(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 113190332 B
(45)授权公告日 2024.09.13

(21)申请号 202010037217.5

(22)申请日 2020.01.14

(65)同一申请的已公布的文献号
申请公布号 CN 113190332 A

(43)申请公布日 2021.07.30

(73)专利权人 伊姆西IP控股有限责任公司
地址 美国马萨诸塞州

(72)发明人 张明 吕烁

(74)专利代理机构 北京市金杜律师事务所 11256
专利代理师 张翠玲

(51)Int.Cl.
G06F 9/46(2006.01)
G06F 16/22(2019.01)
G06F 16/23(2019.01)

(56)对比文件
US 2018173720 A1,2018.06.21
US 8756194 B1,2014.06.17

审查员 么旭君

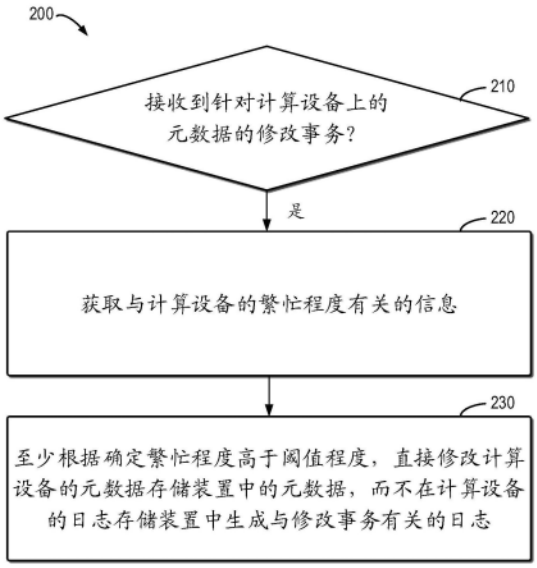
权利要求书4页 说明书9页 附图3页

(54)发明名称

用于处理元数据的方法、设备和计算机程序产品

(57)摘要

本公开的实施例提供了用于处理元数据的方法、设备和计算机程序产品。一种处理元数据的方法包括响应于接收到针对计算设备上的元数据的修改事务,获取与计算设备的繁忙程度有关的信息;以及至少根据确定繁忙程度高于阈值程度,直接修改计算设备的元数据存储装置中的元数据,而不在计算设备的日志存储装置中生成与修改事务有关的日志。通过这个方案,使得计算设备在繁忙状态或是非繁忙状态下,均可以实现较好的处理性能。尤其当计算设备处于繁忙状态时,新触发的修改事件依然可以被及时地执行,计算设备在繁忙状态下的IOPS性能被提高。



CN 113190332 B

1.一种处理元数据的方法,包括:

响应于接收到针对计算设备上的元数据的第一修改事务,获取与所述计算设备的第一繁忙程度有关的信息;以及

执行确定所述第一繁忙程度是否大于阈值程度的第一确定的操作;以及

至少根据所述第一繁忙程度大于所述阈值程度的所述第一确定,直接修改所述计算设备的元数据存储装置中的第一元数据,而不在所述计算设备的日志存储装置中生成与所述第一修改事务有关的第一日志;

其中修改所述第一元数据包括:

获取所述第一元数据的第一修改记录,所述第一修改记录指示是否存在尚未完成的、针对所述第一元数据的其他修改事务;以及

根据第一修改记录指示不存在尚未完成的其他修改事务的第一确定,直接修改所述第一元数据,而不生成所述第一日志。

2.根据权利要求1所述的方法,进一步包括:针对第二元数据的第二修改事务:

根据第二繁忙程度小于所述阈值程度的第二确定,

在所述日志存储装置中生成与所述第二修改事务有关的第二日志;以及

修改所述元数据存储装置中的所述第二元数据。

3.根据权利要求1所述的方法,进一步包括:针对第二元数据的第二修改事务:

获取所述第二元数据的第二修改记录,所述第二修改记录指示是否存在尚未完成的、针对所述第二元数据的其他修改事务;

根据所述第二修改记录指示存在尚未完成的其他修改事务的第二确定,

在所述日志存储装置中生成与所述第二修改事务有关的第二日志;以及

修改所述元数据存储装置中的所述第二元数据。

4.根据权利要求3所述的方法,进一步包括:

确定所述第一修改事务的执行状态;以及

基于所述执行状态,更新所述第一元数据的所述第一修改记录。

5.根据权利要求1所述的方法,其中获取与所述计算设备的第一繁忙程度有关的信息包括:

获取指示所述日志存储装置的存储资源的占用情况的信息。

6.根据权利要求1所述的方法,进一步包括:

响应于接收到针对所述计算设备上的元数据的第二修改事务,获取与所述计算设备的第二繁忙度相关的信息;

执行确定所述第二繁忙程度是否大于所述阈值程度的第二确定的操作;以及

至少根据所述第二繁忙度不大于所述阈值程度的所述第二确定,在所述计算设备的所述日志存储装置中生成与所述第二修改事务相关的日志,并且在所述日志存储装置中生成所述日志之后,修改所述计算设备的所述元数据存储装置中的第二元数据。

7.根据权利要求6所述的方法,其中所述元数据存储装置中的所述第一元数据被主机修改;

其中所述元数据存储装置中的所述第二元数据被与所述主机不同的进程修改;以及

其中所述方法进一步包括:

在所述元数据存储装置中的所述第一元数据被修改之后,由所述主机修改另一元数据存储装置中的所述第一元数据,并且

在所述元数据存储装置中的所述第二元数据被修改之后,由后台程序修改所述另一元数据存储装置中的所述第二元数据。

8.一种电子设备,包括:

处理器;以及

与所述处理器耦合的存储器,所述存储器具有存储于其中的指令,所述指令在被处理器执行时使所述设备执行动作,所述动作包括:

响应于接收到针对计算设备上的元数据的第一修改事务,获取与所述计算设备的第一繁忙程度有关的信息;以及

执行确定所述第一繁忙程度是否大于阈值程度的第一确定的操作;以及

至少根据所述第一繁忙程度大于所述阈值程度的所述第一确定,直接修改所述计算设备的元数据存储装置中的第一元数据,而不在所述计算设备的日志存储装置中生成与所述第一修改事务有关的第一日志;

其中修改所述第一元数据包括:

获取所述第一元数据的第一修改记录,所述第一修改记录指示是否存在尚未完成的、针对所述第一元数据的其他修改事务;以及

根据第一修改记录指示不存在尚未完成的其他修改事务的第一确定,直接修改所述第一元数据,而不生成所述第一日志。

9.根据权利要求8所述的设备,所述动作进一步包括:针对第二元数据的第二修改事务:

根据第二繁忙程度小于所述阈值程度的第二确定,

在所述日志存储装置中生成与所述第二修改事务有关的第二日志;以及

修改所述元数据存储装置中的所述第二元数据。

10.根据权利要求8所述的设备,所述动作进一步包括:针对第二元数据的第二修改事务:

获取所述第二元数据的第二修改记录,所述第二修改记录指示是否存在尚未完成的、针对所述第二元数据的其他修改事务;

根据所述第二修改记录指示存在尚未完成的其他修改事务的第二确定,

在所述日志存储装置中生成与所述第二修改事务有关的第二日志;以及

修改所述元数据存储装置中的所述第二元数据。

11.根据权利要求10所述的设备,所述动作进一步包括:

确定所述第一修改事务的执行状态;以及

基于所述执行状态,更新所述第一元数据的所述第一修改记录。

12.根据权利要求8所述的设备,其中获取与所述计算设备的第一繁忙程度有关的信息包括:

获取指示所述日志存储装置的存储资源的占用情况的信息。

13.根据权利要求8所述的设备,所述动作进一步包括:

响应于接收到针对所述计算设备上的元数据的第二修改事务,获取与所述计算设备的

第二繁忙度相关的信息;

执行确定所述第二繁忙程度是否大于所述阈值程度的第二确定的操作;以及

至少根据所述第二繁忙度不大于所述阈值程度的所述第二确定,在所述计算设备的所述日志存储装置中生成与所述第二修改事务相关的日志,并且在所述日志存储装置中生成所述日志之后,修改所述计算设备的所述元数据存储装置中的第二元数据。

14.根据权利要求13所述的设备,其中所述元数据存储装置中的所述第一元数据被主机修改;

其中所述元数据存储装置中的所述第二元数据被与所述主机不同的进程修改;以及

其中所述动作进一步包括:

在所述元数据存储装置中的所述第一元数据被修改之后,由所述主机修改另一元数据存储装置中的所述第一元数据,并且

在所述元数据存储装置中的所述第二元数据被修改之后,由后台程序修改所述另一元数据存储装置中的所述第二元数据。

15.一种计算机程序产品,所述计算机程序产品被有形地存储在非瞬态计算机可读介质上并且包括计算机可执行指令,所述计算机可执行指令在被执行时使设备:

响应于接收到针对计算设备上的元数据的第一修改事务,获取与所述计算设备的第一繁忙程度有关的信息;以及

执行确定所述第一繁忙程度是否大于阈值程度的第一确定的操作;以及

至少根据所述第一繁忙程度大于所述阈值程度的所述第一确定,直接修改所述计算设备的元数据存储装置中的第一元数据,而不在所述计算设备的日志存储装置中生成与所述第一修改事务有关的第一日志;

其中修改所述第一元数据包括:

获取所述第一元数据的第一修改记录,所述第一修改记录指示是否存在尚未完成的、针对所述第一元数据的其他修改事务;以及

根据第一修改记录指示不存在尚未完成的其他修改事务的第一确定,直接修改所述第一元数据,而不生成所述第一日志。

16.根据权利要求15所述的计算机程序产品,所述计算机可执行指令在被执行时还使所述设备:针对第二元数据的第二修改事务:

根据第二繁忙程度小于所述阈值程度的第二确定,

在所述日志存储装置中生成与所述第二修改事务有关的第二日志;以及

修改所述元数据存储装置中的所述第二元数据。

17.根据权利要求15所述的计算机程序产品,所述计算机可执行指令在被执行时还使所述设备:针对第二元数据的第二修改事务:

获取所述第二元数据的第二修改记录,所述第二修改记录指示是否存在尚未完成的、针对所述第二元数据的其他修改事务;

根据所述第二修改记录指示存在尚未完成的其他修改事务的第二确定,

在所述日志存储装置中生成与所述第二修改事务有关的第二日志;以及

修改所述元数据存储装置中的所述第二元数据。

18.根据权利要求17所述的计算机程序产品,所述计算机可执行指令在被执行时还使

所述设备：

确定所述第一修改事务的执行状态;以及

基于所述执行状态,更新所述第一元数据的所述第一修改记录。

19.根据权利要求15所述的计算机程序产品,其中在被执行时所述设备获取与所述计算设备的第一繁忙程度有关的信息的所述计算机可执行指令还使所述设备：

获取指示所述日志存储装置的存储资源的占用情况的信息。

20.根据权利要求15所述的计算机程序产品,所述计算机可执行指令在被执行时还使所述设备：

响应于接收到针对所述计算设备上的元数据的第二修改事务,获取与所述计算设备的第二繁忙度相关的信息;

执行确定所述第二繁忙程度是否大于所述阈值程度的第二确定的操作;以及

至少根据所述第二繁忙度不大于所述阈值程度的所述第二确定,在所述计算设备的所述日志存储装置中生成与所述第二修改事务相关的日志,并且在所述日志存储装置中生成所述日志之后,修改所述计算设备的所述元数据存储装置中的第二元数据。

用于处理元数据的方法、设备和计算机程序产品

技术领域

[0001] 本公开的实施例涉及存储技术，并且更具体地，涉及用于处理元数据的方法、电子设备和计算机程序产品。

背景技术

[0002] 传统的数据存储系统中，数据分为实际数据和元数据。实际数据对应用于户存储的真实数据(诸如，文档、音频、视频、图片，等等)。元数据指代用来描述实际数据的特征的数据，诸如访问权限、实际数据拥有者以及实际数据块的分布信息等等。在计算设备的运行过程中，涉及大量针对元数据的操作，诸如，元数据的创建、删除、更新/修改等。计算设备根据所执行的任务的不同，所处的繁忙程度也不同。尤其当计算设备处于繁忙状态时，传统的元数据处理方式无法实现较好的处理性能。因此，需要一种高效且可靠的机制，用于实现元数据的维护。

发明内容

[0003] 本公开的实施例提供了一种用于处理元数据的方案。

[0004] 在本公开的第一方面，提供了一种处理元数据的方法。该方法包括：响应于接收到针对计算设备上的元数据的修改事务，获取与计算设备的繁忙程度有关的信息；以及至少根据确定繁忙程度高于阈值程度，直接修改计算设备的元数据存储装置中的元数据，而不在计算设备的日志存储装置中生成与修改事务有关的日志。

[0005] 在本公开的第二方面，提供了一种电子设备。该电子设备包括处理器以及与处理器耦合的存储器，存储器具有存储于其中的指令，指令在被处理器执行时使设备执行动作，动作包括：响应于接收到针对计算设备上的元数据的修改事务，获取与计算设备的繁忙程度有关的信息；以及至少根据确定繁忙程度高于阈值程度，直接修改计算设备的元数据存储装置中的元数据，而不在计算设备的日志存储装置中生成与修改事务有关的日志。

[0006] 在本公开的第三方面，提供了一种计算机程序产品，计算机程序产品被有形地存储在计算机可读介质上并且包括计算机可执行指令，计算机可执行指令在被执行时使设备执行根据上述第一方面的方法。

[0007] 提供发明内容部分是为了简化的形式来介绍对概念的选择，它们在下文的具体实施方式中将被进一步描述。发明内容部分无意标识本公开的关键特征或主要特征，也无意限制本公开的范围。

附图说明

[0008] 通过结合附图对本公开示例性实施例进行更详细的描述，本公开的上述以及其它目的、特征和优势将变得更加明显，其中，在本公开示例性实施例中，相同的参考标号通常代表相同部件。

[0009] 图1示出了能够在其中实现本公开的一些实施例的存储系统的示意框图；

[0010] 图2示出了根据本公开的一些实施例的处理元数据的过程的流程图；

[0011] 图3示出了能够在其中实现本公开的一些实施例的存储系统的示意框图；以及

[0012] 图4示出了可以用来实施本公开的实施例的示例设备的框图。

具体实施方式

[0013] 下面将参考附图中示出的若干示例实施例来描述本公开的原理。虽然附图中显示了本公开的优选实施例，但应当理解，描述这些实施例仅是为了使本领域技术人员能够更好地理解进而实现本公开，而并非以任何方式限制本公开的范围。

[0014] 在本文中使用的术语“包括”及其变形表示开放性包括，即“包括但不限于”。除非特别申明，术语“或”表示“和/或”。术语“基于”表示“至少部分地基于”。术语“一个示例实施例”和“一个实施例”表示“至少一个示例实施例”。术语“另一实施例”表示“至少一个另外的实施例”。术语“第一”、“第二”等等可以指代不同的或相同的对象。下文还可能包括其他明确的和隐含的定义。

[0015] 在本文中使用的术语“事务”指代存储系统(诸如，文件系统等)中的执行任务的原子单位。事务在执行的任务时作为一个整体处于执行成功或执行失败的状态。“修改事务”通常与系统中数据(即，文件)的操作(例如，对文件系统中文件的创建、修改、删除、复制、剪切、粘贴等操作)相关联。

[0016] 在本文中使用的术语“元数据存储装置”指代存储系统(诸如，文件系统等)中的用于存储元数据的存储装置，包括但不限于，内存/高速缓存、磁盘以及硬盘等。

[0017] 在本文中使用的术语“日志存储装置”指代存储系统(诸如，文件系统等)中用于存储与针对元数据的修改事务有关的日志的存储装置。

[0018] 在传统的技术中，元数据除了在磁盘中保存外，为了保证系统的高效运行，计算设备在内存/高速缓存中也会保存部分元数据(通常以元数据的镜像的形式)。因此，对元数据的修改，通常涉及内存/高速缓存中的以及磁盘中的相应存储区域的修改。此外，针对元数据的操作是时间敏感性操作。针对元数据的修改事务需要按照元数据修改事件发生的时间顺序依次被执行，以保证元数据修改的正确性。

[0019] 在传统的方案中，通常借助与修改事务有关的日志来实现元数据的处理(诸如，元数据的修改等)。但是传统的方案过度依赖于日志的创建，使得在计算设备处于繁忙状态时(尤其是方日志存储装置的存储空间不足时)，新触发的修改事务无法被及时地处理，导致计算设备的执行性能不高。

[0020] 根据本公开的实现，提出了一种改进的用于处理元数据的方案。根据本公开，当计算设备处于繁忙的状态时，计算设备直接修改计算设备的元数据存储装置中的元数据，提高了计算设备在繁忙状态下的IOPS。而当计算设备处于非繁忙状态时，则借助与修改事务有关的日志来处理元数据，从而实现快速的输入输出响应。由此，使得计算设备无论在繁忙或是非繁忙状态下(尤其当计算设备处于繁忙状态时)，计算设备均可以及时地执行新触发的针对元数据的修改事务，计算设备的IOPS性能得到改善。

[0021] 图1示出了本公开的实施例可以在其中被实现的存储系统100的示意图。示例数据存储系统100包括计算设备110。计算设备110作为数据存储系统100的核心处理设备，用于执行存储系统100中的各种逻辑操作。

[0022] 计算设备110包括日志存储装置120。日志存储装置120用于存储针对元数据的修改事务的日志140-1、140-2、……、140-N(以下统称为日志140),其中N为生成的日志的数目。根据本公开的一些示例实施例,日志存储装置120是断电不丢失类型的存储装置。

[0023] 计算设备110还包括元数据存储装置130,用于存储计算设备110的元数据150-1、150-1、……、150-M(以下统称为元数据150),其中M为存储的元数据的数目。根据本公开的一些示例实施例,元数据存储装置130可以至少包括两个彼此独立的存储装置,例如,内存/高速换缓存和磁盘。内存/高速换缓存仅在计算设备110运行时用于存储元数据150的镜像,以实现对于元数据150的快速访问和处理。磁盘用于存储全部的元数据150。磁盘的一些示例包括但不限于固态硬盘(SSD)、硬盘驱动器(HDD)、串行高级技术附件(SATA)盘、串行连接(SA)小型计算机系统接口(SCSI)盘SAS盘等等。

[0024] 应当理解,虽然图1示出了特定数目的计算设备110、日志存储装置120以及元数据存储装置130,日志存储装置120存储有特定数目的日志140,元数据存储装置130存储有特定数目的元数据150,在其他的一些示例实施例中,计算设备、日志存储装置、元数据存储装置、日志以及元数据的数量可能发生变化,本公开的范围在此方面不受限制。

[0025] 进一步地,还应当理解,计算设备110与日志存储装置120和元数据存储装置130之间的布置也不限于图1所示的具体示例。在其他一些示例实施例中,日志存储装置120可以被布置在计算设备110的外部,或者甚至被布置在元数据存储装置130中。备选地或附加地,日志存储装置120还可以是远程存储设备、或者云存储设备等,本公开的范围在此方面亦不受限制。元数据存储装置130可以包括多种类型的多个存储装置,多个存储装置可以部分地或全部地被布置在计算设备110的内部,计算设备110的外部,甚至是远程存储设备、或者云存储设备等,本公开的范围在此方面亦不受限制。

[0026] 此外,还应当理解,计算设备110和日志存储设备120以及元数据存储装置130的连接关系也可以是各种形式,例如,有线、无线或者因特网等等,本公开的范围在此方面亦不受限制。

[0027] 如上所讨论的,元数据存储装置130可以包括内存/高速缓存以及磁盘两种类型的存储装置。

[0028] 发明人意识到,处理元数据150(诸如,修改元数据150)的一种可能方案为,首先在计算设备110的日志存储装置120中生成与修改事务有关的日志140,然后再由计算设备110修改计算设备110的元数据存储装置130(即,内存/高速缓存和磁盘)中的元数据150,即更新元数据存储装置130中与该元数据150对应的存储区域。例如,当计算设备110的主机(诸如,CPU)接收到针对元数据150的修改事务后,首先在计算设备110的日志存储装置120中生成与该修改事务有关的日志140。该日志140简要地记录了与该修改事务对应的元数据修改操作的信息。此外,可以在该日志存储装置120与计算设备110的磁盘之间建立映射关系,使得一旦元数据的修改事务被记录在日志存储设备120中,则磁盘中的相应的元数据随后一定会被更新。随后,计算设备110首先更新内存/高速缓存中的元数据150,并将更新后的元数据150记录至脏缓存区。脏缓存区的数据通过后台程序被最终写入磁盘中,以实现对元数据150的修改。

[0029] 在上述操作中,向主机返回响应的操作可以在生成与修改事务有关的日志140后或在更新内存/高速缓冲中的元数据150后立即被执行。该方案的优点在于,由于向主机返

回响应时并未真正地执行磁盘中的元数据150的修改,因此向主机返回响应所需的时间被大大地缩短了,从而提高了计算设备110的每秒进行读写操作的次数(Input/OutputOperations Per Second,IOPS)。

[0030] 发明人进一步意识到,计算设备110的日志储装置120的容量是有限的,当计算设备110处于繁忙状态时,针对元数据150的修改事务的数量将会急剧增长,导致计算设备110的日志存储装置120的存储空间将严重不足。此时,若新的针对元数据150的修改事务被触发,则由于日志存储装置120不存在可用的存储空间,将导致新触发的修改事务不能被立即地执行。计算设备110需要等待之前的修改事务被执行完成并且计算设备110将在日志存储装置120中存储的相应的日志140移除之后,才可以执行新触发的修改事务。因此,该方案在计算设备处于繁忙状态时,执行性能不佳。

[0031] 发明人还意识到,处理元数据150(诸如,修改元数据150)的另一种可能方案为直接修改计算设备110的元数据存储装置130(即,内存/高速缓存以及磁盘)中的元数据150,而不在计算设备110的日志存储装置120中生成与修改事务有关的日志140。例如,当计算设备110的主机接收到元数据修改事务后,首先更新内存/高速缓存中的元数据150,随后更新磁盘中的元数据150。

[0032] 在上述操作中,向主机返回响应的操作需要在磁盘中的元数据更新完成之后才被执行。因此,向主机返回响应所需的时间相对较长。同时发明人也进一步意识到,该方案的优势在于,计算设备110无需在其日志存储装置120中生成与修改事务有关的日志140,因此即使在计算设备110处于繁忙的状态时,或者日志存储装置120已不存在可用存储空间的状态下,新触发的元数据修改事务依然可以被立即执行。

[0033] 发明人在意识到上述两种可能的方案各自的优势和缺点的基础上,进一步提出了本公开的方案,下面将结合附图来详细描述本公开的实施例。

[0034] 图2示出了根据本公开的一些实施例的处理元数据150的过程200的流程图。过程200可以由图1中的计算设备110实现。为便于说明,参考图1来描述过程200。

[0035] 在框210,计算设备110接收针对计算设备110上的元数据150的修改事务。修改事务可以由多种操作而被触发,例如,包括但不限于对存储在计算设备110上的文件/文件夹执行的创建、修改、删除、复制、粘贴、剪切等操作。

[0036] 在框220,当计算设备110确定接收到针对计算设备110上的元数据150的修改事务时,获取与计算设备110的繁忙程度有关的信息。

[0037] 根据本公开的一些实施例,计算设备110获取与计算设备110的繁忙程度有关的信息可以具体为获取指示日志存储装置120的存储资源的占用情况的信息。应当理解,在该特定示例实施例中,指示与计算设备110的繁忙程度有关的信息可以具有多种表示形式,例如日志存储装置120的存储资源的使用百分比、日志存储装置120的剩余可用存储资源的数目,或者其他任何可以指示日志存储装置120的存储资源使用情况的信息。

[0038] 由于日志存储装置120的存储资源的占用情况的信息与是否可以立即生成修改事务的日志140之间存在直接关联关系,因此根据该实施例,计算设备110可以更合理地确定以何种方式执行针对元数据150的修改事务。

[0039] 备选地或附加地,根据本公开的一些实施例,计算设备110获取与计算设备110的繁忙程度有关的信息可以具体为获取指示计算设备110的中央处理单元(CPU)或主机的运

算资源的使用情况的信息。应当理解,在该特定示例实施例中,指示与计算设备110的繁忙程度有关的信息可以有多种表示形式,例如CPU或主机的运算资源的使用百分比、CPU或主机的剩余可用的运算资源的数目,或者其他任何可以指示CPU或主机的运算资源使用情况的信息。

[0040] 备选地或附加地,根据本公开的一些其他实施例,指示与计算设备110的繁忙程度有关的信息还可以是计算设备110的IOPS值等,本公开对于指示与繁忙程度有关的信息的具体参数不加以限定。还应当理解,在其他示例实施例中,与繁忙程度有关的信息还可以是多个参数的组合。

[0041] 在框230,当确定繁忙程度高于阈值程度时,计算设备110直接修改计算设备110的元数据存储装置130中的元数据150,而不在计算设备110的日志存储装置120中生成与修改事务有关的日志140。

[0042] 应当理解,根据与计算设备110的繁忙程度有关的信息的表现形式的不同,阈值程度也可以具有多种不同的表现形式。还应当理解,阈值程度可以是存储系统100的管理人员预先配置的,也可以是计算设备110根据历史数据动态生成的,本公开对于阈值程度的配置方式亦不加以限定。

[0043] 根据本公开的一些示例实施例,计算设备110直接修改计算设备110的元数据存储装置130中的元数据150可以具体为,计算设备110首先修改内存/高速缓存中与该元数据150对应的存储空间(即,元数据150的镜像),并随后由相应的进程修改磁盘中的与该元数据150对应的存储空间。

[0044] 附加地,当磁盘中的与该元数据150对应的存储空间被更新完成后,执行任务的进程向计算设备110(通常是计算设备110的主机)返回响应,以指示该修改事务被执行完成。

[0045] 以此方式,即使计算设备110处于繁忙状态,其依然可以及时地执行针对元数据150的修改事务,从而提高了计算设备110在繁忙时的IOPS。

[0046] 根据本公开的一些示例实施例,当确定繁忙程度不高于阈值程度时,计算设备110首先在计算设备110的日志存储装置120中生成与修改事务有关的日志140,然后修改计算设备110的元数据存储装置130中的相应的元数据150。

[0047] 下面将给出由计算设备110执行上述操作的一种示例实现,应当理解的是,该实现仅作为示例而被提出,不应理解为对本公开的限制。

[0048] 根据本公开的一些示例实施例,计算设备110首先在计算设备110的日志存储装置120中生成与修改事务有关的日志140;随后,计算设备110修改内存/高速缓存中与该元数据150对应的存储空间(即,元数据150的镜像)并将更新后的元数据150存储至脏缓存区。执行该修改事务的进程可以在生成与修改事务有关的日志140后或在更新内存/高速缓冲中的元数据150完成之后向计算设备110(通常是计算设备110的主机)主机返回响应,以向计算设备110指示该修改事务被执行完成。被存储在脏缓冲区中的元数据,可以通过后台程序被最终写入磁盘中,以完成磁盘中的与该元数据150对应的存储空间的修改。

[0049] 根据该示例实施例,当计算设备110处于非繁忙状态时,借助生成与修改事务有关的日志140的方式处理元数据,使得向主机返回响应所需的时间被大大缩短,从而提高了计算设备110在非繁忙状态下的IOPS。

[0050] 根据本公开的上述实施例,计算设备110在执行针对元数据150的修改事务时,获

取与计算设备110的繁忙程度有关的信息,根据获取的繁忙程度信息执行针对修改事务的不同操作,使得在任何情况下,计算设备110均可以及时地执行修改事务,提高了计算设备110的IOPS性能。尤其当系统处于繁忙状态时,本公开可以很好地改善计算设备110的IOPS性能。

[0051] 进一步地,发明人还注意到,在一些场景中,用户会在短时间内对存储系统100中的相同文件进行多次操作,使得在存储系统100中在短时间内存在针对同一元数据150的多个修改事务。如之前所讨论的,针对元数据150的修改事务是时间敏感的。因此,计算设备110需要按照元数据修改事件发生的时间顺序,依次执行针对元数据150的修改事务,以保证元数据150修改的正确性。

[0052] 为了保证元数据修改的准确性,本公开进一步提出了一种保证修改事务按照时间顺序被依次执行的解决方案。具体为,计算设备110在执行针对元数据150的修改事务时,除获取与计算设备110的繁忙程度有关的信息外,还进一步获取元数据150的修改记录。该修改记录可以指示是否存在尚未完成的、针对该元数据150的其他修改事务。计算设备110仅在当计算设备110处于繁忙装置并且不存在未完成的其他修改事务时,才直接修改计算设备110的元数据存储装置130中的元数据150,而不在计算设备110的日志存储装置120中生成与修改事务有关的日志140。以此方式,可以更好地保证修改事务按照时间被依次执行。下面对该方案进行详细说明。

[0053] 根据本公开的一些示例实施例,当确定繁忙程度高于阈值程度时,计算设备110进一步获取元数据150的修改记录。该修改记录可以存储在计算设备110可以访问的任一存储空间中,例如存在计算设备110的内存/高速缓存中。

[0054] 应当理解,修改记录的表示方式可以是多样的。一种可能的表示方式为,针对每个元数据150,采用1比特的信息指示是否存在尚未完成的、针对该元数据150的其他修改事务。另一种可能的表示方式是,记录针对该元数据150的、尚未完成的其他修改事务的数目。又另一可能的表示方式是,针对每个修改事务,生成至少包括元数据150的标识信息的修改条目,以使得计算设备110可以根据记录的标识信息确定该元数据150的修改事务的执行状态信息。应当理解,在其他实施例中,可以采用其他的方式记录修改记录,本公开对此不加以限定。此外,还应当理解,修改记录可以以哈希表、链表等各种方式被记录,本公开对此亦不加以限定。

[0055] 根据公开的一些示例实施方式,执行状态可以包括:正在处理/尚未完成、已完成等。作为另一种备选的实施实施例,执行状态还可以是以百分比形式表示的执行进度。应当理解,执行状态可以具有多种表现形式,本公开的范围在此方面亦不受限制。

[0056] 根据本公开的一些示例实施例,当计算设备110确定修改记录指示不存在尚未完成的其他修改事务时,计算设备110直接修改计算设备110的元数据存储装置130中的元数据150。具体为,计算设备110首先修改内存/高速缓存中与该元数据150对应的存储空间(即,元数据150的镜像),并随后修改磁盘中的与该元数据150对应的存储空间。

[0057] 以此方式,使得当计算设备110直接修改计算设备110的元数据存储装置130中的元数据150时,修改事务依然可以按照时间顺便被依次执行,由此元数据150处理的正确性可以得到保证。

[0058] 附加地,根据本公开的一些示例实施例,当计算设备110确定修改记录指示存在尚

未完成的其他修改事务时,则计算设备110不能直接修改计算设备110的元数据存储装置120中的元数据。此时,若计算设备110处于繁忙装置(诸如,日志存储装置120不存在可用的存储空间),计算设备需要等待。当在先的修改事务执行完毕并且日志存储装置120中的与该修改事务对应的日志140被移除,日志存储装置120存在可用的剩余空间时,该修改事务可以被执行。当执行该修改事务时,计算设备首先在计算设备110的日志存储装置120中生成与修改事务有关的日志140;随后,计算设备110修改内存/高速缓存中与该元数据150对应的存储空间(即,元数据150的镜像)并由相应的进程将更新后的元数据150记录至脏缓存区。在生成与修改事务有关的日志140后或在更新内存/高速缓冲中的元数据150后,执行该修改事务的进程向计算设备110(通常是计算设备110的主机)主机返回响应,以向计算设备110指示该修改事务被执行完成。被记录在脏缓冲区中的数据,通过后台程序被最终写入磁盘中,以完成计算设备110磁盘中的与该元数据150对应的存储空间的修改。

[0059] 以此方式,当存在尚未完成的其他修改事务时,修改事务将被按照时间顺序记录在计算设备110的日志存储装置120中(当日志存储装置120中存在可用的存储空间时)。以此保证针对同一元数据150的修改事务,无论在任何情况下,均可以按照元数据修改事件发生的时间顺序被依次执行。

[0060] 根据本公开的一些示例实施方式,计算设备110进一步地包括维护修改记录的操作。具体为,计算设备110在运行时,确定修改事务的执行状态,并基于确定的执行状态,动态地更新元数据150的修改记录。

[0061] 进一步地,发明人注意到,由于当计算设备110直接修改元数据存储装置130中的元数据150时,不涉及修改事务被延时执行的问题。因此,可以仅在计算设备110需要生成日志140时才执行维护修改记录的操作。

[0062] 根据本发明的一些示例实施例,计算设备110可以在生成日志140或者在完成内存/高速缓存中的元数据150更新操作之后,确定该修改事务尚未被执行完成,根据所确定的尚未被执行完成的状态更新修改记录。相应地,当计算设备110完成磁盘中的元数据150的更新操作后,确定该修改事务已经被执行完毕。根据所确定的已经被执行完成的状态,计算设备110移除日志存储装置120中的与该修改事务相对应的日志140,并且执行更新修改记录的操作。

[0063] 以此方式,尚未完成的修改事务的执行状态可以被动态地实时地更新。因此即使计算设备110直接修改元数据存储装置130中的元数据150,依然可以保证修改事务按照时间顺序被依次执行。

[0064] 为进一步说明本公开的解决方案,将结合图3描述本公开的一些特定应用示例。应当理解的是,结合图3所描述的特定应用示例仅出于说明的目的被提出,不应理解为对本公开的限制。

[0065] 图3示出了能够在其中实现本公开的一些实施例的存储系统100的示意框图。如图3所示,存储系统100包括计算设备110,其作为存储系统的核心处理设备,用于执行存储系统100中的各种逻辑操作。计算设备110进一步包括主机310、日志存储装置120、修改记录存储装置320以及元数据存储装置130-1和130-2。在本特定实施例中,元数据存储装置130-1为内存/高速缓存,元数据存储装置130-2为磁盘。进一步地,如图3所示,日志存储装置120存储有与修改事务有关的日志140-1、140-2、……、140-N,元数据存储装置130-1和130-2存

储有元数据150-1、150-2、……、150-M,以及修改记录存储装置320存储有修改记录322-1-322-2、……、322-L(以下统称为修改记录322),其中L为修改记录的数目。计算设备还包括元数据刷新引擎330,其用于辅助实现内存中元数据150的刷新操作。

[0066] 应当理解,虽然图3示出了特定数目的计算设备110、日志存储装置120、元数据刷新引擎330、主机310、修改记录存储装置320以及元数据存储装置130,日志存储装置120存储有特定数目的日志140,元数据存储装置130存储有特定数目的元数据150,修改记录存储装置存储有特定数据的修改记录322,在其他的一些示例实施例中,计算设备、日志存储装置、元数据存储装置、元数据刷新引擎、主机、修改记录存储装置、日志、修改记录以及元数据的数量可能发生变化,本公开的范围在此方面不受限制。

[0067] 还应当理解,计算设备110、日志存储装置120、元数据刷新引擎330、主机310、修改记录存储装置320以及元数据存储装置130之间的布置和连接关系也不限于图3所示出的特定布置和连接关系。此外,上述部件在其他实施例中,可以进行拆分或组合。

[0068] 图3还进一步示出了计算设备110在处理元数据时的操作时序。应当理解,该特定操作时序仅作为示例被提出,在其他实施例中,操作可以被添加或省略,操作的执行顺序也可以被改变。

[0069] 在一应用示例中,计算设备110的主机310接收到针对元数据150的修改事务,例如,修改事务为对元数据150-1进行修改。主机310获取与计算设备110的繁忙程度有关的信息。在本特定实施例中,主机310可以获取指示日志存储设备120的存储资源使用信息。计算设备110根据获取的信息确定计算设备110的繁忙状态。如果主机310确定计算设备110的繁忙程度高于阈值程度,则主机310从修改记录存储装置320获取350修改记录322。在该特定实施例中,修改记录322-1可以用于指示针对元数据150-1的修改记录。如果修改记录322-1指示不存在尚未完成的、针对元数据150-1的其他修改事务,则主机310直接修改352元数据存储装置130-1(即,内存/高速缓冲)中的元数据150-1,以及进一步修改354元数据存储装置130-2(即,磁盘)中的元数据150-1。随后,执行该修改事务的相应的进程向主机310返回356响应,以指示针对元数据150-1的修改事务被执行完成。

[0070] 在另一应用示例中,计算设备110的主机310接收到针对元数据150的修改事务,例如,修改事务为对元数据150-2进行修改。主机310获取与计算设备110的繁忙程度有关的信息。在本特定实施例中,主机310可以获取指示日志存储设备120的存储资源使用信息,并根据获取的信息确定计算设备110的繁忙状态。如果主机310确定计算设备110的繁忙程度不高于阈值程度,则主机310首先在日志记录装置120中生成日志140。在该特定实施例中,主机310在日志装置120中生成360日志140-1,日志140-1用于记录该针对元数据150-2的修改事务。执行该修改事务的相应的进程随后修改362元数据存储装置130-1(即,内存/高速缓冲)中的元数据150-2,该操作可以通过元数据刷新引用330刷新364元数据存储装置130-1中的元数据150-2来辅助实现。此时,针对元数据150-2的修改事务被确定为处于尚未完成的状态,根据该确定的状态,执行该修改事务的相应的进程更新修改记录存储装置320中的修改记录322。在该特定实施例中,修改记录322-2用于指示针对元数据150-2的修改事务的执行状态。执行该修改事务的相应的进程将更新修改记录322-2,以使得修改记录322-2指示存在尚未完成的、针对元数据150-2的其他修改事务。随后,主机310接收到368响应。

[0071] 在元数据存储装置130-1中的更新后的元数据150-2将被存储至脏缓存区。被存储

在脏缓冲区中的元数据,可以通过后台程序被最终写入370元数据存储装置130-2(即,磁盘)中,以完成元数据存储装置130-2中的与该元数据150-2对应的存储空间的修改。

[0072] 最后,执行该修改事务的相应的进程移除372日志存储装置120中的与该修改事务相对应的日志140(即,日志140-1),并且执行更新374修改记录322(即,修改记录322-2)的操作。

[0073] 图4示出了可以用来实施本公开的实施例的示例设备400的示意性框图。设备400可以被实现为图1的计算设备110。设备400可以用于实现图2的过程200。

[0074] 如图所示,设备400包括中央处理单元(CPU)401,其可以根据存储在只读存储器(ROM)402中的计算机程序指令或者从存储单元408加载到随机访问存储器(RAM)403中的计算机程序指令,来执行各种适当的动作和处理。在RAM 403中,还可存储设备400操作所需的各种程序和数据。CPU 401、ROM 402以及RAM 403通过总线404彼此相连。输入/输出(I/O)接口405也连接至总线404。

[0075] 设备400中的多个部件连接至I/O接口405,包括:输入单元406,例如键盘、鼠标等;输出单元407,例如各种类型的显示器、扬声器等;存储单元408,例如磁盘、光盘等;以及通信单元409,例如网卡、调制解调器、无线通信收发机等。通信单元409允许设备400通过诸如因特网的计算机网络和/或各种电信网络与其他设备交换信息/数据。

[0076] 处理单元401执行上文所描述的各个方法和处理,例如过程200。例如,在一些实施例中,过程200可以被实现为计算机软件程序或计算机程序产品,其被有形地包含于机器可读介质,诸如非瞬态计算机可读介质,诸如存储单元408。在一些实施例中,计算机程序的部分或者全部可以经由ROM 402和/或通信单元409而被载入和/或安装到设备400上。当计算机程序加载到RAM 403并由CPU 401执行时,可以执行上文描述的过程200的一个或多个步骤。备选地,在其他实施例中,CPU 401可以通过其他任何适当的方式(例如,借助于固件)而被配置为执行过程200。

[0077] 本领域的技术人员应当理解,上述本公开的方法的各个步骤可以通过通用的计算装置来实现,它们可以集中在单个的计算装置上,或者分布在多个计算装置所组成的网络上,可选地,它们可以用计算装置可执行的程序代码来实现,从而可以将它们存储在存储装置中由计算装置来执行,或者将它们分别制作成各个集成电路模块,或者将它们中的多个模块或步骤制作成单个集成电路模块来实现。这样,本公开不限制于任何特定的硬件和软件结合。

[0078] 应当理解,尽管在上文的详细描述中提及了设备的若干装置或子装置,但是这种划分仅仅是示例性而非强制性的。实际上,根据本公开的实施例,上文描述的两个或更多装置的特征和功能可以在一个装置中具体化。反之,上文描述的一个装置的特征和功能可以进一步划分为由多个装置来具体化。

[0079] 以上所述仅为本公开的可选实施例,并不用于限制本公开,对于本领域的技术人员来说,本公开可以有各种更改和变化。凡在本公开的精神和原则之内,所作的任何修改、等效替换、改进等,均应包含在本公开的保护范围之内。

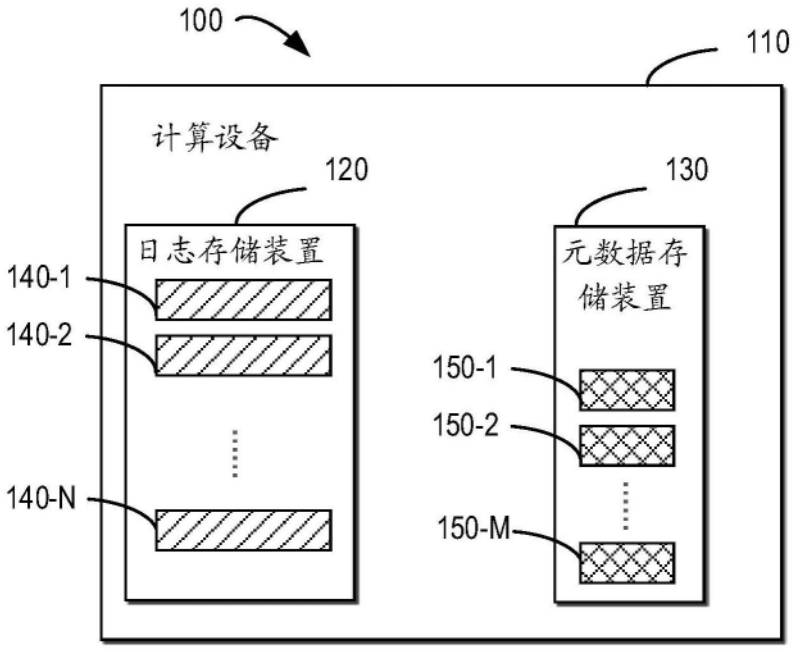

100
110
计算设备
120
130
日志存储装置
元数据存储装置
140-1
140-2
140-N
150-1
150-2
150-M

图1

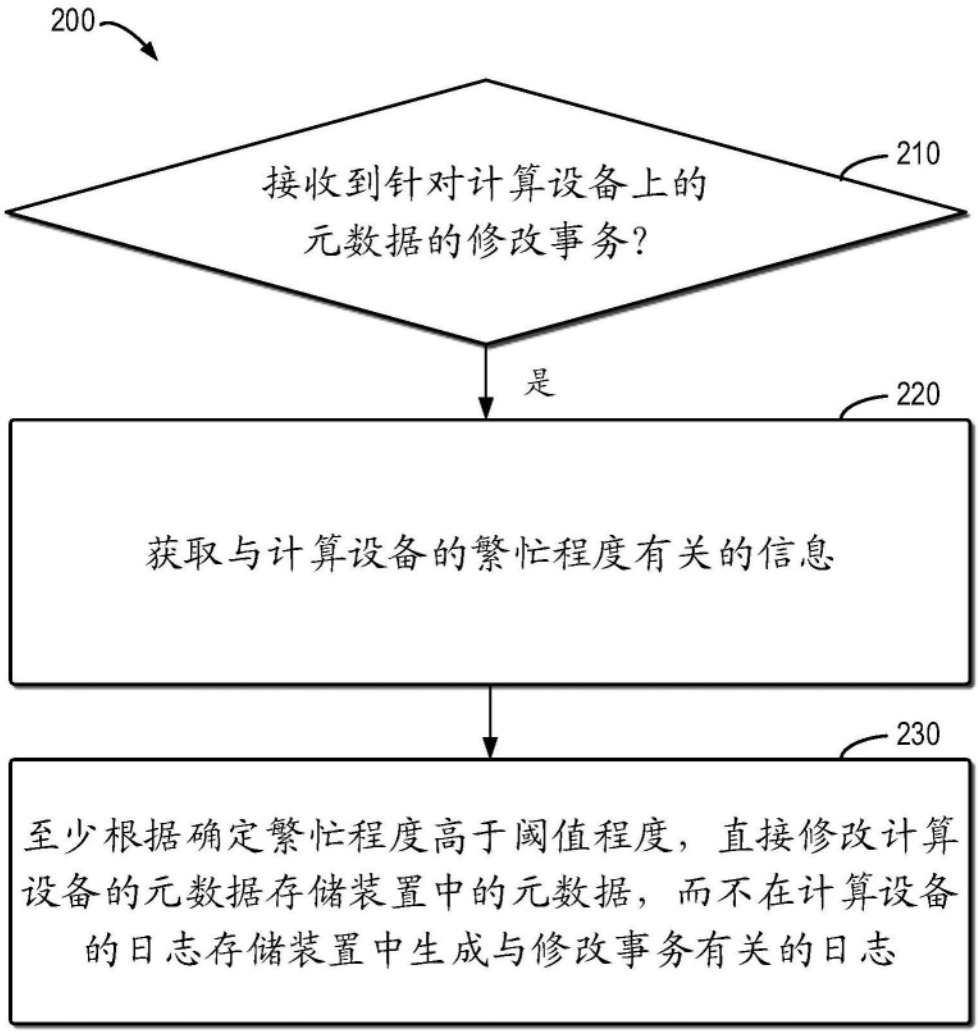

200
接收到针对计算设备上的元数据的修改事务?
210
是
220
获取与计算设备的繁忙程度有关的信息
230
至少根据确定繁忙程度高于阈值程度，直接修改计算设备的元数据存储装置中的元数据，而不在计算设备的日志存储装置中生成与修改事务有关的日志

图2

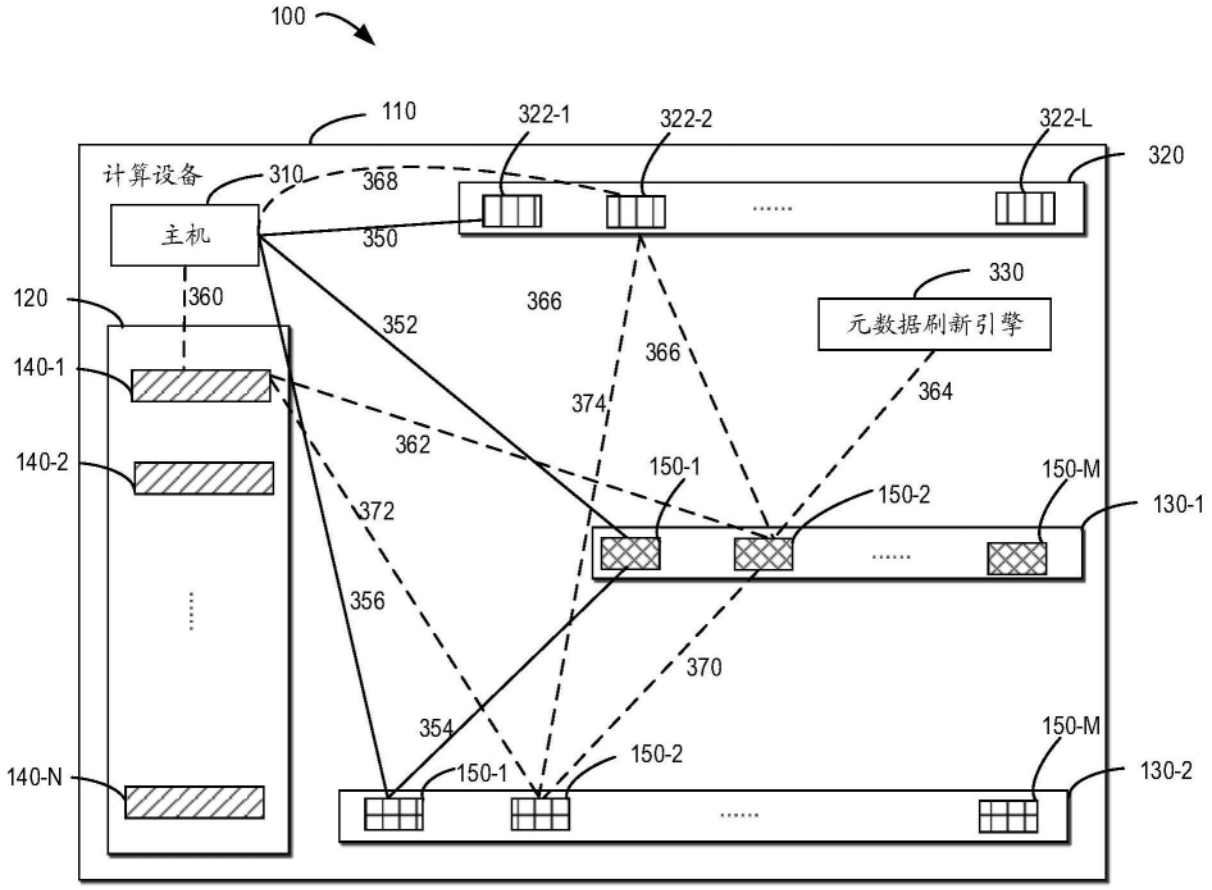

100
110
322-1
322-2
322-L
320
计算设备
310
368
主机
350
330
元数据刷新引擎
120
360
366
352
366
140-1
374
364
362
140-2
150-1
150-2
150-M
130-1
372
356
370
354
150-M
150-1
150-2
140-N
130-2

图3

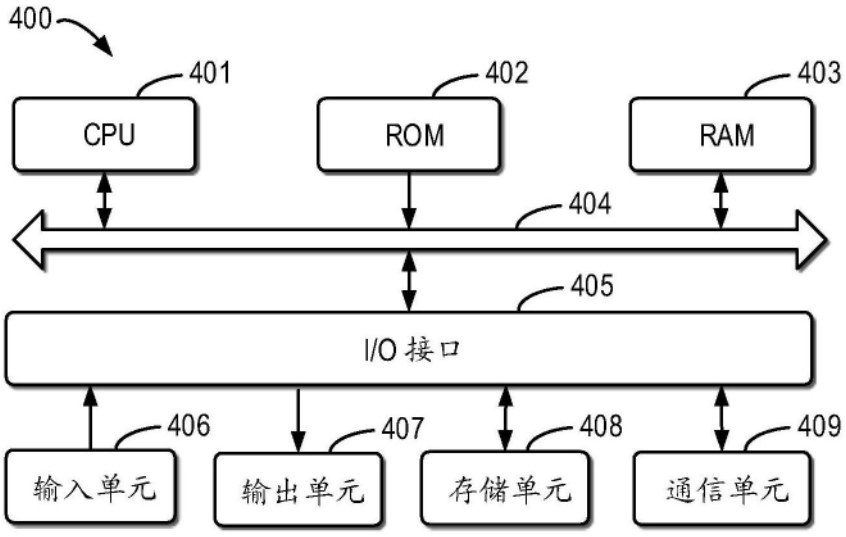

400
401
402
403
CPU
ROM
RAM
404
405
I/O 接口
406
407
408
409
输入单元
输出单元
存储单元
通信单元

图4