



(12) 发明专利申请

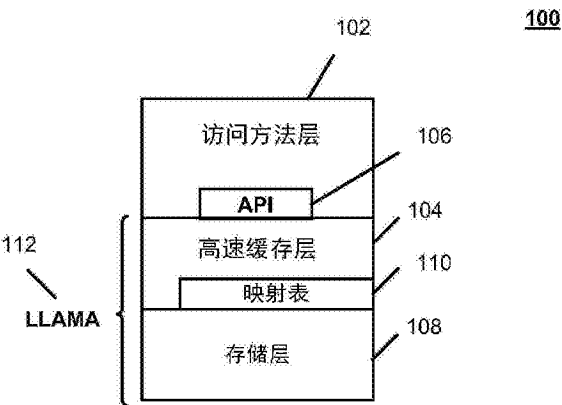
(10) 申请公布号 CN 105408895 A
(43) 申请公布日 2016. 03. 16

(21) 申请号 201480035652. 7
(22) 申请日 2014. 06. 20
(30) 优先权数据
13/924, 567 2013. 06. 22 US
(85) PCT国际申请进入国家阶段日
2015. 12. 22
(86) PCT国际申请的申请数据
PCT/US2014/043299 2014. 06. 20
(87) PCT国际申请的公布数据
W02014/205298 EN 2014. 12. 24
(71) 申请人 微软技术许可有限责任公司
地址 美国华盛顿州
(72) 发明人 D·B·洛美特 J·莱万多斯基
S·森古普塔
(74) 专利代理机构 上海专利商标事务所有限公司 31100
代理人 陈小刚
(51) Int. Cl.
G06F 17/30(2006. 01)

权利要求书2页 说明书25页 附图14页

(54) 发明名称
用于多个访问方法的无锁存、日志结构化的
存储
(57) 摘要

数据管理器可包括配置成向任意选择的面向
页的访问方法提供对页数据存储的接口访问的数
据不透明接口,该接口访问包括对页数据存储的
无锁存访问。在另一方面,可基于发起部分交换增
量记录到与第一页相关联的页状态的前缀,来发
起高速缓存层存储中的所述第一页的一部分到辅
助存储中的位置的交换操作,部分交换增量记录
包括指示转储清除增量记录的存储位置的主存储
器地址,转储清除增量记录指示所述第一页的丢
失部分在辅助存储中的位置。在另一方面,页管理
器可基于对转储清除增量记录的原子操作来发起
高速缓存层存储中的第一页到辅助存储中的位置
的转储清除操作。



1. 一种系统,包括:

包括至少一个处理器的设备,所述设备包括数据管理器,所述数据管理器包括被有形地体现在计算机可读存储介质上以供所述至少一个处理器执行的指令,所述数据管理器包括:

配置成向任意选择的面向页的访问方法提供对页数据存储的接口访问的数据不透明接口,所述对页数据存储的接口访问包括对所述页数据存储的无锁存访问。

2. 如权利要求 1 所述的系统,其特征在于:

所述数据不透明接口被配置成向所述任意选择的面向页的访问方法提供对页数据存储的接口访问,所述对页数据存储的接口访问包括对所述页数据存储的日志结构化访问。

3. 如权利要求 1 所述的系统,其特征在于,进一步包括:

高速缓存层管理器,所述高速缓存层管理器包括:

配置成在与所述数据不透明接口相关联的间接地址映射表上发起表操作的映射表管理器,所述表操作包括在所述间接地址映射表中的各条目上发起原子比较并交换操作,以用与所述页数据存储相关联的页的新状态来替换所述页的先前状态。

4. 如权利要求 3 所述的系统,其特征在于:

所述映射表管理器被配置成在与所述数据不透明接口相关联的所述间接地址映射表上发起表操作,其中所述间接地址映射表被共同用于对包括高速缓存层存储和辅助存储的数据存储的管理。

5. 如权利要求 3 所述的系统,其特征在于:

所述间接地址映射表将页的逻辑位置与所述页的对应物理位置分开,其中:

所述页数据存储的用户根据所述页的物理位置地址值来将页标识符值存储在引用所述页数据存储的数据结构中的其他位置。

6. 如权利要求 3 所述的系统,其特征在于,进一步包括:

日志结构化存储层管理器,被配置成使用针对所述间接地址映射表中的条目的无锁存比较并交换操作来控制与得自页转储清除的日志结构化相关联的页位置变化。

7. 如权利要求 1 所述的系统,其特征在于,进一步包括:

配置成基于将转储清除增量记录的指针安置到映射表中经由比较并交换(CAS)操作来将页状态转储清除到辅助存储的页管理器,所述转储清除增量记录被前缀到经由所述CAS操作在所述映射表中被替换的现有页状态。

8. 如权利要求 7 所述的系统,其特征在于:

所述页管理器被配置成:

确定所述CAS操作是否成功,并在确定所述CAS操作成功的情况下发起写操作以将所述现有页状态写入辅助存储转储清除缓冲区。

9. 一种系统,包括:

包括至少一个处理器的设备,所述设备包括数据管理器,所述数据管理器包括被有形地体现在计算机可读存储介质上以供所述至少一个处理器执行的指令,所述数据管理器包括:

配置成基于以下操作来发起高速缓存层存储中的第一页到辅助存储中的位置的转储清除操作的页管理器:

发起所述第一页的页状态到辅助存储缓冲区的复制，

发起转储清除增量记录到所述页状态的前缀，所述转储清除增量记录包括指示所述第一页在辅助存储中的存储位置的辅助存储地址以及与调用者相关联的注释，以及

基于将所述转储清除增量记录的地址安置在映射表中经由比较并交换（CAS）操作来发起对所述页状态的更新。

10. 一种系统，包括：

包括至少一个处理器的设备，所述设备包括数据管理器，所述数据管理器包括被有形地体现在计算机可读存储介质上以供所述至少一个处理器执行的指令，所述数据管理器包括：

页管理器，所述页管理器被配置成基于发起部分交换增量记录到与第一页相关联的页状态的前缀，来发起高速缓存层存储中的所述第一页的一部分到辅助存储中的位置的交换操作，所述部分交换增量记录包括指示转储清除增量记录的存储位置的主存储器地址，所述转储清除增量记录指示所述第一页的丢失部分在辅助存储中的位置。

用于多个访问方法的无锁存、日志结构化的存储

[0001] 背景

[0002] 电子设备的用户常常需要访问数据库系统以获得各种类型的信息。许多不同的技术已被设计用于对数据项的存储和检索。例如,一些最近硬件平台利用最近硬件发展,如多核处理器、多级存储器分层结构以及辅助存储设备(如闪存),来力图提供更高性能。这提高了潜在的系统性能,但对系统而言,对新开发的平台方面以及常规平台方面作出高效使用是困难的。

[0003] 概述

[0004] 根据一个一般方面,一种系统可包括包含至少一个处理器的设备,该设备包括包含有形地体现在计算机可读存储介质上以供至少一个处理器执行的指令的数据管理器。数据管理器可包括配置成向任意选择的面向页的访问方法提供对页数据存储的接口访问的数据不透明接口,该接口访问包括对页数据存储的无锁存访问。

[0005] 根据另一方面,一种系统可包括包含至少一个处理器的设备,该设备包括包含有形地体现在计算机可读存储介质上以供至少一个处理器执行的指令的数据管理器。数据管理器可包括配置成基于安置到映射表中的转储清除增量(Δ)记录的指针经由比较并交换(CAS)操作来将页状态转储清除到辅助存储的页管理器,转储清除增量记录被前缀到经由CAS操作在映射表中被替换的现有页状态。

[0006] 根据另一方面,一种系统可包括包含至少一个处理器的设备,该设备包括包含有形地体现在计算机可读存储介质上以供至少一个处理器执行的指令的数据管理器。数据管理器可包括配置成基于以下操作来发起高速缓存层存储中的第一页到辅助存储中的位置的转储清除操作的页管理器:发起所述第一页的页状态到辅助存储缓冲区的复制,发起转储清除增量记录到所述页状态的前缀,所述转储清除增量记录包括指示所述第一页在辅助存储中的存储位置的辅助存储地址以及与调用者相关联的注释,以及基于将所述转储清除增量记录的地址安置在映射表中经由比较并交换(CAS)操作来发起对所述页状态的更新。

[0007] 根据另一方面,一种系统可包括包含至少一个处理器的设备,该设备包括包含有形地包括在计算机可读存储介质上以供至少一个处理器执行的指令的数据管理器。数据管理器可包括配置成经由无锁存更新操作来控制对日志结构化辅助存储缓冲区的更新的缓冲区管理器。

[0008] 根据另一方面,一种系统可包括包含至少一个处理器的设备,该设备包括包含有形地包括在计算机可读存储介质上以供至少一个处理器执行的指令的数据管理器。数据管理器可包括页管理器,所述页管理器被配置成基于发起部分交换增量记录到与第一页相关联的页状态的前缀,来发起高速缓存层存储中的所述第一页的一部分到辅助存储中的位置的交换操作,所述部分交换增量记录包括指示转储清除增量记录的存储位置的主存储器地址,所述转储清除增量记录指示所述第一页的丢失部分在辅助存储中的位置。

[0009] 提供本概述以便以简化形式介绍将在以下详细描述中进一步描述的一些概念。本发明内容并不旨在标识所要求保护主题的关键特征或必要特征,也不旨在用于限制所要求保护主题的范围。一个或多个实现的细节在以下的附图和说明书中阐述。其它特征将从描

述和附图、以及权利要求书中显见。

[0010] 附图

[0011] 图 1 解说用于高速缓存 / 存储层的访问方法的示例体系结构分层。

[0012] 图 2 是用于多种访问方法的无锁存、日志结构化存储的示例体系结构的框图。

[0013] 图 3 示出示例映射表。

[0014] 图 4a-4b 示出示例映射表上的示例增量更新。

[0015] 图 5 描绘示例部分页换出以及示例部分交换增量。

[0016] 图 6 示出各示例时期以及它们相应的垃圾收集列表。

[0017] 图 7a-7c 示出了在闪存上的示例日志结构化存储组织。

[0018] 图 8 描绘了示例转储清除缓冲区状态。

[0019] 图 9 示出了示例事务模板。

[0020] 图 10 示出了示例检查点数据。

[0021] 图 11 是用于多种访问方法的无锁存、日志结构化存储的示例系统的框图。

[0022] 图 12a-12d 是示出图 11 的系统的示例操作的流程图。

[0023] 详细描述

[0024] I. 介绍

[0025] 硬件平台的最近发展利用了多核处理器、多级存储器分层结构以及辅助存储设备（如闪存）来力图提供更高性能。例如，中央处理单元（CPU）变化包括了多核处理器以及涉及多级高速缓存的主存储器访问。例如，闪存存储和硬盘供应商认识到就地更新损害了容量，导致对日志结构化的增加的使用。例如，云数据中心提高了系统规模，且商用硬件的使用日益着重于高可用性技术。

[0026] 然而，尽管潜在系统性能可能提高，但对系统而言，对这些最近平台方面作出高效使用是困难的。例如，支持多个用户访问大量数据的数据中心系统可能利用针对过去多年来它存在于其上的硬件来设计的软件体系结构（例如，它们可能以单处理器为目标，操作单级存储器（极少处理器高速缓存，且只具有对主存储器的适度等待时间），以及访问磁盘）。

[0027] 改变该方法的努力已经改进了该环境，但继续遗漏了显著的潜在性能增益。例如，已经有努力来避免锁存，锁存在对数据的访问相冲突时造成阻塞；然而，这些努力可能涉及分区，使得各线程避免这样的冲突，这可引入显著开销。例如，就地更新数据可能对存储器性能产生负面影响，这可导致考虑高速缓存行对齐并使用逻辑树代替向量上的二分搜索。然而，继续存在不利数量的就地更新，这不利地影响处理器高速缓存性能，例如经由高速缓存无效。此外，各实现已开始利用闪存来用于其更高访问 / 辅助以及降低的访问等待时间。然而，随机更新可能是相对昂贵的，甚至使用闪存转换层。

[0028] J. Levandoski 等人的“Deuteronomy: Transaction Support for Cloud Data (Deuteronomy: 对云数据的事务支持)”，Conference on Innovative Data Systems Research (新数据系统研究会议) (CIDR) (2011 年 1 月)，第 123-133 页以及 D. Lomet 等人的“Unbundling Transaction Services in the Cloud (云中的分别处理事务服务)”，Conference on Innovative Data Systems Research (CIDR)，2009 年，讨论了用于提供云环境中的一致性（即，事务）的示例技术。本文讨论的示例技术可集中于示例 DEUTERONOMY

数据组件 (DC) 以及当前硬件上最大化其性能。例如,DC 可以管理对经由 CRUD(创建、读、更新、删除)原子操作访问的数据的存储和检索。例如,DC 可以是非分布式的,代替使用可经由其上的软件层被合并到分布式系统的本地机制(例如,DEUTERONOMY 事务组件(TC)和/或查询引擎)。

[0029] 如本文进一步讨论的,看来存在由当前硬件提出的可能影响访问方法(例如,B 树、散列、多属性、时间,等)的问题。此外,如本文讨论的,这些问题可以用适用于大多数(例如,任意选择的)访问方法的示例一般机制来解决。

[0030] 例如,根据本文讨论的示例技术,无锁存技术可被用来达到有利的处理器利用以及针对多核处理器进行伸缩。例如,如本文讨论的,降低高速缓存无效的增量更新可被用来达到针对基于多级高速缓存的存储器系统的有利性能。例如,写受限存储(其具有随机写的受限性能和闪存写限制)可经由日志结构化来被克服。

[0031] 例如,BW 树(参见例如 J. Levandoski 等人的“The Bw-Tree: A B-tree for New Hardware Platforms(Bw-树:用于新硬件平台的 B 树)”,第 29 届 IEEE International Conference on Data Engineering(IEEE 数据工程国际会议)(ICDE2013),2013 年 4 月 8-11 日,有点类似于 B 树(参见例如 R. Bayer 等人的“Organization and Maintenance of Large Ordered Indices(大型有序索引的组织和维护)”,Acta Informatica(Acta 信息学),卷 1,第 3 期,1972 年,第 173-189 页,以及 D. Comer 的“The Ubiquitous B-tree(无所不在的 B 树)”,ACM Computing Surveys(ACM 计算综述)(CSUR),卷 11,第 2 期,1979 年 6 月,第 121-137 页)的索引,是可利用这些示例技术的 DC 或键-值存储的示例。示例 BW 树可涉及用于更一般地达成无锁存和日志结构化的技术的范例。根据本文讨论的示例技术,无锁存和日志结构技术可被实现在能支持多种访问方法的高速缓存/存储子系统中,有点类似于常规高速缓存/存储子系统可处理对被写回盘来作为就地更新的固定大小页的锁存访问。

[0032] 根据本文讨论的示例技术,本文中可被称为 LLAMA(无锁存、日志结构化访问方法知晓)的示例系统包括用于(至少)最近开发的硬件环境(例如,闪存、多核)的高速缓存和存储子系统,但数据处理领域的技术人员将理解,这样的示例技术不仅限于最近开发的硬件。

[0033] 例如,LLAMA 可以支持用于任意选择的面向页的访问方法的应用编程接口(API),该 API 提供高速缓存和存储管理两者,从而优化处理器高速缓存和辅助存储。例如,高速缓存(CL)和存储(SL)层可使用将页的逻辑和物理位置分开的共同映射表。例如,高速缓存层(CL)可经由其映射表上的无锁存比较并交换原子状态变化来支持数据更新和管理更新(例如,用于索引再组织)。

[0034] 例如,存储层(SL)可以使用同一映射表来处理由每一页转储清除时的日志结构化所产生的页位置变化。例如,无锁存 BW 树实现(例如,使用 BW 树的实现,作为有序 B 树型索引的示例)可被使用。在这一上下文中,“转储清除”操作可以指通过将页复制到输出缓冲区来将该页从主存储器(例如,高速缓存存储)传送到辅助存储。

[0035] 本文讨论的示例技术可以提供可将页的位置和大小两者虚拟化的映射表。例如,这样的虚拟化可被用于主存储器设计和稳定存储设计(例如,日志结构化的存储设计)两者,如本文进一步讨论的。

[0036] 在这一上下文中,“页”可以指存储中的对象,这可以经由物理存储地址来访问。如

本文所使用的,“页”可以与灵活的大小相关联,且可表示分布在多个不毗连地存储的存储段上的存储的页单位。存储可包括易失性和 / 或稳定存储。

[0037] 本文讨论的示例技术可以将访问方法层与高速缓存 / 存储管理分开。作为示例,本文讨论的技术可被用来实施预写日志协议。例如,在将页转储清除之前,常规数据库内核可检查页日志序列号 (LSN) 来确定是否存在尚未被稳定在事务日志中的更新。例如,LLAMA 高速缓存管理可以利用示例增量更新来“换出”部分页。例如,它可以从高速缓存丢弃该页的已存在于辅助存储 (它不包括最近增量更新) 上的那一部分。例如,访问方法层将定期转储清除以用于事务日志检查点。因而,高速缓存管理器将找出足够的候选 (可能是部分) 页来满足任何缓冲区大小约束。

[0038] 本文讨论的示例技术可以提供允许大量访问方法 (即,并非单个实例) 通过实现提供这些技术的子系统层来利用这些技术的框架。此外,日志结构化存储可被实现以用于将数据写入提供有利效率的辅助存储。因此,访问方法可集中于其索引的主存储器方面,且本文讨论的示例技术可以提供该框架以用于达成与 BW 树的性能度量类似的性能度量。

[0039] 例如,诸如 LLAMA 等技术通过其 API 可以提供无锁存页更新,这在主存储器中经由针对映射表的比较并交换 (CAS) 原子操作来实现。

[0040] 例如,在管理高速缓存时,诸如 LLAMA 等技术可以通过仅丢弃从存储器中先前已转储清除的页部分来回收主存储器,因而不涉及任何输入 / 输出 (I/O) 操作,甚至在换出“脏”页时。因而,诸如 LLAMA 等技术可能控制其缓冲区高速缓存存储器大小而无需来自其访问方法用户的输入。

[0041] 例如,为了高效地管理辅助存储,诸如 LLAMA 等技术可以利用日志结构化。例如,与常规日志结构化相比,诸如 LLAMA 等技术可以通过使用部分页转储清除以及具有基本上非空空间 (即,基本上 100% 存储利用率) 的页来改进性能。这些可以在将页转储清除时降低输入 / 输出操作 (I/O) 的数量以及每页消耗的存储量,并且因此可降低在使用日志结构化时体验到的写放大。此外,所有存储相关操作可以是完全无锁存的。

[0042] 例如,诸如 LLAMA 等技术可以提供 (至少) 有限形式的系统事务。在这一意义上,系统事务不是用户级事务,而是相反,利用日志结构化存储来提供原子性,纯用于访问方法的“私有使用” (例如,对于索引结构修改 (SMO))。例如,这可允许索引在并发更新继续的同时随着它们增加来适配。

[0043] 例如,BW 树结构可包括一种类型的无锁存 B 树结构。例如,对 BW 树节点的更新可以基于将更新增量前缀到先前页状态来执行。因而,BW 树可以是无锁存的,因为它可允许多个线程对各页的并发访问。因为这样的增量更新保留了页的先前状态,所以它也可以提供经改进的处理器高速缓存性能。

[0044] 使用 BW 树的示例技术可进一步提供也是无锁存且可采用 B 链路树型侧指针的页拆分技术。拆分 (以及其他结构修改操作) 可以是在主存储器内以及在稳定时两者情况下是原子的。例如,原子记录存储可以基于 BW 树体系结构来实现。

[0045] 数据处理领域技术人员将明白,可存在许多完成本文讨论的无锁存和日志结构化存储的多种方式,而不背离本文讨论的精神。

[0046] II. 示例操作环境

[0047] 在此讨论的特征被提供为可用数据处理领域的技术人员所理解的多种不同方式

来实现,而不背离在此的讨论的精神。这样的特征被解释为仅为示例实施例特征,并且不旨在被解释为限制到仅为那些详细的描述。

[0048] 图1解说用于高速缓存/存储层的访问方法的示例体系结构分层。访问方法层102是顶层,如图1所示。访问方法层102与作为中间层的高速缓存层104交互。应用编程接口(API)106可被用于访问方法层102与高速缓存层104之间的活动。示例存储层108可以与映射表110交互,映射表110在高速缓存层104与存储层108之间共享。例如,LLAMA 112包括高速缓存层104和存储层108。例如,存储层可以支持日志结构化闪存存储。根据本文讨论的示例技术,日志结构化存储可以管理闪存和盘存储两者。这一设计可在体系结构上兼容现有数据库内核,同时还适于作为独立或DEUTERONOMY型原子记录存储(ARS)。

[0049] 例如,诸如LLAMA等技术可支持页抽象,从而支持针对高速缓存/存储层的访问方法实现。此外,事务组件(例如,DEUTERONOMY型事务组件)可被添加到顶部。图2是用于多种访问方法的无锁存、日志结构化存储的示例体系结构的框图。如图2所示,事务组件202可以支持事务键-值存储,且可与可包括原子键-值存储的数据组件204一起操作。如图2所示,数据组件204可包括无锁存有序索引206和/或无锁存线性散列索引208。如图2所示,数据组件204可进一步包括示例无锁存、日志结构化的、访问方法知晓(LLAMA)的存储引擎210(例如,图1的LLAMA 112)。

[0050] 示例API 106可以是“数据不透明”的,意味着示例LLAMA实现没有“看到”(例如,没有检查、或分析、或依赖于)(例如,访问方法层102的)访问方法将什么放入各页或增量记录,且独立于访问方法在各页或增量记录中提供的事物来动作。因而,本文讨论的示例LLAMA实现可以响应于不依赖于访问方法所提供的事物的特定操作来动作,如上所述。

[0051] 如图3所示,页302可经由将页标识符(PID)306映射到主存储器高速缓存312中或辅助存储314上的状态308(例如,经由存储在映射表304中的“物理地址”310)的映射表304被访问。例如,主存储器高速缓存312可包括随机存取存储器(RAM)。例如,辅助存储314可包括闪存存储器。例如,页302可以按需从辅助存储314读入主存储器高速缓存312,它们可被转储清除到辅助存储314,且它们可在处于高速缓存312中时被更新以改变页状态。例如,根据本文讨论的示例技术,基本上所有页状态变化(数据状态和管理状态两者)可被提供作为原子操作。如图3所示,示例物理地址310可包括指示物理地址与闪存还是存储器(例如,高速缓存)存储相关联的闪存/存储器标志316(例如,1位,如在该示例中所示),具有用于(至少)地址本身的地址字段318(例如,63位,如在该示例中所示)。数据处理领域的技术人员将明白,存在用于表示“物理地址”(例如,不同于64位表示)的多种方式,而不背离本文讨论的精神。

[0052] 根据本文讨论的示例技术,LLAMA通过其API可以经由针对映射表304的比较并交换(CAS)原子操作(例如,根据保护页免于被阻塞线程并发访问的常规锁存)来提供无锁存页更新。例如,CAS策略可有利地提高存储器利用率并改进多核伸缩。

[0053] 根据本文讨论的示例技术,在管理高速缓存时,LLAMA可通过只从存储器丢弃先前转储清除的页部分来收回主存储器,因而不使用任何I/O,甚至在换出“脏”页时。因而,诸如LLAMA等示例体系结构可控制其缓冲区高速缓存存储器大小,而不必检查由其访问方法用户存储在各页中的数据(例如,在诸如LLAMA等示例体系结构不知晓事务和预写记入日志时)。

[0054] 诸如 LLAMA 等示例体系结构可以使用日志结构化来管理辅助存储（例如，提供避免随机写的优点，从而降低经由大型多页缓冲区的写的数量，以及使用闪存存储器涉及的磨损平衡）。此外，诸如 LLAMA 等示例体系结构可以使用部分页转储清除以及具有基本上非空空间（即，基本上 100% 利用率）的页来改进性能（与常规日志结构化相比）。例如，这些可以在将页转储清除时降低 I/O 的数量以及每页消耗的存储，并且因此可降低在使用日志结构化时以其他方式体验到的写放大。此外，基本上所有存储相关操作可以是完全无锁存的。

[0055] 另外，诸如 LLAMA 等示例体系结构可以支持（至少）有限形式的系统事务（参见例如 D. Lomet 等人的“Unbundling Transaction Services in the Cloud”，Conference on Innovative Data Systems Research (CIDR)，2009 年，参见系统事务）。例如，系统事务可能不是用户事务，而是可提供原子性，纯用于访问方法的“私有使用”（例如，对于索引结构修改 (SMO)——参见例如 C. Mohan 等人的“ARIES/IM: An Efficient and High Concurrency Index Management Method Using Write-Ahead Logging (ARIES/IM: 使用预写记入日志的高效且高一致性的索引管理方法)”，1992 年 ACM SIGMOD 数据管理国际会议会议录 (SIGMOD'92)，1992 年，第 371-380 页）。例如，与事务日志分开记录的系统事务的属性可以是有效的是 DEUTERONOMY 方法对分解数据库内核的有利洞察的示例。

[0056] 以下讨论包括访问方法实现者在使用诸如 LLAMA 等示例体系结构时可能遇到的示例操作接口的进一步描述，进一步讨论了它可被如何使用。以下讨论进一步包括根据本文讨论的示例技术的示例高速缓存层的描述，以及日志结构化存储层的示例设计。此外，提供了根据本文讨论的示例技术的相对于示例系统事务机制和示例测量的讨论，该测量可被采取来提供原子性。此外，提供了根据本文讨论的示例技术的相对于从系统崩溃进行示例日志结构化存储恢复的讨论。

[0057] 在设计诸如 LLAMA 等示例系统时，设计目标可包括尽可能“通用”的目标，这有时可导致尽可能“低级”的目标。然而，为使诸如 LLAMA 等示例系统“通用”，在尽可能少地知晓访问方法在使用其工具时做什么的同时高效地操作可能是合乎需要的。因而，诸如 LLAMA 等示例系统的操作可以是“基本的”，以高速缓存管理和页的更新为目标。例如，诸如 LLAMA 等示例系统可包括一些附加工具以支持可被有利地包括以用于 SMO（例如，页拆分和归并）的基本事务机制。

[0058] 根据本文讨论的示例技术，诸如 LLAMA 等示例系统可在接口中不包括与日志序列号 (LSN)、预写记入日志或事务日志的检查点有关的事物。根据本文讨论的示例技术，诸如 LLAMA 等示例系统可不包括用于用户操作的幂等测试。此外，本文讨论的示例技术，诸如 LLAMA 等示例系统可不包括事务恢复（例如，根据本文讨论的示例技术，它可由访问方法使用诸如 LLAMA 等示例系统来处理）。

[0059] 根据本文讨论的示例技术，示例访问方法可以响应于用户操作来改变状态。例如，用户可能想要创建 (C)、读取 (R)、更新 (U)、或删除 (D) 记录（例如，CRUD 操作）。根据本文讨论的示例技术，诸如 LLAMA 等示例系统可不直接支持这些操作。相反，示例访问方法可以将它们实现成对 LLAMA 页的状态的更新。

[0060] 例如，还可能存在作为示例访问方法操作的一部分的结构变化。例如，BW 树页拆分可涉及将拆分增量张贴到原始页 0，使得搜索者知晓新页现在包含 0 中的键的子范围的

数据。例如,这些也可作为对 LLAMA 页 0 的更新被处理。

[0061] 根据本文讨论的示例技术,诸如 LLAMA 等示例系统可以支持两种形式的更新,例如增量更新以及替换更新。例如,访问方法可以根据用户的希望来选择利用这些形式的更新例如,BW 树可作出一系列增量更新并且在某时决定通过将增量更新应用于基础页来“合并”并优化该页。例如,BW 树随后可使用替换更新来生成新基础页。

[0062] 根据本文讨论的示例技术,诸如 LLAMA 等示例系统可在贯穿本文讨论的更新操作和替换操作期间保留与页在辅助存储中的位置有关的信息,使得系统 100 具有辅助存储页位置信息以在它被换出主存储器高速缓存并用于垃圾收集的情况下重新读取该页,如本文进一步讨论的。因而,系统 100 可记住先前页位置和稳定页状态信息。

[0063] 例如,增量更新可被指示为 Update-D(PID, in-ptr, out-ptr, data) (更新 D(PID, 输入指针, 输出指针, 数据))。例如,增量更新可将描述变化的增量前缀到该页的先前状态。例如,对于 BW 树,Update-D 的“data”参数可包括至少 <lsn, key, data>(<lsn, 键, 数据>), 其中 lsn 启用幂等。例如,“in-ptr”指向该页的先前状态,且“out-ptr”指向该页的新状态。

[0064] 例如,替换更新可被指示为 Update-R(PID, in-ptr, out-ptr, data)。例如,替换更新可造成该页的整个新状态。在使用 Update-D 时保留的先前状态可被“data”参数替换。因而,“data”参数包含“掺入”了增量的该页的整个状态。

[0065] 例如,“读取”可被指示为 Read(PID, out-ptr) (读取 (PID, 输出指针))。例如,读取可经由“out-ptr”返回该页的在主存储器中的地址。如果该页不在主存储器中,则映射表条目可包含辅助存储地址。例如,在这种情况下,该页可被读入主存储器且映射表可被用新主存储器地址来更新。

[0066] 除支持数据操作之外,本文讨论的示例系统(例如,LLAMA)可以提供用于管理页的存在、位置、以及持久存储的操作。为了对所存储的数据的量进行调整,访问方法可从其所管理的集合中加上或减去各页。为了提供状态持久存储,访问方法可以不时地将各页转储清除到辅助存储。为了管理这一持久存储性,各页可被适当地注释(例如,使用日志序列号(lsn))。例如,页管理器可被配置成控制各页上的转储清除操作、分配操作、以及释放操作。

[0067] 例如,转储清除操作可被指示为 Flush(PID, in-ptr, out-ptr, annotation) (转储清除 (PID, 输入指针, 输出指针, 注释))。例如,Flush(转储清除)可将页状态复制到日志结构化存储(LSS) I/O 缓冲区中。Flush 可在其对主存储器的影响方面有时类似于 Update-D,因为它将增量(具有注释)前缀到先前状态。这一增量可被标记为“转储清除”。根据本文讨论的示例技术,诸如 LLAMA 等示例系统可存储该页所处的 LSS 辅助存储地址(称为闪存偏移)和转储清除增量中的调用者“annotation(注释)”。例如,Flush 可能不向用户确保在它返回时 I/O 缓冲区是稳定的。

[0068] 例如,缓冲区管理器可被配置成经由无锁存更新操作来控制对日志结构化辅助存储缓冲区的更新。因而,例如,多个线程可经由无锁存操作来同时更新日志结构化辅助存储缓冲区。

[0069] 例如,“使得稳定”操作可被指示为 Mk-Stable(LSS address) (使得稳定 (LSS 地址))。例如,Mk_Stable 操作可确保转储清除到 LSS 缓冲区的页,上至 LSS 地址自变量,在

辅助存储上是稳定的。在 Mk_Stable 返回时,所提供的 LSS 地址以及所有更低 LSS 地址被确保在辅助存储上是稳定的。

[0070] 例如,“高稳定”操作可被指示为 Hi-Stable(out-LSS address)(高稳定(输出 LSS 地址))。例如,Hi_Stable 操作可返回当前在辅助存储上稳定的最高 LSS 地址。

[0071] 例如,页管理器可被配置成基于以下操作来发起高速缓存层存储中的第一页到辅助存储中的位置的转储清除操作:发起所述第一页的页状态到辅助存储缓冲区的复制,以及发起转储清除增量记录到所述页状态的前缀,所述转储清除增量记录包括指示所述第一页在辅助存储中的存储位置的辅助存储地址以及与调用者相关联的注释。

[0072] 例如,缓冲区管理器可被配置成发起稳定性操作以用于确定转储清除到辅助存储缓冲区的具有较低地址(上至第一辅助存储地址自变量)的各页在辅助存储中是稳定的。

[0073] 例如,“分配”操作可被指示为 Allocate(out-PID)(分配(输出 PID))。例如,Allocate(分配)操作可返回在映射表中分配的新页的 PID。所有这样的页可被永久记住,使得 Allocate 可被包括作为系统事务的一部分(如下文进一步讨论的),这可自动转储清除其包括的操作。

[0074] 例如,“释放”操作可被指示为 Free(PID)(释放(PID))。例如,Free(释放)操作可以作出由可供重用的 PID 所标识的映射表条目。在主存储器中,PID 可被置于当前时期的 PID 的待决释放列表上(如下文进一步讨论的)。同样,因为活动页可被记住,所以 Free 可被包括作为系统事务的一部分。

[0075] 根据本文讨论的示例技术,示例 LLAMA 系统事务可被用来提供对结构修改(例如,SMO)的相对耐久性和原子性(全部或没有)。例如,LSS 及其面向页的记录可被用作“日志记录”。例如,事务内的所有操作可被自动地转储清除到存储器内 LSS I/O 缓冲区,作为改变高速缓存中的页状态的补充。例如,每一 LSS 条目可包括严格作为“页”存储的示例 LSS 的页的状态。

[0076] 在主存储器中,事务内的所有这样的操作可被保持隔离,直至事务提交,如下文进一步讨论的。例如,在提交时,事务中的所有页变化可被原子地转储清除到 LSS 缓冲区。例如,在放弃时,所有变化可被丢弃。例如,系统事务管理器可被配置成提交事务和放弃事务。

[0077] 例如,系统事务可经由 LLAMA 支持的操作来被发起和终止。

[0078] 例如,“事务开始”操作可被指示为 TBegin(out-TID)(事务开始(输出 TID))。例如,由事务 ID(TID)标识的事务可被发起。这可涉及将它输入到由示例 LLAMA 高速缓存层(CL)管理器维护的活动事务表(ATT)。

[0079] 例如,“事务提交”操作可被指示为 TCommit(TID)(事务提交(TID))。例如,事务可从活动事务表移除且事务可被提交。例如,事务中的页状态变化可被安置在映射表中并转储清除到 LSS 缓冲区。

[0080] 例如,“事务放弃”操作可被指示为 TAbort(TID)(事务放弃(TID))。例如,事务可从活动事务表中移除,经改变的页可在高速缓存中被复位成“事务开始”,且没有变化被转储清除。

[0081] 根据本文讨论的示例技术,除了 Allocate 和 Free 之外,Update-D 操作可被准许在事务内改变页状态。例如,Update-R 可能不被使用,因为它可能使事务撤消复杂化,如下文进一步讨论的。

[0082] 根据本文讨论的示例技术,事务操作可全部具有输入参数:TID和注释(annotation)。例如,TID可被添加到高速缓存中的增量,且annotation(注释)可被添加到在事务中更新的每一页(例如,如同它被转储清除一样)。在被安置在转储清除缓冲区并提交时,高速缓存中所有经更新的页可被前缀了描述它们的位置的转储清除增量(例如,如同它们被独立于事务来转储清除一样)。

[0083] BWE树(参见例如J.Levandoski等人的“The Bw-Tree:A B-tree for New Hardware Platforms”,第29届IEEE数据工程国际会议(ICDE 2013),2013年4月8日-11日,)可提供可允许支持用户事务的示例键-值存储(例如,针对事务组件202)。例如,它可以管理LSN,实施预写日志(WAL)协议,并对设置检查点请求作出响应,如DEUTERONOMY数据组件(DC)所预期的(参见例如J.Levandoski等人的“Deuteronomy:Transaction Support for Cloud Data”,新数据系统研究会议(CIDR)(2011年1月),第123-133页以及D.Lomet等人的“Unbundling Transaction Services in the Cloud”,新数据系统研究会议(CIDR),2009年)。本文的讨论包括解决在使用诸如LLAMA等示例系统时它可如何实现。

[0084] Update-D和Update-R LLAMA操作的“数据”内容可包括键、LSN、以及键值存储的“数据部分”。例如,BW树因而可经由这些操作实现键值存储,经由LSN提供幂等,经由Update-D执行增量更新,经由Update-R执行其页合并,以及使用LLAMA Read或Flush操作来访问用于读或写的各页。例如,该系统可包括可被配置成基于更新增量记录操作和替换更新操作来控制更新的记录管理器。

[0085] 例如,访问方法可将LSN存储在它经由更新操作提供给LLAMA的数据中。此外,存储在转储清除增量中的Flush操作annotation参数可以提供附加信息来描述页内容。例如,这些可准许BW树实施预写记入日志(WAL)。例如,在转储清除页之后的稳定操作(例如,Mk-Stable)可使更新稳定以用于事务日志检查点。

[0086] 例如,Allocate和Free操作可准许示例BW树实现生长和缩小其树。例如,开始事务(BeginTrans)(例如,TBegin)和提交/放弃(例如,TCommit/TAbort)可启用在执行结构修改操作(SMO)时预期的原子性。

[0087] 例如,更新操作(例如,Update-D/Update-R)可不被限于“用户级”数据。例如,BW树可以在实现SMO时使用Update-D来张贴其“归并”和“拆分”增量,如下文参考系统事务进一步讨论的。

[0088] 根据本文讨论的示例技术,参考高速缓存层数据操作,页更新可通过使用比较并交换操作(CAS)将新页状态指针402安置在映射表304中来实现,而不论增量更新(如图4所示)还是替换更新(例如,如下文参考图7进一步讨论的)。例如,替换更新(例如,Update-R(PID, in-ptr, out-ptr, data))可包括所需新状态和页的先前状态在LSS中的位置两者。例如,新更新增量404(例如,Update-D(PID, in-ptr, out-ptr, data))指向页302的先前状态406,它已经包括这一LSS位置。

[0089] 例如,这样的无锁存方法可避免锁存引入的延迟,但它可导致其自己的惩罚,如“乐观”并发控制方法那样,即CAS可能失败且更新将随后被重试。例如,可能留待示例LLAMA用户来在适当时重试其操作,因为示例LLAMA实现可指示失败何时发生。

[0090] 根据本文讨论的示例技术,尽管在数据处于高速缓存(例如,312)中时没有操作

可阻塞,但从辅助存储读取页可涉及等待该页出现在高速缓存中。映射表(例如,映射表 304)将指向 LSS 页,甚至对于经高速缓存的页,如上所述,从而允许各页在高速缓存和 LSS 之间移动以用于高效的高速缓存管理。

[0091] 根据本文讨论的示例技术,在页被转储清除时,示例 LLAMA 实现可确保高速缓存(例如,312)中表示的事物与 LSS(例如,314)中的事物相匹配。因而,转储清除增量可将 PID 和 LSS 偏移两者包括在转储清除增量中,且可通过将它前缀到该页 302 而包括 LSS 缓冲区和高速缓存(例如,312)中的增量。

[0092] 根据本文讨论的示例技术,因为示例 LLAMA 实现可支持增量更新,所以页状态可包括不毗连片段。将这一特征与转储清除活动相组合可使得高速缓存内的页使其状态的一部分处于 LSS 中(早先已被转储清除),同时最近更新可只存在于高速缓存中。在这一点发生时,降低下一次转储清除的存储成本是可能的。

[0093] 因而,示例 LLAMA 实现可通过将写入只包括自先前转储清除以来的变化的增量来将这样的页转储清除。例如,通过写入毗连形式的增量(在本文中这可被称为“C-增量”),具有指向 LSS 中的该页的其余部分的指针,可使高速缓存中的多个更新增量全部毗连用于转储清除。因而,整个页在 LSS 中是可访问的,但在可能若干片段中。

[0094] 根据本文讨论的示例技术,Flush 操作可观察到可具有已随时间以此方式被转储清除的若干部分的经高速缓存的页状态,从而造成其中表示了分开的片段以及它们的 LSS 地址的经高速缓存的页。根据本文讨论的示例技术,在任何时刻,Flush 可通过毗连地写入不毗连页片段的内容,来将这些片段在 LSS 存储中集合在一起。例如,用户可能希望在 LSS 使用闪存存储时将各片段留待是分开的,而在 LSS 使用盘存储时需要毗连性(由于不同的读访问成本)。

[0095] 根据本文讨论的示例技术,在页被转储清除时,对于系统而言,在转储清除之前知晓该页的什么状态被转储清除可能是合乎需要的。例如,这可使用锁存来容易地查明,因为系统可简单地锁存该页并执行转储清除。然而,在无锁存方法中,该系统在页被转储清除之前防止对页的更新进行转储清除时可能具有很大困难。例如,这在实施预写记录协议时或在转储清除作为结构修改的一部分发生时可能引起问题。例如,在它们执行它们的 CAS 时,使不适当的转储清除失败可能是合乎需要的。因而,根据本文讨论的示例技术,指向要在 CAS 中转储清除的页状态的指针可被使用,这随后可仅捕捉特定状态并且在该状态在转储清除完成之前已被更新的情况下可能失败。然而,这可引起其他问题。

[0096] 在研究本文讨论的示例技术时,在确定在执行高速缓存管理并将各页转储清除到 LSS 时可能有利的这种强不变量时遇到困难。例如,不变量可包括诸如以下属性:

[0097] 成功转储清除到 LSS 的页在已被转储清除时在高速缓存中被立即看到,且该页的经转储清除的状态将在所有稍后状态的转储清除之前处于 LSSI/O 缓冲区中。其转储清除失败的页将不在高速缓存中表现为已被转储清除,且在查看 LSS 时将明白转储清除未成功。

[0098] 例如,两个替换方法可包括:

[0099] 1) 转储清除的成功可通过首先执行 CAS 来确保。一旦 CAS 成功,该页就可被张贴到 LSS。例如,如果这一点完成,则竞争条件可削弱值得信赖的 LSS 恢复。例如,依赖于早先转储清除的页可能随后被转储清除,其中这一“稍后”转储清除在系统崩溃之前成功写入

LSS,同时“早先”转储清除过慢而没有完成且没有出现在稳定 LSS 中。这一情况可能损害某种形式的因果关系。

[0100] 2) 需要被转储清除的页的页状态可被捕捉,且被写入 LSS 缓冲区。随后,CAS 可被尝试,且 CAS 可能失败。因而,页被写入 LSS,而没有在系统崩溃的情况下用于区分转储清除是成功还是失败的指示。例如,可存在在各个时刻被写入 LSS 的多个这样的页。例如,该页的稍后状态可被写入,它看起来在 LSS 中早于失败的 CAS。如上所示,它较晚开始但在较早转储清除之前获得其缓冲区槽。

[0101] 根据本文讨论的示例技术,上述两难可被解决,如下文讨论的。例如,如果 CAS 被执行得足够早,则可以确定在将页的状态复制到日志缓冲区之前转储清除是否将成功。因而,示例转储清除过程可被如下执行:

[0102] 步骤 1:标识旨在被转储清除的页的状态。

[0103] 步骤 2:占用将该状态写入到的 LSS 缓冲区中的空间。

[0104] 步骤 3:执行 CAS 来确定转储清除是否将成功。转储清除增量中的 LSS 偏移将被获得以这样做(如在以上步骤 2 提供的)。

[0105] 步骤 4:如果步骤 3 成功,则写入状态以供保存到 LSS 中。在这被写入 LSS 中时,本文讨论的示例 LLAMA 技术可防止缓冲区被写入 LSS 辅助存储。

[0106] 步骤 5:如果步骤 3 失败,则将指示“失败的转储清除”的指示写入缓冲区中的保留空间。这可消耗存储,但解决了与转储清除成功还是失败有关的模糊性。

[0107] 这一示例过程的结果在于 LSS 在恢复期间可能没有观察到作为已失败的 CAS 的结果的页。例如,这还保留了以下属性:稍后(在它在“日志”中的位置方面)出现在 LSS 中的任何页将是比 LSS 日志中该页的所有较早实例更晚的页状态。

[0108] 根据本文讨论的示例技术,使示例 LLAMA 实现管理高速缓存并换出数据以满足其存储器约束可能是合乎需要的。例如,示例 LLAMA 实现可能知晓增量更新、替换更新、以及转储清除,且可识别这些中的每一者。然而,示例 LLAMA 实现将不知晓页的内容,如果它将是通用的话。因而,示例 LLAMA 实现不知晓访问方法层是否通过在页中维护 LSN 而支持事务。因而,可能发生的问题包括与示例 LLAMA 实现在其不能看到 LSN 并实施预写日志协议时可如何提供高速缓存空间管理(包括逐出各页)有关的潜在问题。

[0109] 根据本文讨论的示例技术,已被转储清除的任何数据可从高速缓存中丢弃。例如,就地更新各页的系统可被防止换出(从高速缓存丢弃)任何最近更新的且“脏”的页。然而,因为增量更新,示例 LLAMA 实现可确定页的哪些部分已被转储清除。例如,每一这样的部分可以用转储清除增量来描述,且这些经转储清除的部分可被“换出”高速缓存。

[0110] 在“换出”页部分时,简单地解除分配存储并重用它可能是不合乎需要的,因为可能留下对经换出的部分的虚悬(dangling)引用。因而,根据本文讨论的示例技术,描述页的什么部分已被换出的增量可被使用。

[0111] 例如,对于被完全换出的页,其在映射表 304 中的主存储器地址可被来自该页的最近转储清除增量的 LSS 指针替换。

[0112] 图 5 描绘示例部分页换出以及示例部分交换增量。例如,对于部分换出的页,CAS 可被用来插入“部分交换”增量记录 502。例如,这一增量记录 502 可指示该页已被部分换出(例如,使得该页不能被正常访问),且可指向指示 LSS 中的用于定位页 506 的丢失部分

的位置信息的转储清除增量记录 504。例如,一旦“部分交换”增量 502 被用 CAS 进行了安置,针对该页的被丢弃的该部分的存储器可使用示例时期机制被释放,如下文进一步讨论的。

[0113] 例如,页管理器可被配置成基于发起部分交换增量记录到与第一页相关联的页状态的前缀,来发起高速缓存层存储中的所述第一页的一部分到辅助存储中的位置的交换操作,所述部分交换增量记录包括指示转储清除增量记录的存储位置的辅助存储地址,所述转储清除增量记录指示所述第一页的丢失部分在辅助存储中的位置。

[0114] 例如,页管理器可被进一步配置成使用时期机制来发起与第一页的所述部分相关联的高速缓存层存储的释放操作。

[0115] 根据本文讨论的示例技术,这一方法可有利地向用户提供若干有用特征。例如,这样的示例 LLAMA 实现的高速缓存层(例如,312)可收回存储器,而无需与页的实际内容有关的知识。例如,丢弃经转储清除的页和页的经转储清除的部分可不涉及 I/O 操作。例如,与多个部分在 LSS 中的完全经转储清除的页的情况相比,将部分地经转储清除的页带回主存储器可涉及较少的 LSS 读。

[0116] 例如,若干示例高速缓存管理策略可被用来管理高速缓存存储(例如,最近最少使用(LRU)、LRU(k)、时钟,等等——参见例如 W.Effelsberg 等人的“Principles of database buffer management(数据库缓冲区管理原理)”,ACM 数据库系统会议录(TODS),卷 9,第 4 期(1984 年 12 月),第 560-595 页以及 E.O'Neil 等人的“The LRU-K page replacement algorithm for database disk buffering(数据库盘缓冲的 LRU-K 页替换算法)”,1993ACM SIGMOD 数据管理国际会议管理会议录(SIGMOD'93),第 297-306 页。这些示例可涉及附加记账,但可能没有引起显著困难。

[0117] 根据本文讨论的示例技术,使用这样的示例无锁存方法,操作可检查页和页状态两者,甚至在它们已被指定为“垃圾”之后。例如,在没有使用常规“锁存”时,该系统可能未能阻止以下任一情况:1)Update-R 操作替换整个页状态,从而在另一操作正在读取它时解除分配先前状态;或 2)在另一操作正在检查映射表中的页时“释放”它的 De-allocate 操作。

[0118] 根据本文讨论的示例技术,存储或 PID 都不可被允许重用,直至不存在另一操作正在访问它们的可能性。因而,可确立“已释放资源”和“可重用资源”之间的区别。例如,“已释放资源”已被一操作指定为垃圾。例如,“可重用资源”已被释放且可被确保不能由任何其他操作访问。例如,时期可被用来保护已解除分配的对象被过早重用(参见例如 H.Kung 等人的“Concurrent manipulation of binary search trees(二分搜索树的并发操纵)”,ACM 数据库系统会议录(TODS),卷 5,第 3 期(1980 年 9 月),第 354-382 页。

[0119] 根据本文讨论的示例技术,每一操作可在访问 PID 或页状态之前加入当前时期 E,并且一旦这样的访问完成就退出 E。例如,操作可总是将已释放资源张贴在当前时期的列表上,这可能是 E(它加入的时期)或稍后时期(如果当前时期已前进的话)。例如,E 的列表上的资源都不可被重用,直至加入 E 的所有操作已退出。

[0120] 例如,时期可被编号,且不时地,新时期 E+1 可变成“当前”时期。新操作因而可继续加入当前时期,现在是 E+1。例如,时期机制不变量是:时期 E+1 或稍后时期中的操作都不可看到且使用时期 E 中释放的资源。

[0121] 因而,基于这一不变量,一旦所有操作已退出 E,没有活动操作能访问 E 中释放的资源。图 6 示出两个示例时期 602、604 以及它们相应的垃圾收集列表 606、608。如图 6 所示,垃圾收集项 610 与时期 1 (602) 中的“线程 1”相关联,垃圾收集项 612 与时期 1 (602) 中的“线程 2”相关联,且垃圾收集项 614 与时期 2 (604) 中的“线程 3”相关联。如图 6 所示,时期 2 (604) 的垃圾收集列表 608 中的垃圾收集项 616 与时期 1 (602) 的“线程 1”相关联。

[0122] 例如,一旦“线程 1”和“线程 2”已退出时期 1 (602),没有活动操作可访问时期 1 (602) 中释放的资源(例如,垃圾收集项 610 和垃圾收集项 (612))。

[0123] 例如,第一时期管理器可被配置成在第一处理器操作访问页信息之前,发起第一处理器操作加入第一时期登记列表。

[0124] 第一时期管理器可被配置成将第一处理器操作释放的一个或多个资源张贴在第一时期垃圾收集列表中。第一时期管理器可阻塞张贴在第一时期垃圾收集列表中的经张贴的资源的重用,直至第一时期登记列表不包括当前加入的处理器操作。

[0125] 根据本文讨论的示例技术,示例 LLAMA 实现可按类似于日志结构化文件系统 (LFS) 的日志结构化的方式(参见例如 M. Rosenblum 等人的“The Design and Implementation of a Log-Structured File System(日志结构化文件系统的设计和实现)”,ACM 计算机系统会议录 (TOCS),卷 10,第 1 期,1992 年 2 月,第 26-52 页)来组织辅助存储(例如,闪存存储)上的数据。因而,每一页转储清除重新分配该页在闪存上的位置。例如,这可提供使用本文讨论的示例映射表 304 的附加原因。例如,日志结构化存储可有利地减少每页写的数量,并使得写“顺序化”。因而,许多随机写可被转换成一个大型多页写。

[0126] 如上所述,“逻辑页”可包括基础页和指示对该页的更新的零个或多个增量记录,从而允许一页在被转储清除时以各片段的形式来被写到闪存。因而,闪存上的逻辑页可与可能在使用文件偏移作为指针来链接在一起的不同物理设备块上的记录相对应。此外,物理块可包括来自多个逻辑页的记录。图 7a 示出了闪存 314 上的示例日志结构化存储组织 700a。

[0127] 例如,逻辑页可通过从闪存上的链的头开始(其在顺序日志 702 中的偏移可以从映射表 304 获得)并遵循链接的记录来从闪存 314 读到存储器(例如,RAM 312)。例如,偏移 704 可以从映射表 304 获得以用于访问增量记录 706,以获得当前状态以及基础页 708,以用于将对应的“逻辑页”从闪存 314 读到存储器 312。

[0128] 例如,偏移 710 可以从映射表 304 获得以用于访问增量记录 712,以获得增量和链接来访问第二增量记录 714 以及随后访问基础页 716,以用于将对应的“逻辑页”从闪存 314 读到存储器 312。

[0129] 例如,转储清除过程可在将同一逻辑页的多个增量记录一起转储清除时将它们有利地合并成闪存上的毗连 C- 增量。此外,当它在被合并并在存储器中之后被转储清除时,逻辑页可被合并并在闪存上,这可有利地改进页读取性能。

[0130] 图 7b 描绘示例映射表 304,指示基于用页 742 的新状态 744 的物理地址(例如,得自页 742 与先前前缀的增量记录的合并)替换第一存储对象 746 的物理地址(例如,它包括具有多个先前前缀的增量记录的基础页 742,在图 7b 中),用页 742 的新状态 744 替换页 742 的先前状态 740。

[0131] 例如,如图 7c 所示,用页 742 的新状态 744 替换页 742 的先前状态 740 可包括将

多个增量记录合并到毗连 C- 增量 750, 这随后可与基础页 742 一起被转储清除。

[0132] 例如, 用页 742 的新状态 744 替换页 742 的先前状态 740 可包括经由映射表 304 上的原子比较并交换操作, 生成当前页 742 的经修改版本或确定用于替换当前页 742 的另一页, 并用页 742 的新状态 744 的物理地址来替换当前页 742 的物理地址 (例如, 用于替换的经修改版本或另一页)。

[0133] 例如, 作为图 7b 和图 7c 的特征之间的区别, 在将页写入辅助存储时, LLAMA 可执行图 7c 中所示的合并, 但它依赖于执行 Update-R 的访问方法来执行图 7b 的合并。

[0134] 根据本文讨论的示例技术, 示例 LLAMA 实现可整个无锁存。此外, 专用线程可不被用来转储清除 I/O 缓冲区, 因为这可使保持线程工作负荷平衡复杂化。因而, 所有线程可参与管理这一缓冲区。例如, 常规方法利用了锁存。然而, 这样的常规技术可能只在分配缓冲区中的空间时锁存, 从而在数据传输之前释放锁存, 这随后可并行进行。

[0135] 根据本文讨论的示例技术, 示例 LLAMA 实现可避免用于缓冲区空间分配的常规锁存, 代替使用 CAS 来用于原子性, 如在本文讨论的示例系统中的别处完成的。例如, 这涉及定义 CAS 在其上执行的状态。例如, 缓冲区状态的恒定部分可包括其地址 (基础 (Base)) 和大小 (Bsize)。例如, 缓冲区中所使用的存储的当前高水位标记可以用相对于基础的偏移来跟踪。例如, 针对缓冲区的使用的每一请求可以用为页转储清除保留空间 Size (大小) 的努力来开始。

[0136] 根据本文讨论的示例技术, 为保留缓冲区中的空间, 线程可获得当前偏移并计算偏移 + 大小 (Size)。例如, 如果偏移 + 大小 $\leq$ Bsize, 则该请求可被存储在缓冲区中。例如, 线程可使用当前偏移作为比较值且偏移 + 大小作为新值来发出 CAS。如果 CAS 成功, 则偏移可被设为新值, 且空间可被保留, 并且缓冲区写入者可将数据传送到缓冲区。

[0137] 根据本文讨论的示例技术, 这一逻辑可处理缓冲区中的空间分配。例如, 写入缓冲区以及管理多个缓冲区可涉及 CAS 状态中的更多, 这在下文进一步讨论。

[0138] 在将缓冲区写入辅助存储时, 如果偏移 + 大小 $>$ Bsize, 则缓冲区中没有足够空间来保持该线程的记录。在这种情况下, 线程可密封缓冲区, 从而将它标记为不再被使用, 且为被准备写入到辅助存储。这一条件可以用转储清除缓冲区状态中的“密封 (Sealed)”位来跟踪。例如, CAS 可将“密封”位从 F (例如, 假) 改变成 T (例如, 真)。例如, 密封的缓冲区可不再被更新, 且遇到密封的缓冲区的线程将寻求不同的 (未密封的) 缓冲区。

[0139] 根据本文讨论的示例技术, 密封的缓冲区可不再接受新更新请求。然而, 该示例系统可能尚未确信先前写 (它们全部成功获得了缓冲区空间) 已完成传输它们的数据到缓冲区。根据本文讨论的示例技术, “活动 (Active)”计数可指示将数据传送到缓冲区的写入者的数量。例如, 在保留缓冲区中的空间时, 写入者的 CAS 可包括表示偏移、密封、以及活动的值。例如, 写入者的 CAS 可获取这一结构, 将它的净荷大小与偏移相加, 将“活动”递增 1, 并且如果 ~ 密封, 则可执行 CAS 来更新这一状态并保留空间。例如, 在写入者完成时, 它可重新获取这一状态, 将“活动”递减 1, 并可执行 CAS 来实现该改变。例如, 操作可在失败的情况下按需重做。

[0140] 例如, 如果缓冲区是密封的且活动 = 0, 则该缓冲区可能是可转储清除的。例如, 造成这一条件的写入者可负责发起 I/O。例如, 在 I/O 完成时, 缓冲区的偏移和活动用户两者可被设为零, 且缓冲区可以是未密封的。

[0141] 根据本文讨论的示例技术,对于多个缓冲区,多个缓冲区的集合中的各缓冲区中的每一者具有以上指示的状态。图 8 描绘示例完成转储清除缓冲区状态 800。如图 8 的示例所示,每缓冲区状态 802 可包括 32 位,包括用于下一个写的偏移的 24 位 804、用于活动写入者的数量的 7 位 806、以及用于“密封位”指示符的 1 位 808(例如,指示密封的缓冲区)。例如,当前活动的缓冲区号 (CURRENT(当前))810 可指示当前活动的缓冲区(例如,8 位,如图所示)。

[0142] 例如,缓冲区可循环式地被访问和使用,使得一个缓冲区被密封(如密封位指示符 808 所指示的),本文的示例技术可进至缓冲区“环”中的下一缓冲区(例如,使用 CURRENT 810)。根据本文讨论的示例技术, CURRENT 810 可被用来指示缓冲区集合中的哪一缓冲区当前正接受新写入请求。

[0143] 根据本文讨论的示例技术,密封当前活动缓冲区的线程(例如,经由“密封位”指示符 808)在它密封缓冲区时也将更新 CURRENT 810。例如,这一线程随后可选择下一 CURRENT 缓冲区。例如,在缓冲期 I/O 完成时, I/O 线程可将缓冲区解除密封,但可不设置 CURRENT 810,因为可能存在充当当前缓冲区的另一缓冲区。

[0144] LSS 是日志结构化存储,且因此在概念上“仅限追加”。例如, LSS 的实现可涉及持续收回空间来用于追加页的新版本,与任何典型的日志结构化文件系统 (LFS) 一样。例如,这一技术在本文中可被称为“清除”(参见例如以上 M. Rosenblum 等人)。

[0145] 因为示例页的不同版本可具有不同寿命,所以示例“日志”的旧部分(重用可能是合乎需要的)将包括当前页状态是可能的。例如,为重用示例日志的这一“旧”部分,仍然当前的页状态可被移至日志的活动尾部,将它们追加在此,使得较旧部分可被循环用于后续使用。例如,清除的这一副作用可增加写的数量(这在本文中可被称为“写放大”——参见例如, X.-Y. Hu 等人的“Write amplification analysis in flash-based solid state drives(基于闪存的固态驱动器中的写放大分析)”, SYSTOR 会议录 2009: 以色列实验系统会议 (SYSTOR' 09), 第 10 篇论文)。

[0146] 例如,清除努力可被简单地组织。例如,日志可作为大型“圆缓冲区”被管理,其中最旧部分(例如,日志的头)可被“清除”且添加作为日志的活动尾部处的新空间,在此写入新页状态。

[0147] 根据本文讨论的示例技术,在重新分配的每一页被重写时,使它是毗连的(例如,在页被重新追加到 LSS 存储时,“重写”事项是毗连的)。因而,如它可能具有的许多增量转储清除,页的所有部分现在是毗连的,从而有利地优化页在 LSS 中的可访问性。

[0148] 根据本文讨论的示例技术,增量(它在本文中可被称为“重新分配增量”)上的 CAS 可在该页的映射表条目处执行,从而提供新位置并描述页的哪些部分已被重新分配(即,管理高速缓存以安置新位置信息)。例如,并发更新或转储清除可使得这一 CAS 失败,在这种情况下, CAS 被再次尝试。

[0149] 存储效率可具有对日志结构化存储系统的有利正面影响。根据本文讨论的示例技术,对于分配给 LSS 的任何给定量的空间,它使用该空间的效率越高,它可执行的清除就越少,这可涉及更少的页移动例如,页移动可造成对存储的附加写(例如,写放大)。

[0150] 关于潜在 LSS 存储效率,被转储清除的页中没有空空间。例如,它们可被写成打包的可变长度串(例如,平均地,常规 B 树页可能只被利用 69%)。此外,因为只有自先前转

储清除以来的增量可能频繁被转储清除,所以每页转储清除可消耗较少空间。另外,将经更新的页换出高速缓存将不涉及附加转储清除,因为高速缓存中的主存储器可只针对先前转储清除的页部分被收回。

[0151] 访问方法的一个示例方面是它们作出结构修改操作(SMO)以准许这样的结构生长和缩小。例如,SMO 预期将存在影响索引的原子变化的方式,以便普通更新可在存在正在进行的 SMO 的情况下正确地执行且是原子的(全部或没有)。例如,示例 BW 树可利用系统事务作为其 SMO 的机制。

[0152] 根据本文讨论的示例技术,系统事务的耐久性可经由日志来实现。然而,本文讨论的一些示例日志不是事务日志,而是示例 LSS “页”存储,在给定事务系统通常可能只将操作记入日志的情况下,这可能看起来有点低效。然而,在增量更新的情况下,页状态可通过只将自先前页转储清除以来的增量更新记入日志来被记入日志。在提交处的耐久性未被涉及,使得提交不“迫使”LSS 缓冲区。然而,根据本文讨论的示例技术,使用事务的结果的所有后续操作可被确保发生在 LSS 中的事务提交之后。

[0153] 根据本文讨论的示例技术,类似于非事务操作,所有事务操作可经由映射表中的页指针上的 CAS 来安置。本文讨论的示例技术可确保高速缓存中的内容被可信地表示在 LSS 中,且反之亦然。因而,系统事务内的基本上所有更新可包括转储清除操作。例如,每一系统事务更新可被记录在 LSS 缓冲区中,并且因而可被“记入日志”。例如,信息的两个表示可以是等同的,因而确保在系统崩溃的情况下,高速缓存的状态能可信地根据由 LSS 稳定捕捉的最后高速缓存来重建。

[0154] 在各动作涉及一个以上页时,如对于 SMO,这一等同可能在常规上是有问题的。例如,B-LINK(B 链接)树中的节点拆分 SMO 既分配新页又更新其兄弟页链接指针以引用新页。例如,基于锁存的系统中的 SMO 通常可使用锁存来提供隔离,使得多页 SMO 的内部状态在高速缓存管理器中不可见,直至 SMO 完成。例如,无锁存设计可意味着隔离活动(并且因此未提交)事务更新的能力可被限制。

[0155] 根据本文讨论的示例技术,示例 LLAMA 实现可以提供准许对页的基本上任意访问的事务接口(即,任意页上的操作可被置于事务内)。然而,在事务期间更新的页可能未被保护免于由该事务外部的操作访问。然而,根据本文讨论的示例技术,可以设计不涉及完全通用隔离能力的 SMO。例如,图 9 示出了可被用于捕捉 SMO 事务的示例事务模板 900。

[0156] 例如,在步骤 1(902),在映射表中各页被分配或释放。在步骤 2(904),各页被按需更新。在步骤 3(906),现有页被更新以将新页连接到索引的其余部分或在更新另一页时移除现有页。

[0157] 根据本文讨论的示例技术,节点拆分的新节点(使用图 9 的示例模板)在它被连接到树且事务被提交时对其他线程而言是不可见的,直至图 9 的步骤 3。因而,这样的 SMO 事务可以提供原子性和隔离两者。

[0158] 有点类似于常规事务系统,活动事务表可被维护以用于系统事务,这在本文中可被称为活动事务表(ATT)。例如,ATT 可包括每活动系统事务的条目,该条目包含事务的事务 id(TID)和指向事务的紧接先前操作的指针(这在本文中可被称为“IP”(意指“紧接先前”)),它指向(或以其他方式引用)该事务的最近操作的存储器地址。

[0159] 例如,开始事务操作(例如, TBegin)可将新条目添加到 ATT,具有高于任何在前

事务的事务 id(TID), 其中 IP 被设为空值。例如, 事务操作的执行可创建该操作的“日志记录”, 指回由 IP 标识的操作的日志记录, 且 IP 可被更新以引用新操作。例如, 这可用于链接回事务的操作的“日志记录”, 其中所有“日志记录”在主存储器中。此外, 根据本文讨论的示例技术, 系统事务内的操作可只经由映射表更新改变高速缓存状态 (即, 不改变 LSS 缓冲区状态)。根据本文讨论的示例技术, 这些页可在事务提交时被转储清除。根据本文讨论的示例技术, 在事务的结束 (提交或放弃) 发生时, 该事务可以从 ATT 移除。

[0160] 例如, 系统事务管理器可被配置成将第一事务的事务标识符 (TID) 添加到由高速缓存层管理器维护的活动事务表 (ATT)。例如, 事务提交管理器可被配置成基于将 TID 从 ATT 移除、将与第一事务相关联的页状态变化安置在映射表中、以及发起与第一事务相关联的页状态变化到辅助存储缓冲区的转储清除, 来提交第一事务。

[0161] 根据本文讨论的示例技术, 在提交操作时, 被事务改变的页将以原子的方式被转储清除到 LSS 缓冲区。作为示例技术, 这些页写可以用事务的开始和结束记录囊括在 LSS 中; 然而, 这可涉及崩溃恢复以撤消中断的事务。例如, 这样的撤消恢复可涉及将撤消信息写入 LSS。根据本文讨论的示例技术, 这可通过在提交被事务改变的所有页时执行原子转储清除来被避免, 如下文进一步讨论的。

[0162] 根据本文讨论的示例技术, 依赖于 SMO 的后续动作将在 LSS 缓冲区中看起来晚于描述 SMO 事务的信息。因而, 在 SMO 的状态在高速缓存中变得对操作系统事务的线程以外的线程可见时, 这些其他线程可依赖于 SMO 已被提交给 LSS 并已经存在于 LSS 缓冲区中。

[0163] 如图 9 所示, 步骤 3 指示“更新现有页, 以在更新另一页时将新页连接到索引的其余部分或移除现有页”。因而, 本文讨论的示例技术可经由原子转储清除, 将主存储器中的更新 (使事务状态可见) 和 LSS 缓冲区中的事务提交两者封装, 从而使用 Update-D 的示例“提交”能力来实现这一点 (即, 将更新和事务提交组合在一起)。

[0164] 根据本文讨论的示例技术, LSS 可通过将更新和其 CAS 安置与事务中改变的所有页的原子转储清除相组合来启用事务 Update-D “提交”操作。例如, 在多个页的提交时的这一转储清除可针对各单独的页转储清除来类似地执行。例如, LSS 缓冲区空间可被分配给在事务中改变的所有页。随后, 安置前缀有转储清除增量的 Update-D 增量的 CAS 可被执行。如果 CAS 成功, 则在事务中更新的页可被写入 LSS 转储清除缓冲区。在事务的所有页的转储清除完成之后, 转储清除过程可递减转储清除缓冲区中的写入者的数量。例如, 作为单个单位来为事务中所有页分配空间 (这成立直至 LSS 缓冲区上的写入者递减) 可确保 LSS 存储中的事务的原子性。

[0165] 例如, 事务提交管理器可被配置成经由比较并交换 (CAS) 操作安置与映射表中的事务相关联的更新增量记录, 该更新增量记录被前缀有转储清除增量记录。例如, 事务提交管理器可被配置成确定 CAS 操作是否成功。如果事务提交管理器确定 CAS 操作成功, 则事务提交管理器可以发起写操作来将事务中更新的页写入辅助存储转储清除缓冲区。

[0166] 根据本文讨论的示例技术, 如果 CAS 失败, 则与其他转储清除失败类似地进行响应。例如, 被分配以使得 LSS 在恢复期间不将该空间与任何其他空间混淆的空间可以是空的 (VOIDed)。因而, 示例恢复过程可完全不知晓系统事务。相反, 系统事务可以仅是示例高速缓存层的能力。因而, 在无需确保跨系统崩溃或重启的 TID 唯一性的情况下进行是可接受的。

[0167] 根据本文讨论的示例技术,已放弃的系统事务的操作可在高速缓存中被撤消,因为恢复没有看到未完成的事务。因而,事务的日志记录的后链(在主存储器中被链接在一起)可被遵循,且撤消可基于事务的 ATT 列表上的操作的性质来提供。例如,增量更新可通过移除该增量来被撤消,分配可以用“释放”来被撤消,且“释放”可以通过将页恢复到其在“释放”之前的状态来被撤消。除撤消“释放”之外,不需要针对这些操作的额外信息,除了描述操作成功的信息之外。

[0168] 根据本文讨论的示例技术,发生在事务内的动作是临时的,包括存储和映射表条目(PID)的分配和释放。例如,在事务执行期间,PID 可被分配或释放,且 Update-D 增量可被生成。例如,这些资源的管理可基于时期机制来被实现,如本文中讨论的。例如,因为 SMO 在单个用户操作请求内执行,所以线程可保持在其时期中达该事务的历时。

[0169] 根据本文讨论的示例技术,示例 LLAMA 实现可依赖于事务提交或放弃来收回资源。例如,对于提交操作,FreePage PID(释放页 PID)可被添加到当前时期的 PID 待释放列表。例如,对于放弃操作,AllocatePage PID(分配页 PID)可在撤消期间被释放并类似地添加到 PID 待释放列表。例如,对于 Update-D 操作,在事务放弃的情况下,更新增量可被添加到当前时期的存储待释放列表。

[0170] 如本文讨论的,“崩溃恢复”一般不指“事务恢复”。如本文讨论的,“检查点”一般不指用来管理事务日志的检查点。相反,如本文讨论的,“崩溃恢复”可以指用于 LSS(例如,日志结构化存储)以将其各页以及它们的状态的映射表恢复到系统崩溃前的时刻的示例技术。这一特定类型的恢复步骤通常不关心常规就地更新存储系统。

[0171] 关于本文讨论的“崩溃恢复”,映射表可被认为是一种类型的“数据库”。例如,对这一数据库的更新可包括转储清除到 LSS 的页状态。因而,每一页转储清除可更新“映射表数据库”。在系统崩溃的情况下,LSS“日志”可被重放以使用转储清除的页作为重做日志记录以更新映射表,来恢复“映射表数据库”。

[0172] 在支持以上策略的情况下,映射表可被周期性地设置检查点,以避免无限期地维护 LSS 更新。例如,上述 LFS 清除技术可被用于这一目的(即,缩短恢复日志);然而,这样的技术可留下比高速恢复所需的日志基本上更大的恢复日志(LSS 日志结构化存储)。

[0173] 根据本文讨论的示例技术,有利策略可被用于设置检查点。例如,示例 LLAMA 实现可以是异步的且在对两个替换位置之一设置检查点期间递增地写入完整映射表。图 10 是根据本文讨论的示例技术的示例检查点数据 1000。例如,这两个替换位置可被选作两个不同的“公知位置”(WKL),使得该系统将知道这些位置,甚至在可能丢失与各实体的位置有关的其他“当前”信息的系统崩溃后。因而,指向与系统的状态(在崩溃时存在的状态)有关的信息的指针可被保存(例如,使用 WKL)。例如,通过使用两个检查点,用户可不就地更新“实况”检查点。

[0174] 例如,作为完整映射表的补充,每一位置可在闪存日志 1006 中存储恢复起始位置(RSP)1002 和垃圾收集偏移 GC 1004,如图 10 所示。例如,RSP 1002 可包括在发起映射表 304 的复制时 LSS 存储中的结束偏移。例如,GC 偏移 1004 可标记垃圾收集“边境”。

[0175] 根据本文讨论的示例技术,较晚检查点具有较高 RSP 1002,因为 LSS 偏移通过被虚拟化而单调递增。例如,在系统崩溃后,具有最高 RSP 1002 的已完成的检查点可被用来初始化经恢复的映射表 304 的状态。例如,RSP 1002 指示 LSS“日志”(1006)中的位置以

用于开始重做恢复。为标识最后完成检查点，RSP 1002 不被写入检查点，直至映射表 304 已被完全捕捉。因而，先前高 RSP（来自替换位置）将是最高 RSP 1002，直至当前检查点完成。

[0176] 根据本文讨论的示例技术，作为检查点的一部分写出映射表 304 不是映射表 304 的逐字节复制，因为它存在于高速缓存中。例如，映射表 304 的经高速缓存的形式具有针对经高速缓存的页的映射表条目中的主存储器指针，而本文讨论的示例所需检查点涉及捕捉各页的 LSS 地址。作为另一示例，当前未被分配的映射表条目被维护在使用映射表条目作为列表项的空闲列表上。因而，空闲映射表条目要么具有零要么具有紧接先前空闲映射表条目的地址（基于它们被添加到空闲列表的时间而处于时间次序）。例如，可使用的空闲列表可能在映射表的异步“复制”期间不被捕捉，如本文讨论的。例如，映射表 304 的复制（如本文讨论的）被异步且递增地写入，这可帮助最小化对正常执行的影响。

[0177] 根据本文讨论的示例技术，示例 LLAMA 实现可首先将 LSS 存储的当前结束偏移保存为 RSP 1002，并可当前 LSS 清除偏移保存为 GC 1004。例如，映射表 304 可被扫描（例如，与正在进行的操作并发地），且针对每一 PID 条目的页的最近转储清除的 LSS 地址（存储在最近转储清除增量中）可被标识，且该 LSS 地址可被存储在该映射表 304 条目的示例检查点中。例如，如果条目是空闲的，则该条目在检查点复制中可被置零。例如，空闲列表可在重做恢复的结束处被重建。此外，在映射表 304 的复制完成时，先前保存的 RSP 1002 和 GC 1004 可被写入稳定的检查点区，从而完成检查点。

[0178] 根据本文讨论的示例技术，恢复可通过将具有最高 RSP 1002 的检查点（即，最近完成的检查点）的映射表 304 复制到高速缓存 312 来发起。例如，日志 1006 随后可从 RSP 1002 前进至 LSS 的结束来被读出。例如，遇到的每一页转储清除可被带入高速缓存 312，如同它是页读取的结果一样。

[0179] 例如，页的内容可被读取，且增量可被设置使得 LSS 中的位置在转储清除增量中被引用。例如，在遇到分配页（AllocatePage）操作时，所分配的 PID 的映射表 304 条目可被初始化为“空”，如 AllocatePage（分配页）操作所预期的。例如，在遇到释放页（FreePage）操作时，映射表 304 条目可被设为零。例如，LSS 清除者可恢复对来自从检查点读取的 GC 偏移（1004）的日志进行垃圾收集。

[0180] 根据本文讨论的示例技术，在恢复期间，所有空闲映射表 304 条目可被设为零。例如，重建的映射表 304 可被扫描。例如，在遇到零条目，它可被添加到空闲列表，这可作为栈来管理（即，要被重用的第一条目是被添加到列表的最后一个条目）。根据这些示例技术，低阶 PID 可被重用（作为重用中的优选），这可趋向于保持表大小集群化且很小（至少作为恢复的结果）。例如，高水位标记可被维护在映射表中，从而指示迄今使用的最高 PID。例如，在空闲列表被耗尽时，PID 可从表的未使用部分添加，从而递增高水位标记。

[0181] 如在本文进一步讨论的，图 11 是用于管理无锁存和日志结构化存储的系统 1100 的框图。数据处理领域的技术人员将认识到系统 1100 可以用硬件实现、软件实现或其组合来实现。如图 11 所示，系统 1100 可包括包含至少一个处理器 1104 的设备 1102。设备 1102 可包括数据管理器 1106，数据管理器 1106 可包括可被配置成向任意选择的面向页的访问方法 1110 提供对页数据存储 1112 的接口访问的数据不透明接口 1108，所述对页数据存储 1112 的接口访问包括对所述页数据存储 1112 的无锁存访问。例如，面向页的访问方

法 1110 可以是任何任意访问方法。例如,页数据存储 1112 可包括任何类型的页数据存储,包括(至少)易失性存储(如主存储器)和更稳定的存储(例如,更非易失性的存储)(诸如“辅助存储”,它可包括闪存存储以及其他类型的盘驱动器,等等)。数据处理领域技术人员将明白,可存在可与本文讨论的技术一起使用的许多类型的页数据存储,而不背离本文讨论的精神。

[0182] 根据一示例实施例,数据管理器 1106 或其一个或多个部分可包括可被存储在有形计算机可读存储介质上的可执行指令,如下所讨论。根据一示例实施例,计算机可读存储介质可包括任何数量的存储设备以及任何数量的存储介质类型,包括分布式设备。

[0183] 在该上下文中,“处理器”可包括被配置成处理与计算系统关联的处理指令的单个处理器或多个处理器。处理器由此可包括以并行方式和/或分布式方式处理指令的一个或多个处理器。虽然设备处理器 1104 在图 11 中被描绘为在数据管理器 1106 外部,但数据处理领域的技术人员将理解,设备处理器 1104 可被实现为单个组件和/或可位于数据管理器 1106 和/或其任何元件的内部或外部的分布式单元。

[0184] 例如,系统 1100 可包括一个或多个处理器 1104。例如,系统 1100 可包括存储可由一个或多个处理器 1104 执行的指令的至少一个有形计算机可读存储介质,可执行指令被配置成使至少一个数据处理装置执行与包括在系统 1100 中的各个示例组件相关联的操作,如本文所讨论地。例如,一个或多个处理器 1104 可被包括在至少一个数据处理装置中。数据处理领域的技术人员将理解,存在可根据本文的讨论配置的处理器和数据处理装置的许多配置,而不背离这样的讨论的精神。

[0185] 在本上下文中,“组件”可指的是可被配置成执行某些操作的指令或硬件。这样的指令可被包括在指令的组成群组内,或可跨多于一个群组分布。例如,与第一组件的操作相关联的一些指令可被包括在与第二组件(或更多组件)的操作相关联的指令的群组中。例如,此处的“组件”可以指可由指令实现的一种功能,这些指令可以位于单个实体中或者可以分散或分布在多个实体上,并且可以与同其它组件相关联的指令和/或硬件重叠。

[0186] 根据一示例实施例,数据管理器 1106 可关联于一个或多个用户设备来被实现。例如,数据管理器 1106 可与服务器通信,如以下进一步讨论。

[0187] 例如,一个或多个数据库可经由数据库接口组件 1122 被访问。数据处理领域的技术人员将理解,存在用于存储本文中所讨论的信息的许多种技术,诸如各种类型的数据库配置(例如关系数据库、分层数据库、分布式数据库)和非数据库配置。

[0188] 根据一示例实施例,数据管理器 1106 可包括可存储诸如中间结果等对象的存储器 1124。在该上下文中,“存储器”可包括被配置成存储数据和/或指令的单个存储器设备或多个存储器设备。此外,存储器 1124 可跨多个分布式存储设备。此外,存储器 1124 可以分布在多个处理器中。

[0189] 根据一示例实施例,用户接口组件 1126 可管理用户 1128 与数据管理器 1106 之间的通信。用户 1128 可与接收设备 1130 相关联,接收设备 1130 可与显示器 1132 和其它输入/输出设备相关联。例如,显示器 1132 可被配置成经由内部设备总线通信或经由至少一个网络连接与接收设备 1130 通信。

[0190] 根据示例实施例,显示器 1132 可被实现为平面屏幕显示器、印刷形式显示器、二维显示器、三维显示器、静态显示器、移动显示器、传感显示器(诸如触觉输出、听觉输出以

及任何其它形式输出)以用于与用户(例如,用户 1128)通信。

[0191] 根据一示例实施例,数据管理器 1106 可包括网络通信组件 1134,该网络通信组件 1134 可管理数据管理器 1106 和可经由至少一个网络 1136 与数据管理器 1106 通信的其它实体之间的网络通信。例如,网络 1136 可包括因特网、至少一个无线网络、或至少一个有线网络的至少其中之一。例如,网络 1136 可包括蜂窝网络、无线网络或可支持用于数据管理器 1106 的数据传输的任何类型的网络。例如,网络通信组件 1134 可管理数据管理器 1106 与接收设备 1130 之间的网络通信。例如,网络通信组件 1134 可管理用户接口组件 1126 与接收设备 1130 之间的网络通信。

[0192] 例如,数据不透明接口 1108 可被配置成向任意选择的面向页的访问方法 1110 提供对页数据存储 1112 的接口访问,所述接口访问包括对所述页数据存储 1112 的日志结构化访问。

[0193] 例如,高速缓存层管理器 1138 可包括映射表管理器 1140,映射表管理器 1140 可被配置成在与所述数据不透明接口 1108 相关联的间接地址映射表 1142 上发起表操作,所述表操作包括在所述间接地址映射表 1142 中的各条目上发起原子比较并交换(CAS)操作,以用与所述页数据存储 1112 相关联的页的新状态来替换所述页的先前状态。

[0194] 例如,映射表管理器 1140 可被配置成在与数据不透明接口 1108 相关联的间接地址映射表 1142 上发起表操作,其中间接地址映射表 1142 被共同用于对包括高速缓存层存储 1144 和辅助存储 1146 的数据存储的管理。

[0195] 例如,间接地址映射表 1142 将页的逻辑位置与页的对应物理位置分开,其中页数据存储的用户根据页的物理位置地址值来将页标识符值存储在引用页数据存储的数据结构中的其他位置。

[0196] 例如,更新管理器 1148 可被配置成使用间接地址映射表 1142 中的条目上的无锁比较并交换操作来控制数据更新和管理更新,以实现间接地址映射表 1142 上的原子状态改变。

[0197] 例如,存储层 1149 可包括日志结构化存储层管理器 1150,日志结构化存储层管理器 1150 可被配置成使用针对所述间接地址映射表 1142 中的条目上的无锁比较并交换操作来控制与得自页转储清除的日志结构化相关联的页位置变化。

[0198] 例如,缓冲区管理器 1151 可被配置成经由无锁存更新操作来控制对日志结构化辅助存储缓冲区的更新。因而,例如,多个线程可经由无锁存操作来同时更新日志结构化辅助存储缓冲区。

[0199] 例如,缓冲区管理器 1151 可被配置成发起稳定性操作以用于确定转储清除到日志结构化辅助存储缓冲区的页(具有较低地址),直至第一辅助存储地址自变量,在日志结构化辅助存储中是稳定的。

[0200] 例如,页管理器 1152 可被配置成控制页上的转储清除操作、分配操作、以及释放操作。例如,页管理器 1152 可被配置成基于以下操作来发起高速缓存层存储中的第一页到辅助存储中的位置的转储清除操作:发起所述第一页的页状态到辅助存储缓冲区的复制,发起转储清除增量记录到所述页状态的前缀,所述转储清除增量记录包括指示所述第一页在辅助存储中的存储位置的辅助存储地址以及与调用者相关联的注释,以及基于将所述转储清除增量记录的地址安置在映射表中经由比较并交换(CAS)操作来发起对所述页状态

的更新。

[0201] 例如,页管理器 1152 可被配置成基于发起部分交换增量记录到与第一页相关联的页状态的前缀,来发起高速缓存层存储中的所述第一页的一部分到辅助存储中的位置的交换操作,所述部分交换增量记录包括指示转储清除增量记录的存储位置的主存储器地址,所述转储清除增量记录指示所述第一页的丢失部分在辅助存储中的位置。

[0202] 例如,系统事务管理器 1154 可被配置成提交事务和放弃事务。

[0203] 例如,记录管理器 1156 可被配置成基于更新增量记录操作和替换更新操作来控制更新。

[0204] 例如,时期管理器 1160 可被配置成在第一处理器操作访问页信息之前,发起第一处理器操作加入与第一时期相关联的第一时期登记列表。例如,第一处理器操作可以是线程。

[0205] 例如,页管理器 1152 可被配置成基于将转储清除增量记录的指针安置到映射表中经由比较并交换 (CAS) 操作来将页状态转储清除到辅助存储,所述转储清除增量记录被前缀到经由所述 CAS 操作在所述映射表中被替换的现有页状态。

[0206] 例如,页管理器 1152 可被配置成确定所述 CAS 操作是否成功,并在确定所述 CAS 操作成功的情况下发起写操作以将所述现有页状态写入辅助存储转储清除缓冲区。

[0207] 例如,页管理器 1152 可被配置成在确定 CAS 操作失败的情况下发起对先前分配给现有页的存储空间的空操作。

[0208] 数据处理领域的技术人员将理解,可使用许多不同的技术来用于无锁存、日志结构化存储系统,而不背离本文讨论的精神。

[0209] III. 流程图描述

[0210] 在此讨论的特征被提供为可用数据处理领域的技术人员所理解的多种不同方式来实现,而不背离在此的讨论的精神。这样的特征被解释为仅为示例实施例特征,并且不旨在被解释为限制到仅为那些详细的描述。

[0211] 图 12a-12d 是示出根据一示例实施例的图 11 的系统的示例操作的流程图。在图 12a 的示例中,包括对页数据存储的无锁存访问的对页数据存储的接口访问可被提供给任意选择的面向页的访问方法 (1202)。例如,如上所述,数据不透明接口 1108 可向任意选择的面向页的访问方法 1110 提供对页数据存储 1112 的接口访问,所述接口访问包括对所述页数据存储 1112 的无锁存访问。

[0212] 例如,对页数据存储的接口访问可包括对稳定页数据存储的日志结构化访问 (1204)。例如,如上所述,数据不透明接口 1108 可向任意选择的面向页的访问方法 1110 提供对页数据存储 1112 的接口访问,所述接口访问包括对所述页数据存储 1112 的日志结构化访问。

[0213] 例如,表操作可在与数据不透明接口相关联的间接地址映射表上发起,所述表操作包括在所述间接地址映射表中的各条目上发起原子比较并交换操作,以用与所述页数据存储相关联的页的新状态来替换所述页的先前状态 (1206)。例如,如上所述,映射表管理器 1140 可在与所述数据不透明接口 1108 相关联的间接地址映射表 1142 上发起表操作,所述表操作包括在所述间接地址映射表 1142 中的各条目上发起原子比较并交换 (CAS) 操作,以用与所述页数据存储 1112 相关联的页的新状态来替换所述页的先前状态。

[0214] 例如,如图 12b 所示,间接地址映射表可被共同用于包括高速缓存层存储和辅助存储的数据存储的管理 (1208)。例如,如上所述,映射表管理器 1140 可在与数据不透明接口 1108 相关联的间接地址映射表 1142 上发起表操作,其中间接地址映射表 1142 被共同用于对包括高速缓存层存储 1144 和辅助存储 1146 的数据存储的管理。

[0215] 例如,页的逻辑位置可与页的对应物理位置分开,其中页数据存储的用户根据页的物理位置地址值来将页标识符值存储在引用页数据存储的数据结构中的其他位置 (1210)。例如,如上所述,间接地址映射表 1142 将页的逻辑位置与页的对应物理位置分开,其中页数据存储的用户根据页的物理位置地址值来将页标识符值存储在引用页数据存储的数据结构中的其他位置。

[0216] 例如,可使用间接地址映射表中的条目上的无锁存比较并交换操作来控制数据更新和管理更新,以实现间接地址映射表上的原子状态改变 (1212)。例如,如上所述,更新管理器 1148 可使用间接地址映射表 1142 中的条目上的无锁存比较并交换操作来控制数据更新和管理更新,以实现间接地址映射表 1142 上的原子状态改变。

[0217] 例如,与得自页转储清除的日志结构化相关联的页位置变化可使用间接地址映射表中的条目上的无锁存比较并交换操作来控制 (1214)。例如,如上所述,日志结构化存储层管理器 1150 可使用所述间接地址映射表 1142 中的条目上的无锁存比较并交换操作来控制与得自页转储清除的日志结构化相关联的页位置变化。

[0218] 例如,在图 12c 的示例中,在第一处理器操作访问页信息之前,可发起第一处理器操作加入与第一时期相关联的第一时期登记列表 (1216)。

[0219] 例如,可基于将转储清除增量记录的指针安置到映射表中经由比较并交换 (CAS) 操作来将页状态转储清除到辅助存储,所述转储清除增量记录被前缀到经由所述 CAS 操作在所述映射表中被替换的现有页状态 (1218)。

[0220] 例如,对日志结构化辅助存储缓冲区的更新可经由无锁存更新操作来控制 (1220)。

[0221] 例如,在图 12d 的示例中,基于以下操作来发起高速缓存层存储中的第一页到辅助存储中的位置的转储清除操作:发起所述第一页的页状态到辅助存储缓冲区的复制,发起转储清除增量记录到所述页状态的前缀,所述转储清除增量记录包括指示所述第一页在辅助存储中的存储位置的辅助存储地址以及与调用者相关联的注释,以及基于将所述转储清除增量记录的地址安置在映射表中经由比较并交换 (CAS) 操作来发起对所述页状态的更新 (1222)。

[0222] 例如,可基于发起部分交换增量记录到与第一页相关联的页状态的前缀,来发起高速缓存层存储中的所述第一页的一部分到辅助存储中的位置的交换操作,部分交换增量记录包括指示转储清除增量记录的存储位置的主存储器地址,转储清除增量记录指示所述第一页的丢失部分在辅助存储中的位置 (1224)。

[0223] 数据处理领域的技术人员将理解,可使用许多不同的技术来用于无锁存、日志结构化存储系统,而不背离本文讨论的精神。

[0224] 多年来,在数据处理环境中一直考虑顾客隐私和机密。因此,用于无锁存、日志结构化存储系统的示例技术可使用用户输入和 / 或用户提供的数据,用户经由一个或多个订阅协议 (例如,“服务条款” (TOS) 协议) 向相关联的应用或与这样的分析相关联的服务提

供了许可。例如,用户可提供同意使其输入 / 数据在设备上传送和存储,尽管明确指出(例如,经由用户接受的协议)每一方可控制传送和 / 或存储如何进行,以及如果有则可维持什么级别的存储或存储的持续时间。

[0225] 在此描述的各种技术的各实现可以实现为数字电子电路,或计算机硬件、固件、软件或它们的组合(例如,被配置成执行指令以执行各种功能的装置)。

[0226] 实现可实现为包含在诸如纯传播信号等纯信号中的计算机程序。此类实现在此可被称为经由“计算机可读传输介质”来实现。

[0227] 或者,各实现可实现为被包含在机器可用或机器可读存储设备(例如诸如通用串行总线(USB)存储设备等磁性或数字介质、磁带、硬盘驱动、紧致盘、数字视频盘(DVD)等等)中或在被传播的信号中的计算机程序,用于由数据处理设备(例如,可编程的处理器、计算机或多个计算机)的操作执行或控制数据处理设备的操作。此类实现在此可被称为经由“计算机可读存储介质”或“计算机可读存储设备”实现并由此不同于作为诸如纯传播信号等纯信号的实现。

[0228] 诸如以上描述的计算机程序等的计算机程序能以任何形式的编程语言(包括编译、解释或机器语言)撰写,并能以任何形式部署,包括作为独立程序或作为模块、组件、子例程、或其他适于在计算环境中使用的单元。计算机程序可被有形地体现为机器可使用或机器可读存储设备(例如,计算机可读介质)上的可执行代码(例如,可执行指令)。可实现以上讨论的技术的计算机程序可被部署以在一个站点处的一个计算机或多个计算机上执行或跨多个站点分布并通过通信网络相互连接。

[0229] 各方法步骤可由一个或多个可编程处理器来执行,该一个或多个可编程处理器执行计算机程序以便通过对输入数据进行操作并生成输出来执行功能。一个或多个可编程处理器可并行地执行指令,和 / 或可被以分布式配置来安排以进行分布式处理。本文讨论的示例功能还可由一个或多个硬件逻辑组件执行,且装置可至少部分被实现为一个或多个硬件逻辑组件。例如,不作为限制,可使用的硬件逻辑组件的说明性类型可包括:现场可编程门阵列(FPGA)、程序专用集成电路(ASIC)、程序专用标准产品(ASSP)、片上系统(SOC)系统以及复杂可编程逻辑器件(CPLD)等。

[0230] 适于执行计算机程序的处理器可包括,作为示例,通用和专用微处理器以及任何类型的数字计算机的任意一个或多个处理器。一般地,处理器接收来自只读存储器或随机存取存储器或两者的指令和数据。计算机的元件可包括用于执行指令的至少一个处理器以及用于存储指令和数据的一个或多个存储器设备。一般地,计算机还可包括或可被操作地耦合以接收来自一个或多个大容量存储设备(例如,磁性、磁光盘、或光盘)的数据或将数据发送到大容量存储设备以供存储数据或两者。适于实现计算机程序指令和数据的信息载体包括所有形式的非易失存储器,包括,作为示例,半导体存储设备,例如 EPROM、EEPROM 和闪存设备;磁盘,例如内置硬盘或可移动盘;磁光盘;以及 CD ROM 和 DVD-ROM 盘。处理器和存储器可由专用逻辑电路补充或被合并到专用逻辑电路中。

[0231] 为了提供与用户的交互,各实现可在具有用于向用户显示信息的显示设备(例如,阴极射线管(CRT)、液晶显示器(LCD)、或等离子监视器)以及用户能借此向计算机提供输入的键盘和定点设备(例如,鼠标或追踪球)的计算机上实现。其他种类的设备也能被用以提供与用户的交互;例如,向用户提供的反馈可以是任何形式的感官反馈,例如视觉反

馈、听觉反馈或触觉反馈。例如,可经由任何形式的感官输出来提供输出,这些感官输出包括(但不限于)视觉输出(例如,视觉手势、视频输出)、音频输出(例如,语音、设备声音)、触觉输出(例如,触摸、设备移动)、温度、气味等。

[0232] 此外,来自用户的输入可按照任何形式来接收,包括声音、语音或触觉输入。例如,可经由任何形式的感官输入来从用户接收输入,这些感官输入包括(但不限于)视觉输入(例如,手势、视频输入)、音频输入(例如,语音、设备声音)、触觉输入(例如,触摸、设备移动)、温度、气味等。

[0233] 此外,自然用户界面(NUI)可被用于与用户接口。在这一上下文中,“NUI”可指的是使得用户能够以“自然”方式与设备交互而无需受诸如鼠标、键盘、遥控等输入设备强加的人为约束的任何接口技术。

[0234] NUI技术的示例可包括依赖于语音识别、触摸和指示笔识别、屏幕上和屏幕附近的姿势识别、空中姿势、头部和眼睛跟踪、话音和语音、视觉、触摸、姿势、以及机器智能的那些技术。示例NUI技术包括但不限于:触敏显示、话音和语音识别、意图和目的理解、使用深度相机(如立体相机系统、红外相机系统、RGB(红、绿、蓝)相机系统及其组合)的运动姿势检测、使用加速度计/陀螺仪的运动姿势检测、面部识别、3D显示、头部、眼睛和注视跟踪、沉浸式增强现实和虚拟现实系统,所有这些都提供更自然的接口,以及用于使用电场传感电极(如脑电图仪(EEG)和相关技术)的传感大脑活动的技术。

[0235] 实现可在包括后端组件(例如,作为数据服务器)的计算系统中实现、或可在包括中间软件层组件(例如,应用服务器)的计算系统中实现、或可在包括前端组件(例如,具有用户借此能与实现交互的图形用户界面或web浏览器的客户端计算机)的计算系统中实现、或可在包括这样的后端、中间软件层或前端组件的任意组合的计算系统中实现。各组件可通过任何形式或介质的数字数据通信(例如,通信网络)来互连。通信网络的示例包括局域网(LAN)和广域网(WAN),例如因特网。

[0236] 尽管用结构特征和/或方法动作专用的语言描述了本主题,但可以理解,所附权利要求书中定义的主题不必限于上述具体特征或动作。更确切而言,上述具体特征和动作是作为实现权利要求的示例形式公开的。虽然所描述的各实现的特定特征已经被如此处所描述地解说,但是许多改良、替代、改变与等同物现将由本领域技术人员进行。因此,应当理解,所附权利要求旨在覆盖落入各实施例的范围之内的所有这些改良和改变。

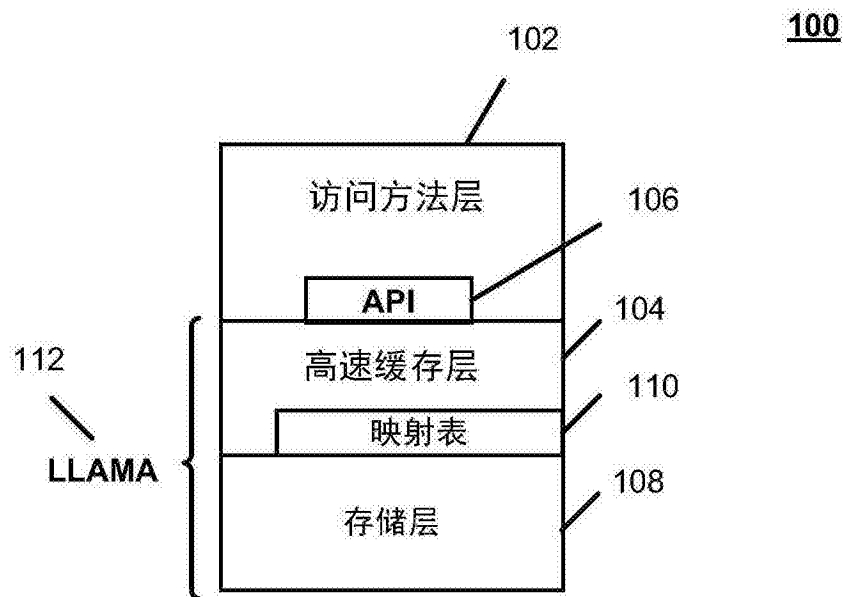


图 1

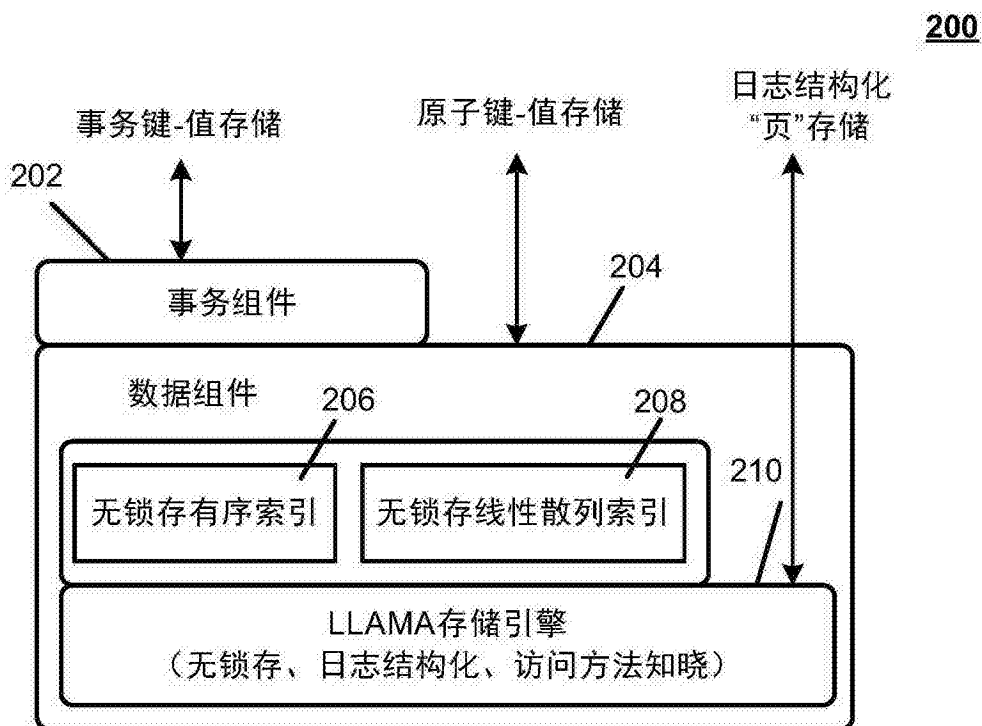


图 2

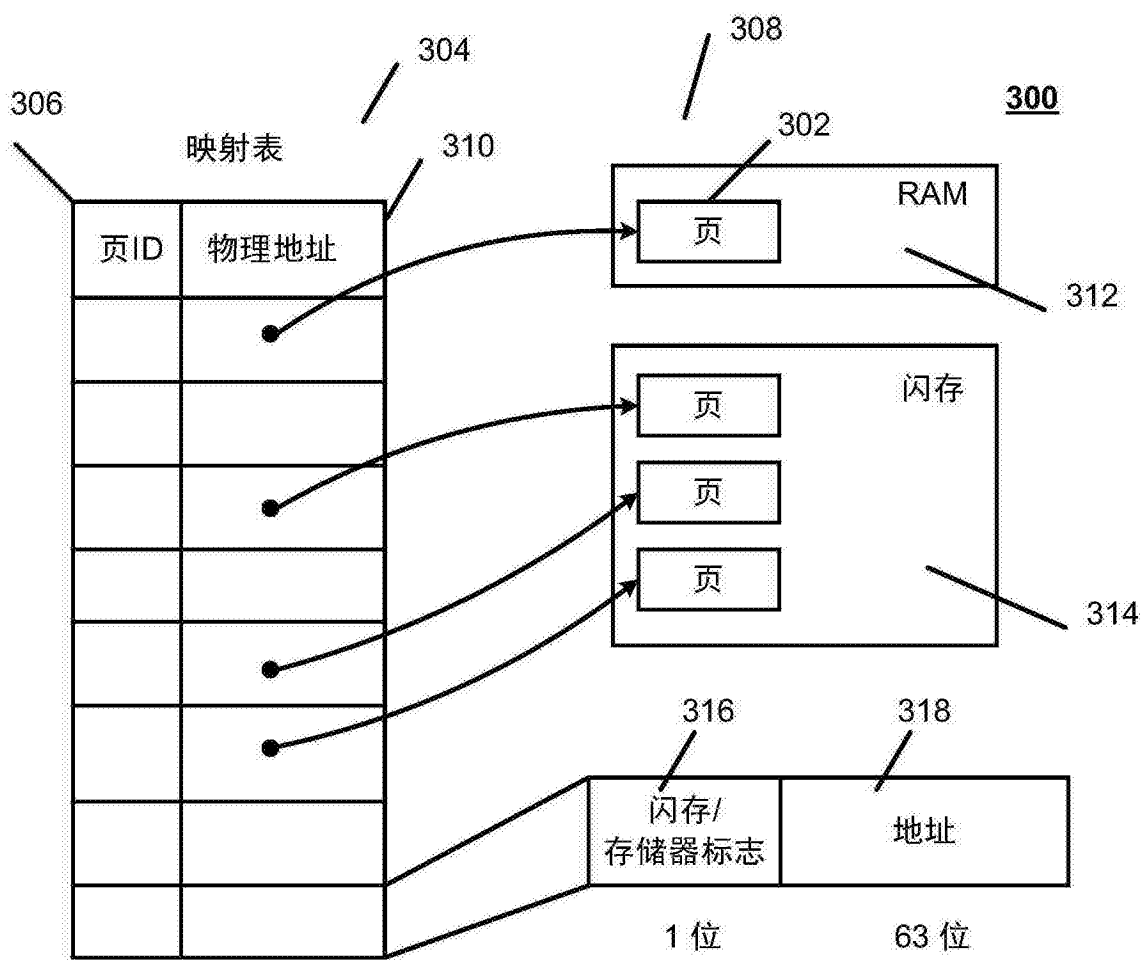


图 3

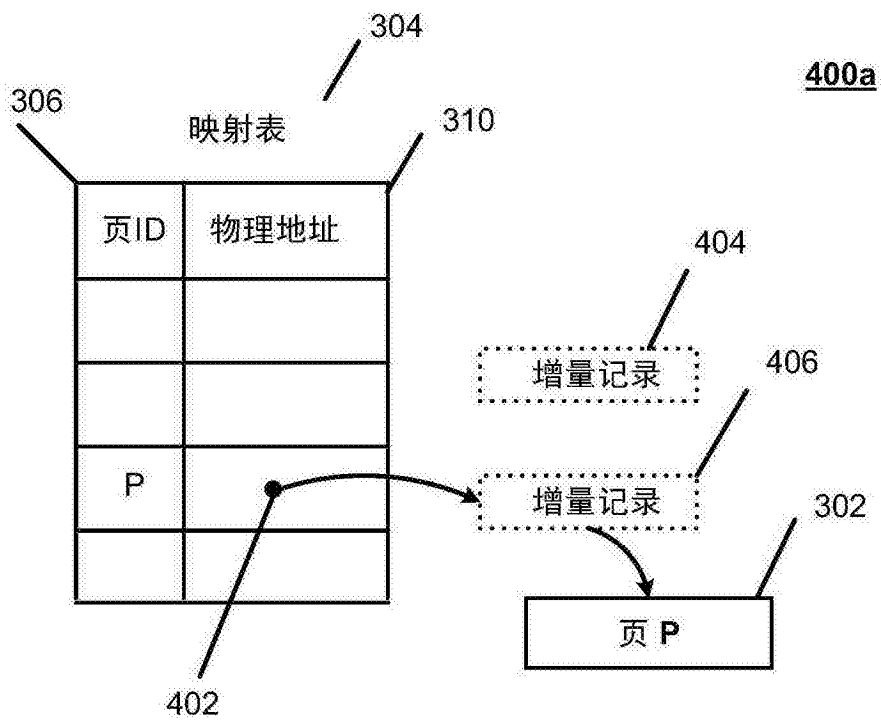


图 4a

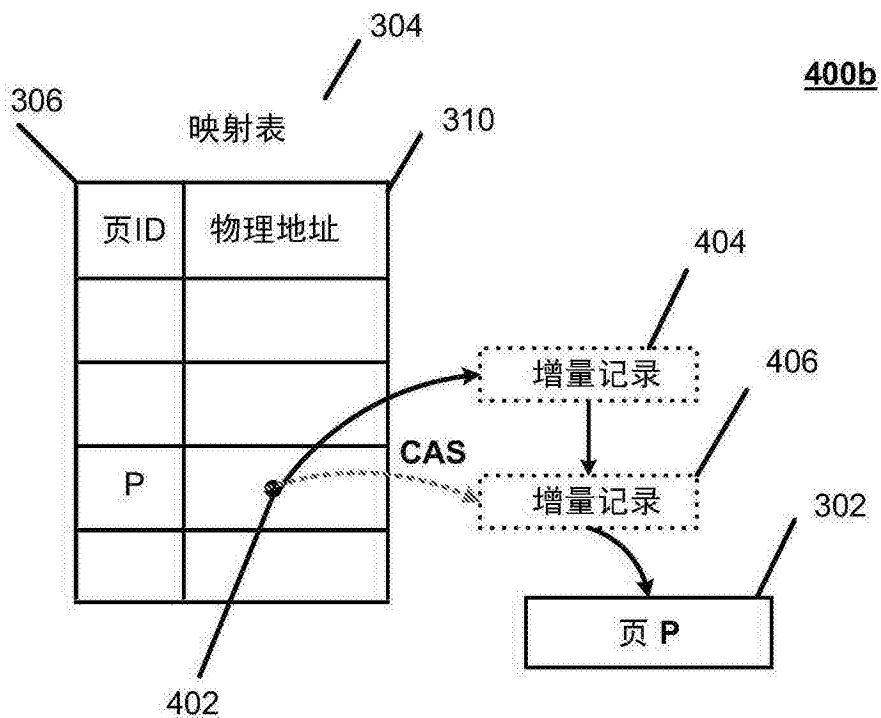


图 4b

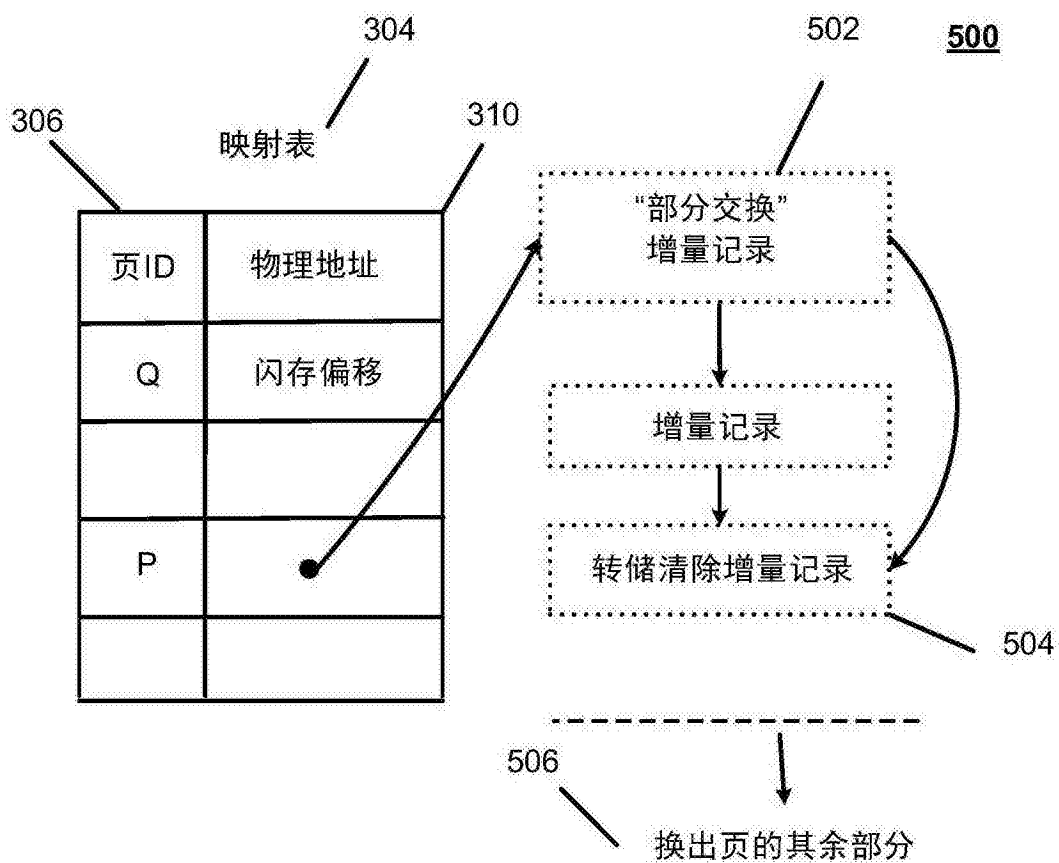


图 5

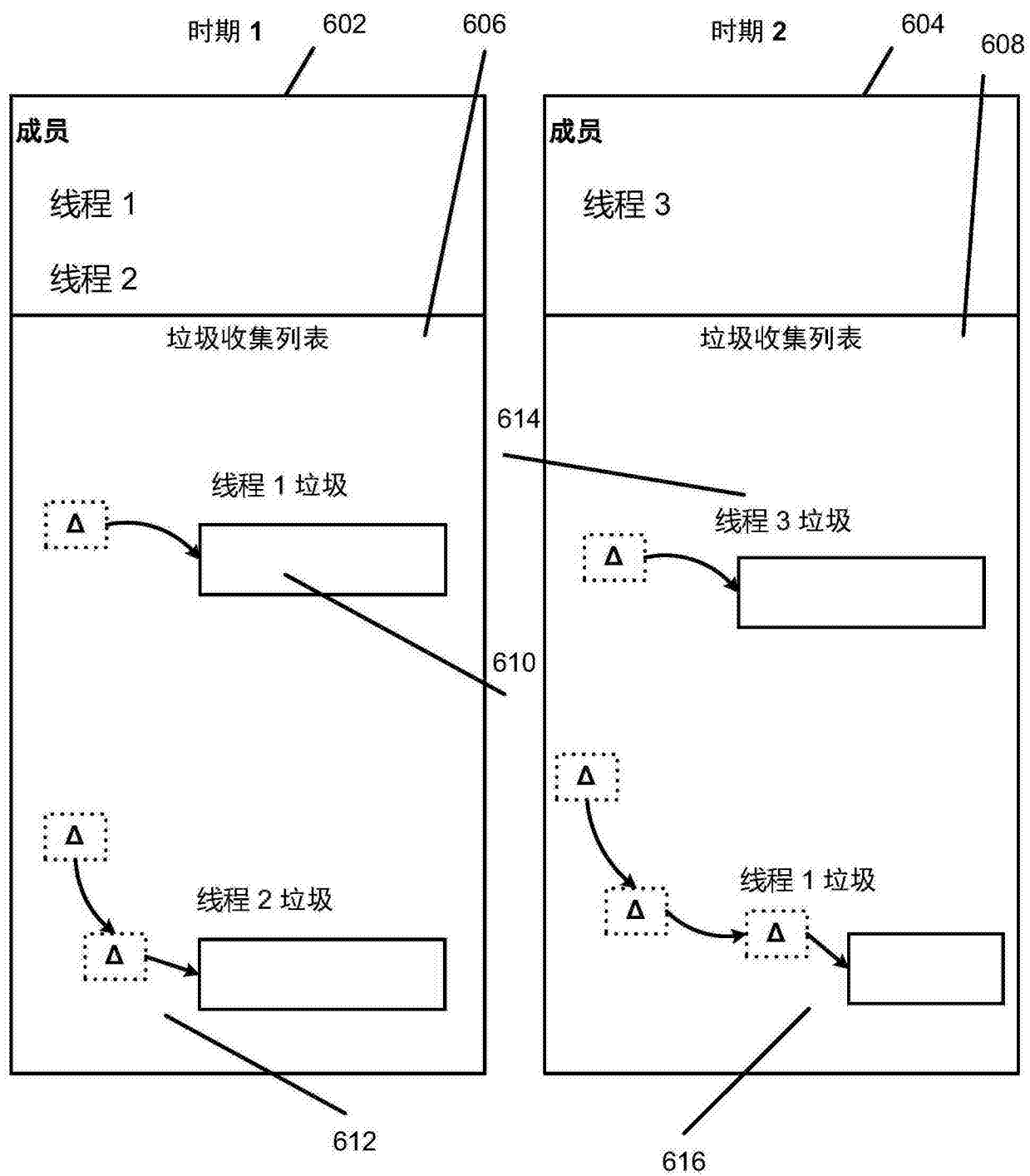
600

图 6

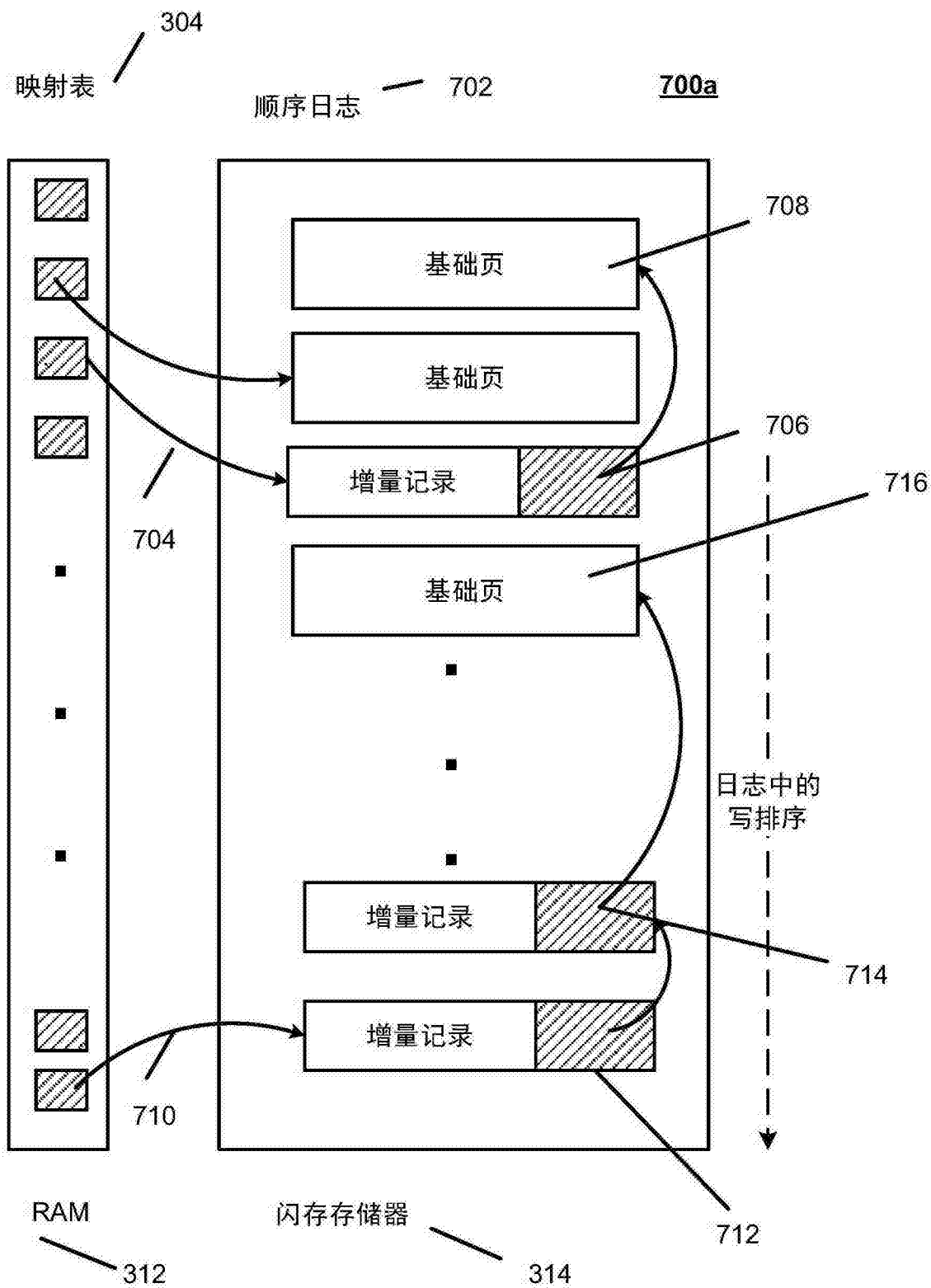


图 7a

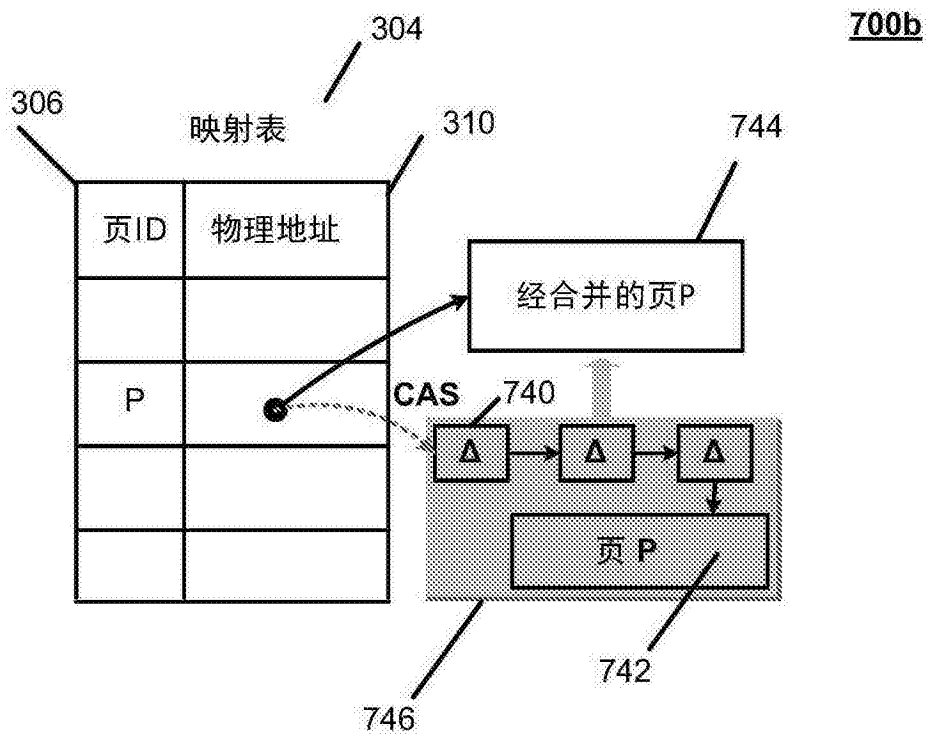


图 7b

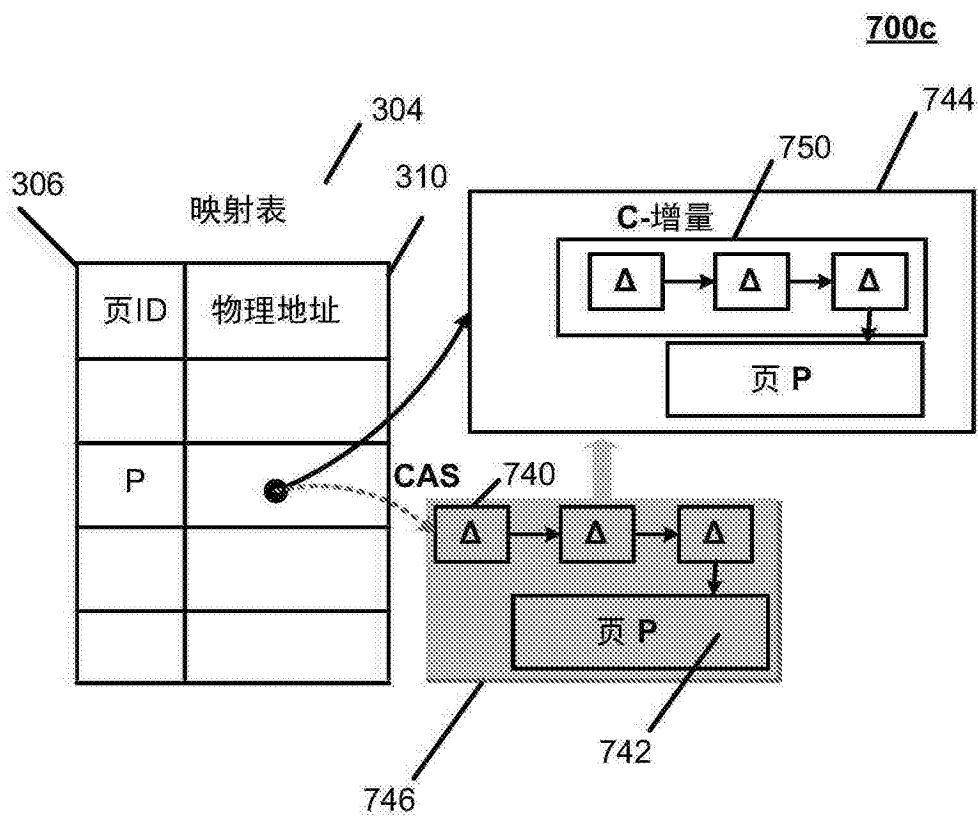


图 7c

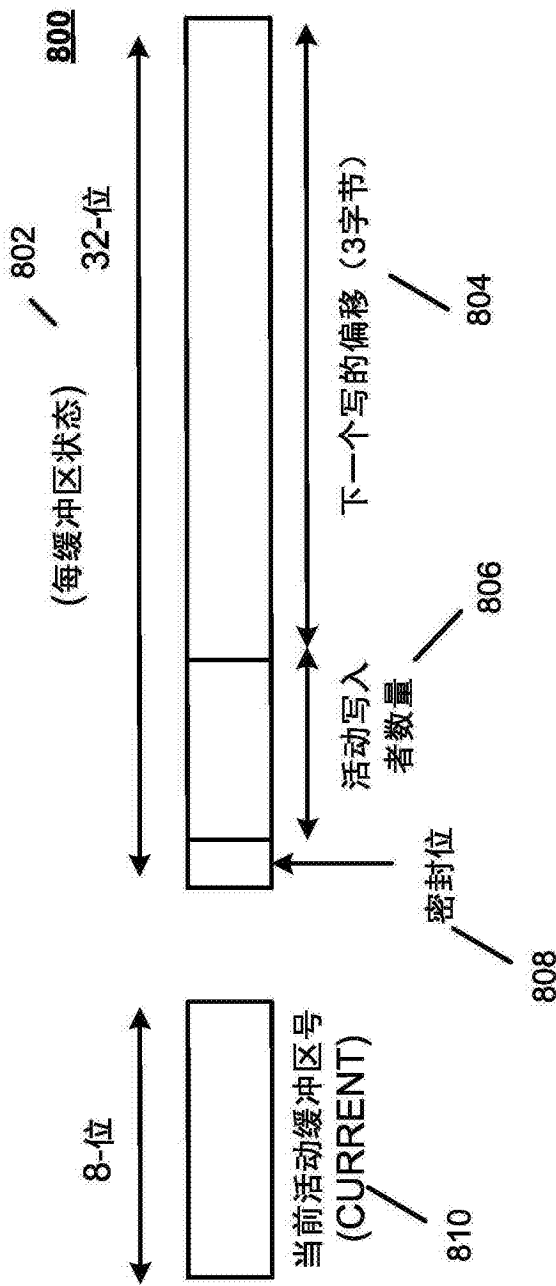


图 8

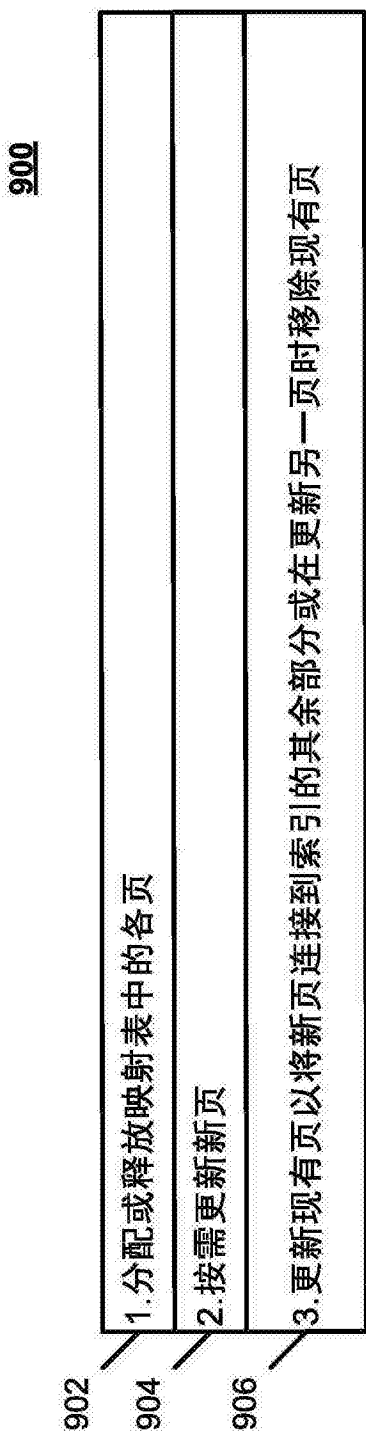


图 9

1000

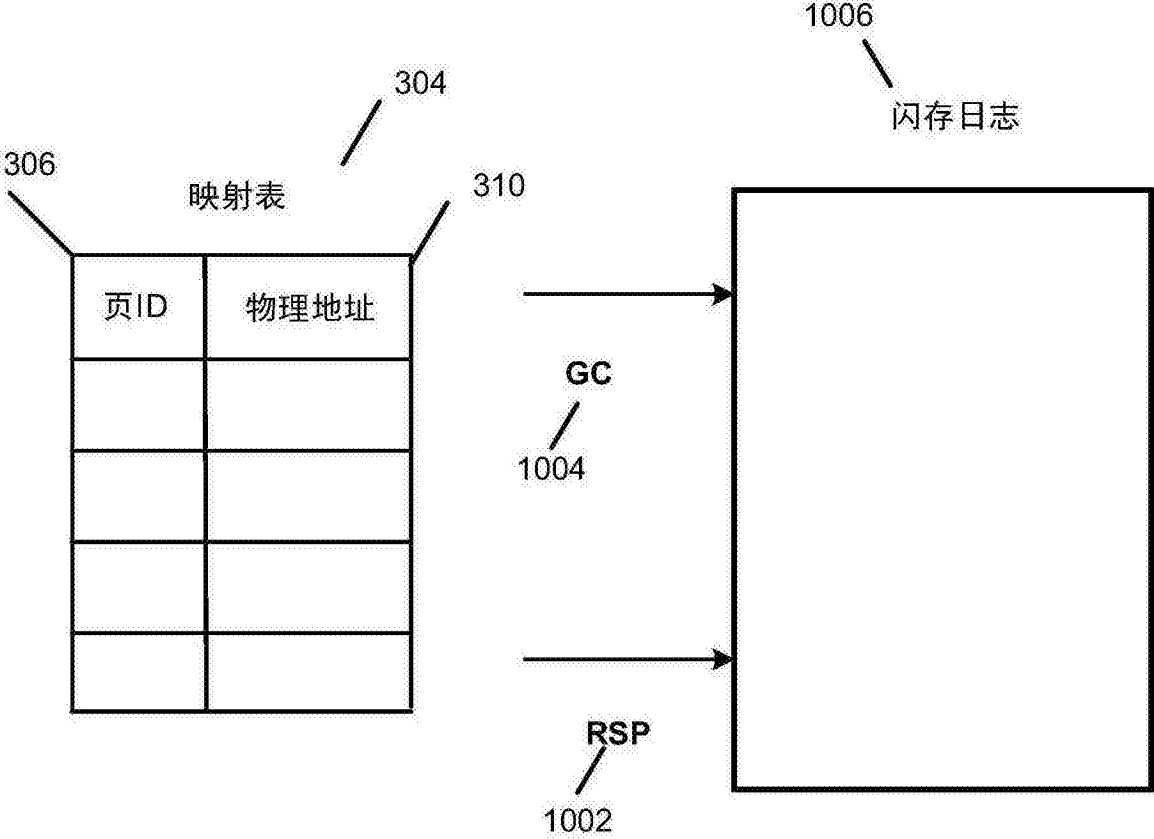


图 10

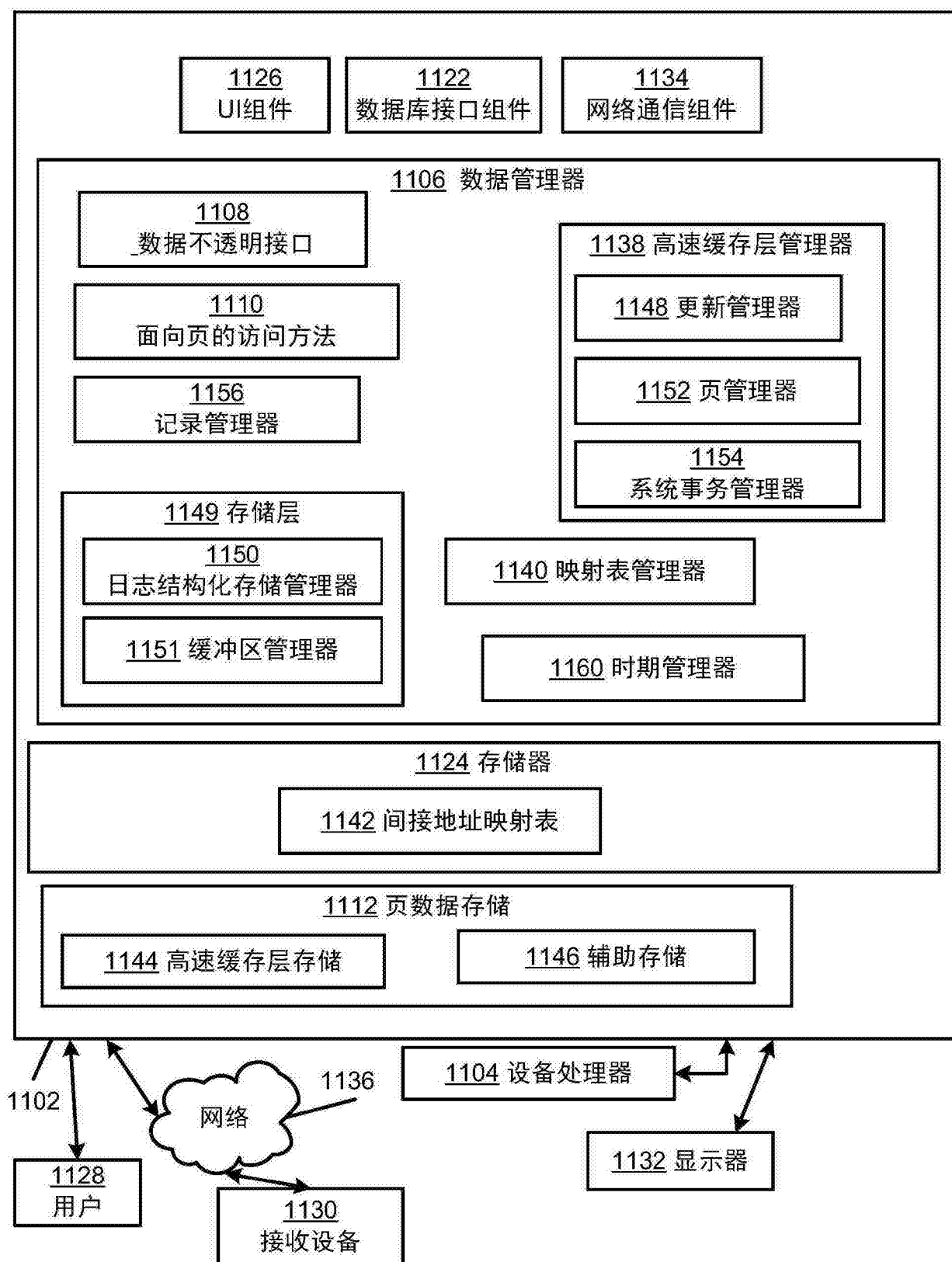
1100

图 11

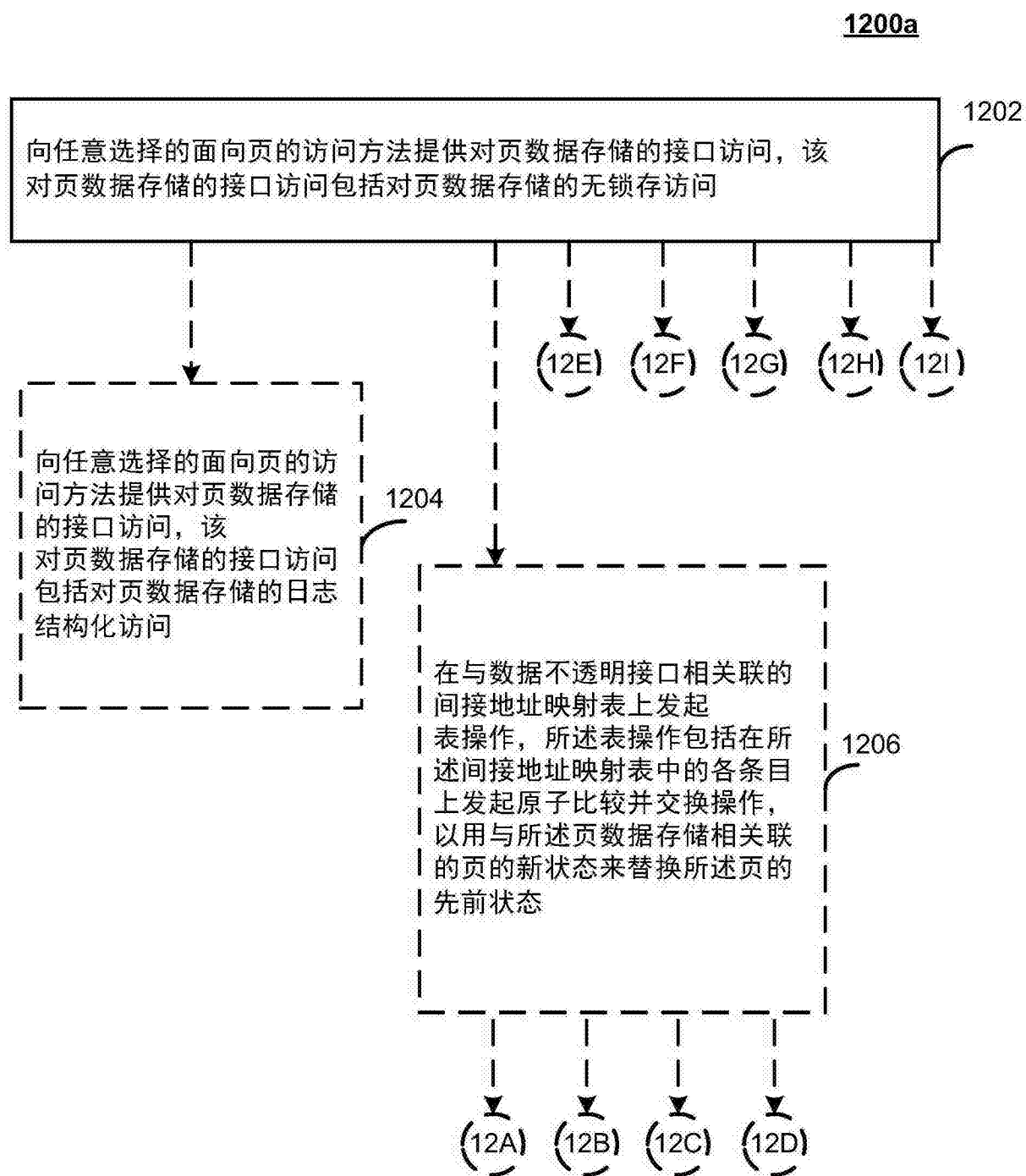


图 12a

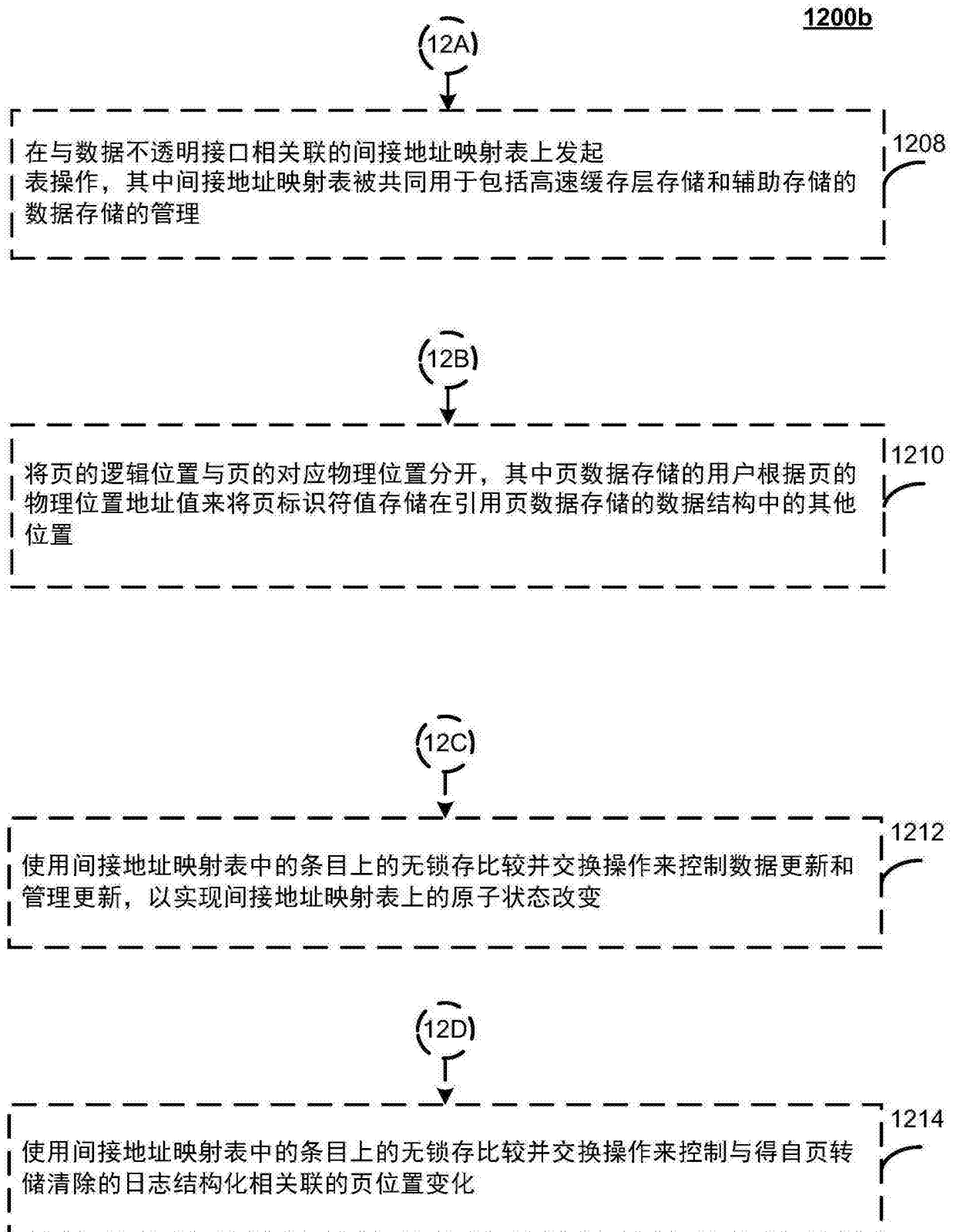


图 12b

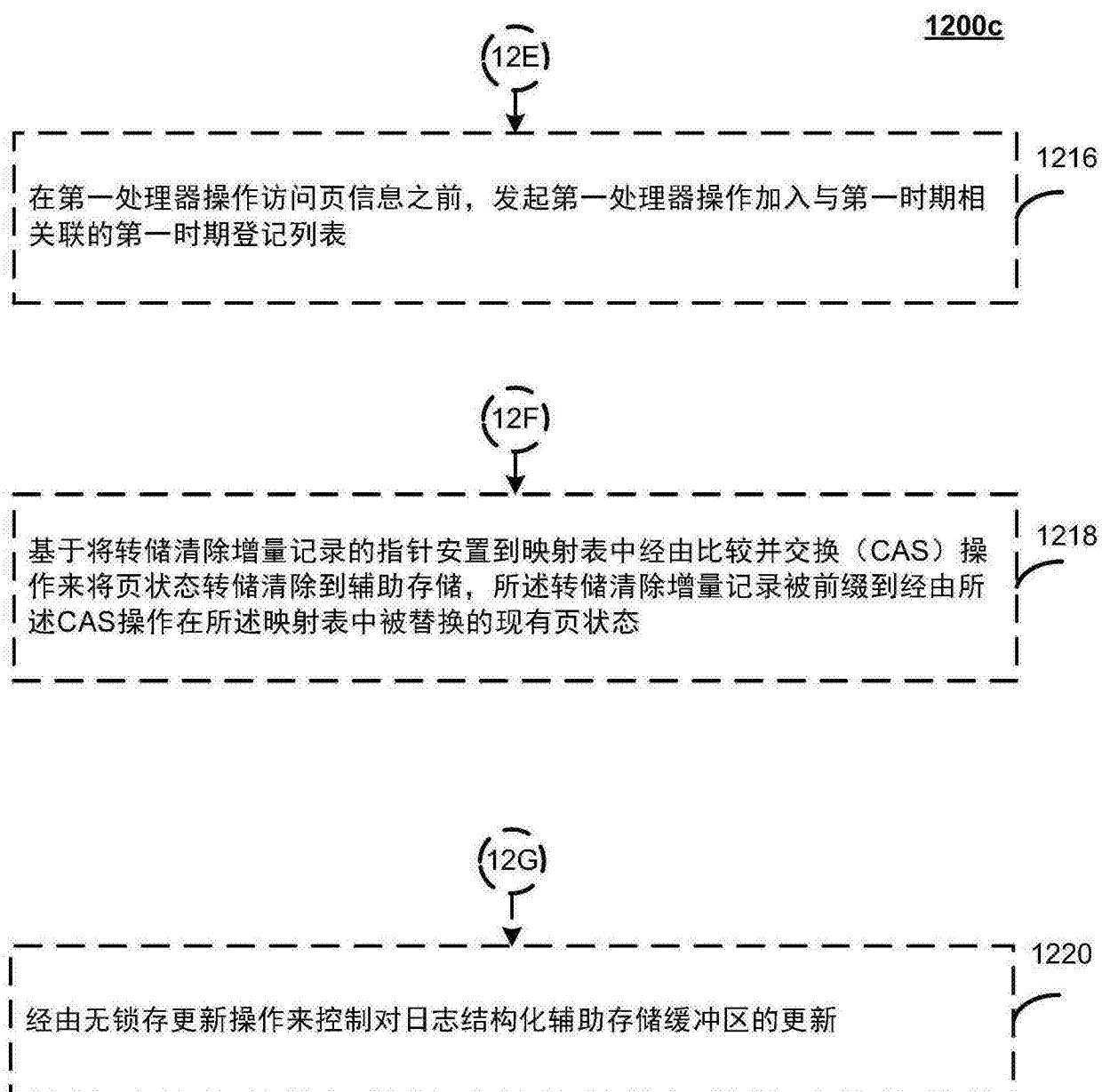


图 12c

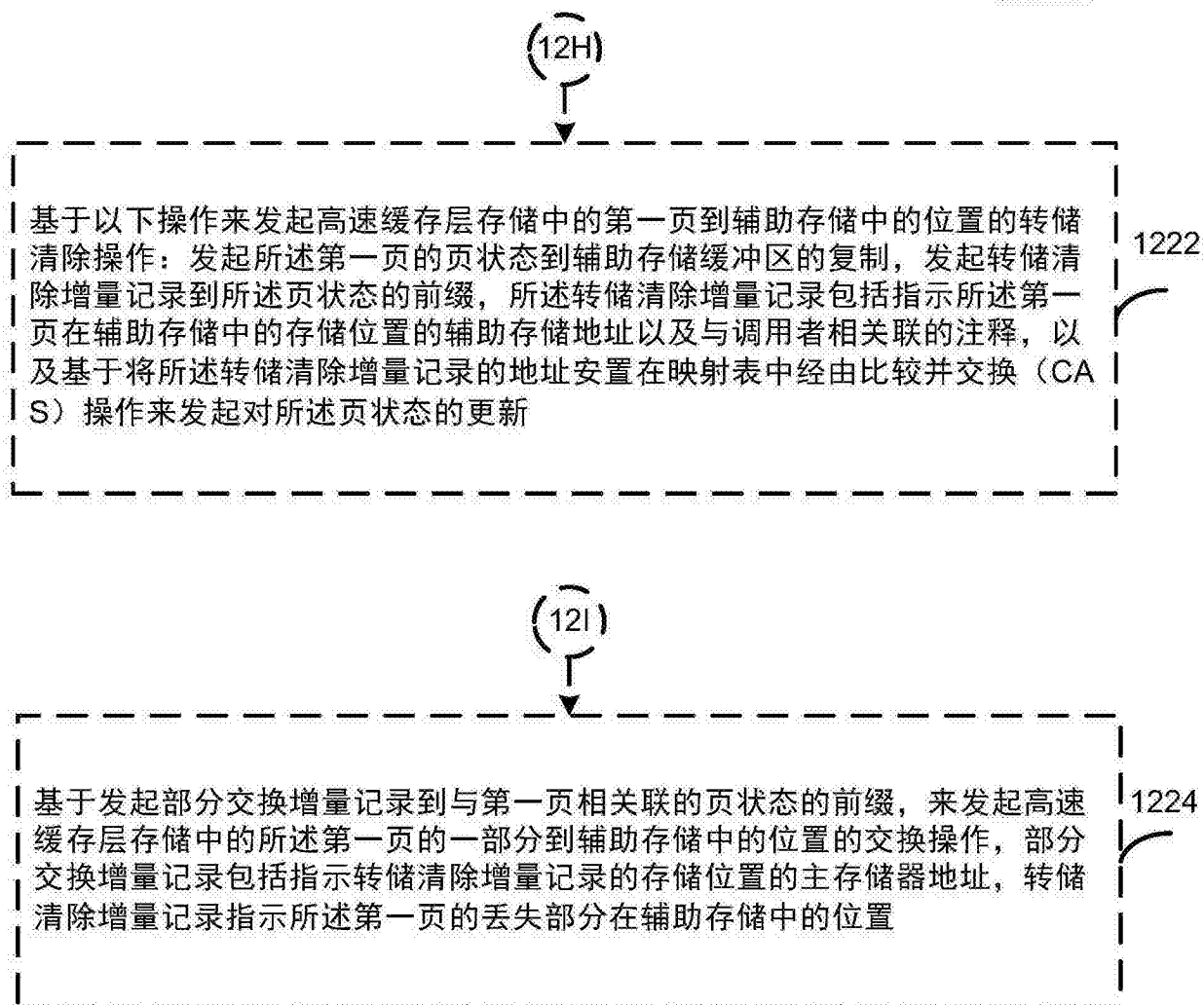
1200d

图 12d