即时执行的推后操作后,新数据被递增地保存到非易失性内存中的区段中。虽然通过拼接操作拼接来正确读取数据是可行的,但当有许多增量时,这可能会变得更加耗时。此外,随着增量段数量的增加,它们占据了越来越大的存储空间,从而降低了非易失性内存的存储密度。因此,PostMerge需要在后台将增量段中的新数据合并到相应的块中,以避免多余区段带来的负面影响。

[0100] 合并操作:如图4所示,在局部性原则在混合存储器体系结构中仍然有效,换句话说,已经被访问的数据可能会被再次请求。为了利用上述原理,PostMerge标记最近拼接的块。此外,具有大量截面的块也被标记。当增量段的数量很大时,增量段之间的逻辑区域经常重叠。为了解决这个问题,本章引入了一种去重叠策略来消除增量段之间的重叠。此外,增量段按照存储它们的顺序进行合并。被标记的块被赋予一定的优先级,以便PostMerge可以根据它们的优先级将这些增量段合并到其中。

[0101] 具体来说,最近访问的块被记录到优先级列表MLog1中。如果块的区段数超过默认最大数量的一半,则该块将被记录到另一个优先级列表MLog2中。PostMerge基于以上两个日志在后台执行合并,执行周期根据读写频率动态变化。默认情况下,在1GB写入或128次访问请求后,它会被处理一次,每当MLog1或MLog2增长时,该周期减半。此外,当其中一个日志已满时,PostMerge会强制进行合并,直到两个日志都为空。平均来说,每分钟在后台执行一次合并操作。重复多次的推后操作可能导致单个块存在多个增量段,并且这些增量段之间可能存在重叠数据。由于合并过程也会会覆盖重叠的数据,因此PostMerge跳过重叠增量段以减少软件开销。

[0102] 下面以基于高校知识图谱的学者推荐案例为例,详细描述本发明的存储系统的增量更新方法的应用过程:

[0103] 在人工智能应用中的实时数据分析任务可能会在终端用户、边缘服务器和云端之间产生密集的数据流,需要边缘服务器承担大量数据写入任务,并采用了重删系统,使得传统的轻量级日志更新机制失效。

[0104] 本发明以具体的基于高校知识图谱的学者推荐案例,作为实时数据分析任务场景的代表,展示本发明的面向非易失性内存内存系统的增量更新机制的整体使用过程。

[0105] 如图5所示,首先,该案例中存储工作负载可以分为四个部分:GB级别数据采集、知识融合和知识推理、模型训练和模型验证,以及模型转发。基于高校知识图谱的学者推荐案例负载要比传统的应用负载复杂,它在不同阶段有不同的性能要求。在数据收集阶段,从用户环境中收集原始数据,并实时进行一些数据清理,所以这个阶段是写密集型,对存储系统的存储空间与写带宽都存在要求。在知识推理阶段,虽然I/O性能不再是核心性能,但系统仍然对延迟敏感。训练阶段和验证阶段是计算密集型的,也有大量的读请求和写请求。此外,由训练过的模型推理出的新数据需要被存储并整合到训练集中,以便不断改进和重新训练模型。

[0106] 在GB级别数据采集阶段,系统从外部获得数据,并不断地写入数据到存储系统中,这个过程,是典型的写密集场景,如果有重删系统的话,重删系统是生效的,以减少数据的重复存储。而本发明面向非易失性内存存储系统的增量更新机制(PostMerge),在数据被请求写入时,即时地直接存储新数据到临时空间中,并且这个写入过程是支持并发写入,这些

推后操作会延后新旧数据的合并，以确保新数据能更快地被持久化到非易失性内存中。在这个阶段，推后操作将会占主导，同时也有少量的数据后台合并操作被执行。通过在存储系统嵌入PostMerge，本发明能够加快数据采集阶段的处理时间。

[0107] 在知识融合和知识推理阶段，系统需要处理刚刚存储的数据，将收集的数据经过数据清洗、知识提取、知识融合、知识推理等步骤重新组织到已存储的知识图谱中。该过程的存储操作主要为读取刚刚采集的新数据，以及写入新数据到已经存储的知识图谱中。在即时响应阶段，本发明PostMerge会被触发拼接操作与推后操作。而在数据采集阶段与知识推理阶段之间，本发明会后台地完成数据采集阶段写入的新数据与系统原有的旧数据的合并。另外，知识推理计算的处理时间也会缩小更新机制之间的存储处理时间差异，因此，本发明对知识推理阶段的处理时间影响是十分有限的。

[0108] 在模型训练和模型验证阶段，系统需要加载现有的模型和知识图谱中的数据到内存中。同样的，在这个过程之前，本发明PostMerge的合并操作会后台地完成，停留在临时空间的增量段或新数据很少。因此，在模型训练和模型验证阶段的拼接操作极少，本发明PostMerge对读操作的影响可忽略；经过长时间的训练和模型验证后，模型需要从运行内存中导出到持久化的非易失性内存环境中，因此，涉及到了两次（训练和验证）大量的数据写入。而本发明PostMerge通过增量地写入新数据，减少其中的数据写入放大，可实现更快的数据导出，从而加快了模型训练和模型验证阶段的存储处理时间。

[0109] 在最后的模型转发阶段，系统需要将经过了验证的模型转发到其他节点中，在这个阶段，涉及到的存储操作是读取节点的模型，然后分发写入到多个节点中。在这个阶段，大部分部署了本发明——面向非易失性内存存储系统的增量更新机制的节点可以以较小的数据写入放大（仍有一定的元数据写入放大）快速接收模型，整体上实现更快的模型转发。

[0110] 综上所述，本发明具有以下优点：

[0111] 1、本发明PostMerge采用了增量的方式更新数据，相比于写时拷贝和日志机制有更少写放大；

[0112] 2、本发明PostMerge采用了分散结构的更新数据，相比于集中式的LFS和LSNVMM，PostMerge对并发执行更友好，更适用于支持多并发和细粒度更新的非易失性内存。

[0113] 在一些可选择的实施例中，在方框图中提到的功能/操作可以不按照操作示图提到的顺序发生。例如，取决于所涉及的功能/操作，连续示出的两个方框实际上可以被大体上同时地执行或所述方框有时能以相反顺序被执行。此外，在本发明的流程图中所呈现和描述的实施例以示例的方式被提供，目的在于提供对技术更全面的理解。所公开的方法不限于本文所呈现的操作和逻辑流程。可选择的实施例是可预期的，其中各种操作的顺序被改变以及其中被描述为较大操作的一部分的子操作被独立地执行。

[0114] 此外，虽然在功能性模块的背景下描述了本发明，但应当理解的是，除非另有相反说明，所述的功能和/或特征中的一个或多个可以被集成在单个物理装置和/或软件模块中，或者一个或多个功能和/或特征可以在单独的物理装置或软件模块中被实现。还可以理解的是，有关每个模块的实际实现的详细讨论对于理解本发明是不必要的。更确切地说，考虑到在本文中公开的装置中各种功能模块的属性、功能和内部关系的情况下，在工程师的常规技术内将会了解该模块的实际实现。因此，本领域技术人员运用普通技术就能够在无

需过度试验的情况下实现在权利要求书中所阐明的本发明。还可以理解的是，所公开的特定概念仅仅是说明性的，并不意在限制本发明的范围，本发明的范围由所附权利要求书及其等同方案的全部范围来决定。

[0115] 所述功能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，本发明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(可以是个人计算机，服务器，或者网络设备等)执行本发明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、移动硬盘、只读存储器(ROM,Read-Only Memory)、随机存取存储器(RAM,Random Access Memory)、磁碟或者光盘等各种可以存储程序代码的介质。

[0116] 在流程图中表示或在此以其他方式描述的逻辑和/或步骤，例如，可以被认为是用于实现逻辑功能的可执行指令的定序列表，可以具体实现在任何计算机可读介质中，以供指令执行系统、装置或设备(如基于计算机的系统、包括处理器的系统或其他可以从指令执行系统、装置或设备取指令并执行指令的系统)使用，或结合这些指令执行系统、装置或设备而使用。就本说明书而言，“计算机可读介质”可以是任何可以包含、存储、通信、传播或传输程序以供指令执行系统、装置或设备或结合这些指令执行系统、装置或设备而使用的装置。

[0117] 计算机可读介质的更具体的示例(非穷尽性列表)包括以下:具有一个或多个布线的电连接部(电子装置)、便携式计算机盘盒(磁装置)、随机存取存储器(RAM)、只读存储器(ROM)、可擦除可编辑只读存储器(EPROM或闪速存储器)、光纤装置以及便携式光盘只读存储器(CDROM)。另外，计算机可读介质甚至可以是可在其上打印所述程序的纸或其他合适的介质，因为可以例如通过对纸或其他介质进行光学扫描，接着进行编辑、解译或必要时以其他合适方式进行处理来以电子方式获得所述程序，然后将其存储在计算机存储器中。

[0118] 应当理解，本发明的各部分可以用硬件、软件、固件或它们的组合来实现。在上述实施方式中，多个步骤或方法可以用存储在存储器中且由合适的指令执行系统执行的软件或固件来实现。例如，如果用硬件来实现，和在另一实施方式中一样，可用本领域公知的下列技术中的任一项或他们的组合来实现:具有用于对数据信号实现逻辑功能的逻辑门电路的离散逻辑电路，具有合适的组合逻辑门电路的专用集成电路，可编程门阵列(PGA)，现场可编程门阵列(FPGA)等。

[0119] 在本说明书的描述中，参考术语“一个实施例”、“一些实施例”、“示例”、“具体示例”、或“一些示例”等的描述意指结合该实施例或示例描述的具体特征、结构、材料或者特点包含于本发明的至少一个实施例或示例中。在本说明书中，对上述术语的示意性表述不一定指的是相同的实施例或示例。而且，描述的具体特征、结构、材料或者特点可以在任何的一个或多个实施例或示例中以合适的方式结合。

[0120] 尽管已经示出和描述了本发明的实施例，本领域的普通技术人员可以理解:在不脱离本发明的原理和宗旨的情况下可以对这些实施例进行多种变化、修改、替换和变型，本发明的范围由权利要求及其等同物限定。

[0121] 以上是对本发明的较佳实施进行了具体说明，但本发明并不限于所述实施例，熟悉本领域的技术人员在不违背本发明精神的前提下还可做出种种的等同变形或替换，这些

等同的变形或替换均包含在本申请权利要求所限定的范围内。

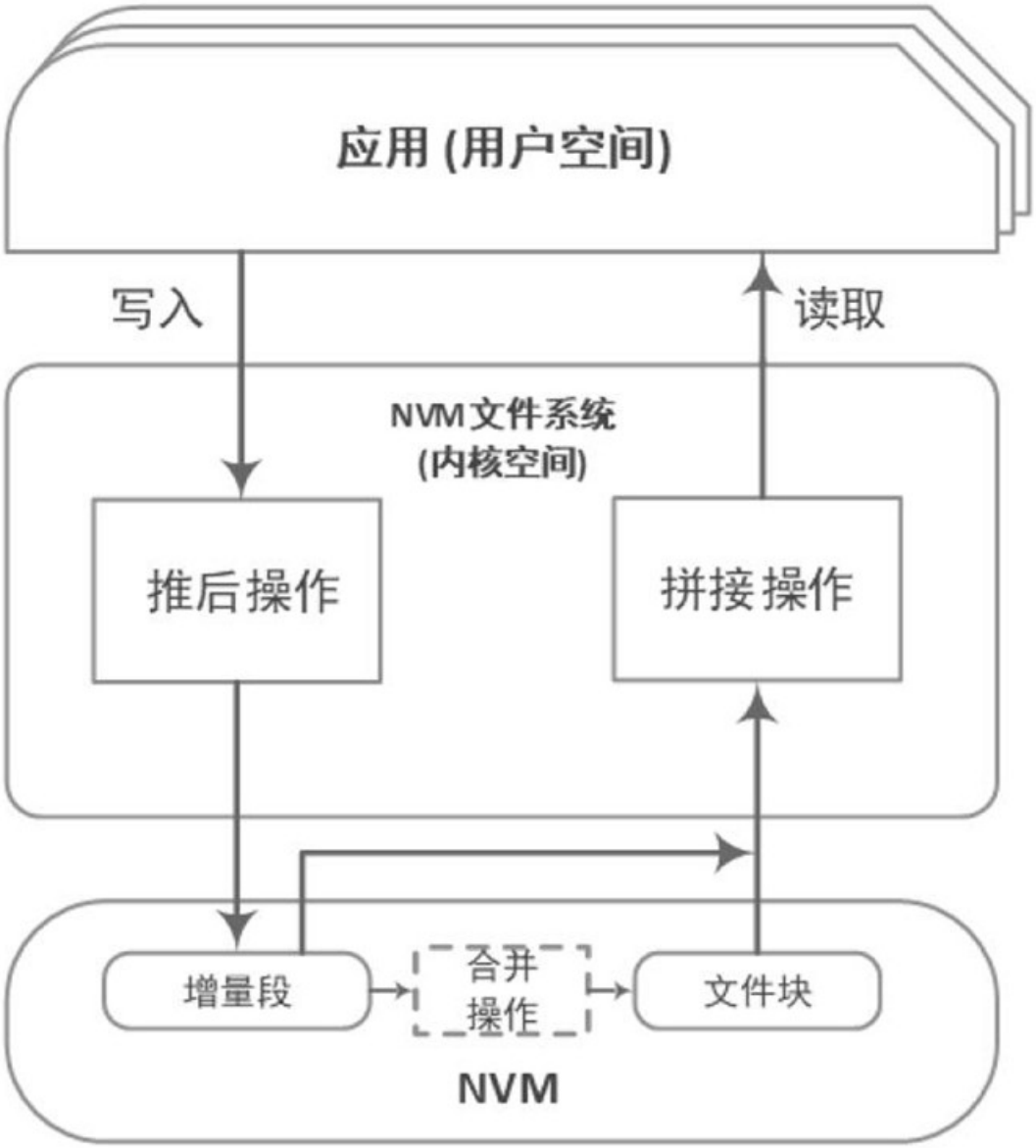


图 1

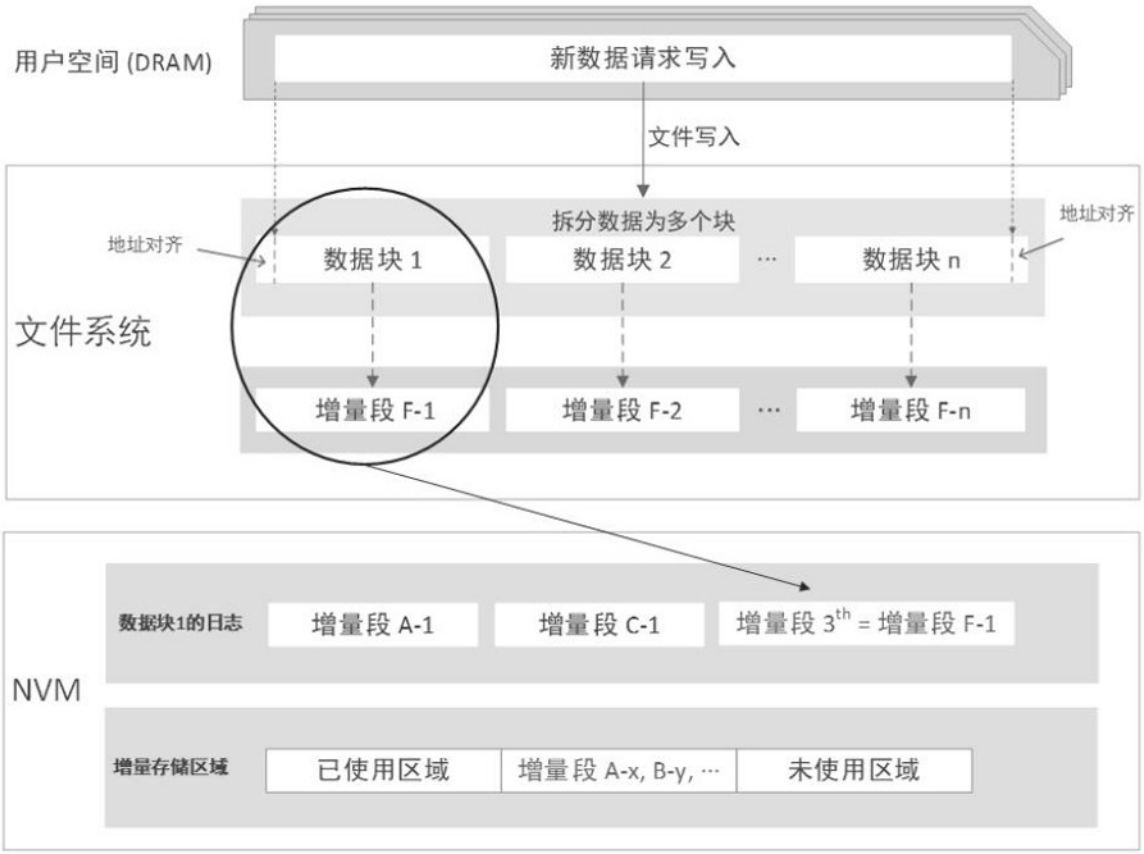

用户空间 (DRAM)
新数据请求写入
文件写入
拆分数据为多个块
地址对齐
地址对齐
数据块 1
数据块 2
…
数据块 n
文件系统
增量段 F-1
增量段 F-2
…
增量段 F-n
NVM
数据块1的日志
增量段 A-1
增量段 C-1
增量段 3th = 增量段 F-1
增量存储区域
已使用区域
增量段 A-x, B-y, …
未使用区域

图 2

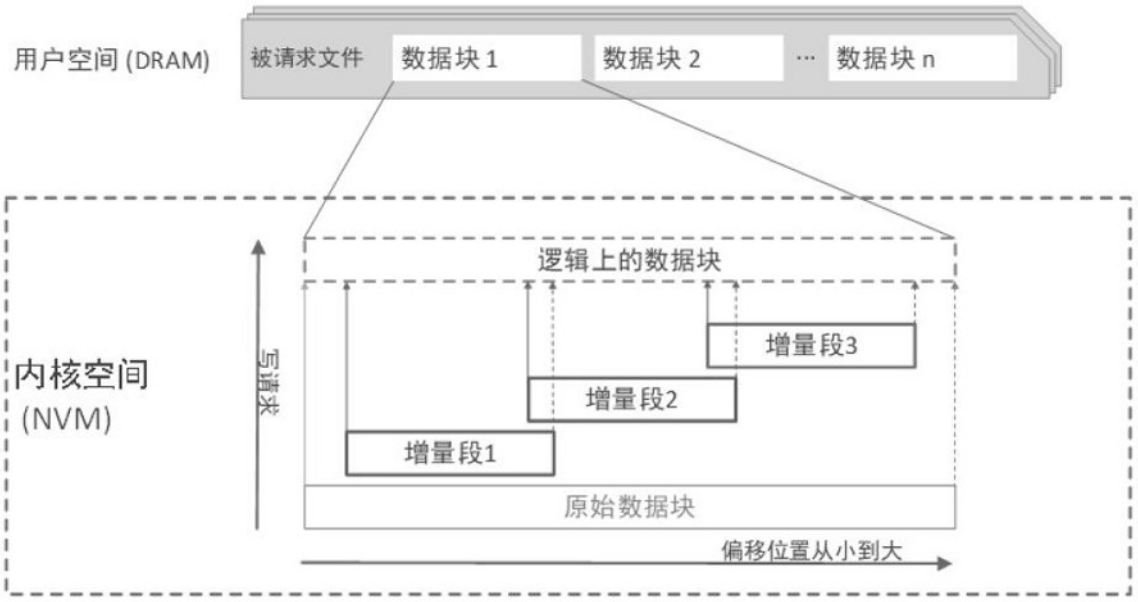

用户空间 (DRAM)
被请求文件
数据块 1
数据块 2
…
数据块 n
逻辑上的数据块
内核空间
(NVM)
写请求
增量段3
增量段2
增量段1
原始数据块
偏移位置从小到大

图 3

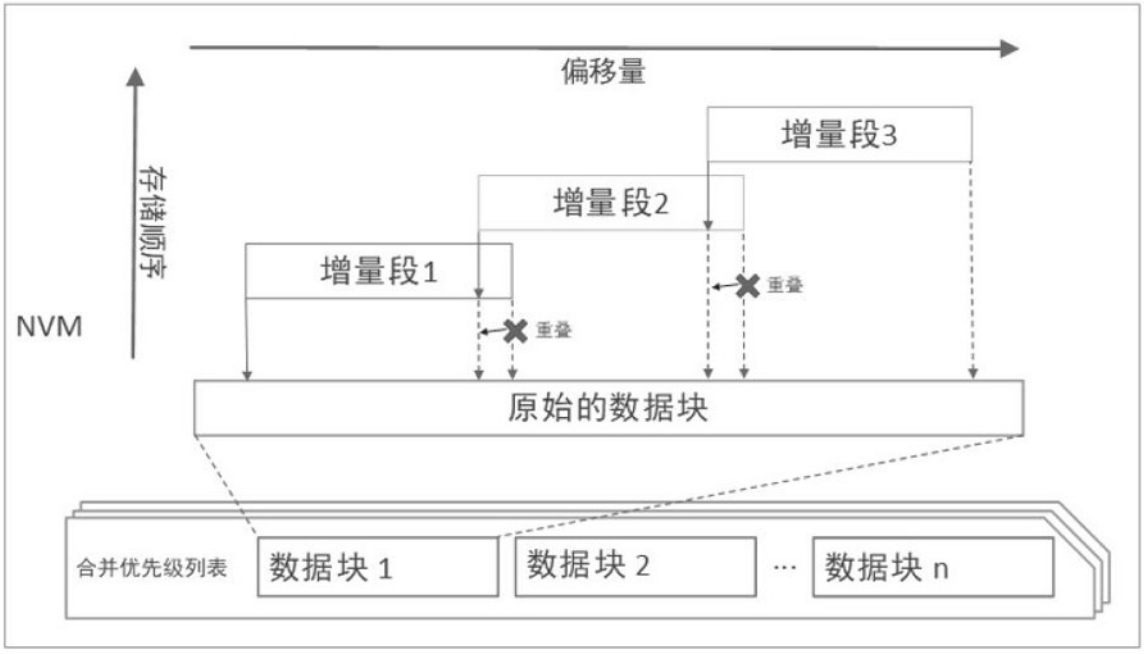

偏移量
存储顺序
NVM
增量段3
增量段2
增量段1
重叠
重叠
原始的数据块
合并优先级列表
数据块 1
数据块 2
…
数据块 n

图 4

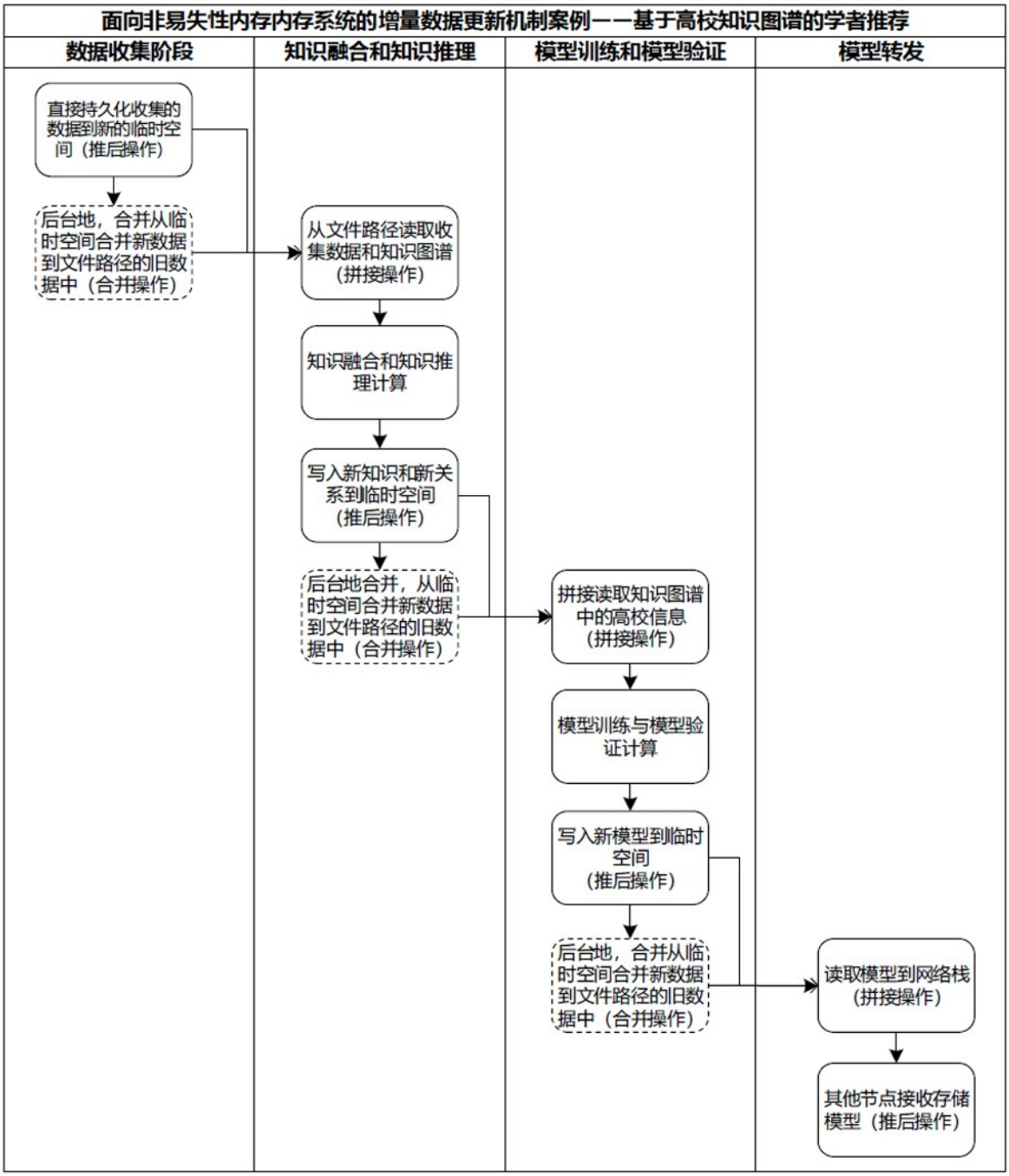

面向非易失性内存内存系统的增量数据更新机制案例——基于高校知识图谱的学者推荐
数据收集阶段
知识融合和知识推理
模型训练和模型验证
模型转发
直接持久化收集的数据到新的临时空间（推后操作）
后台地，合并从临时空间合并新数据到文件路径的旧数据中（合并操作）
从文件路径读取收集数据和知识图谱（拼接操作）
知识融合和知识推理计算
写入新知识和新关系到临时空间（推后操作）
后台地合并，从临时空间合并新数据到文件路径的旧数据中（合并操作）
拼接读取知识图谱中的高校信息（拼接操作）
模型训练与模型验证计算
写入新模型到临时空间（推后操作）
后台地，合并从临时空间合并新数据到文件路径的旧数据中（合并操作）
读取模型到网络栈（拼接操作）
其他节点接收存储模型（推后操作）

图 5