



(12) 发明专利申请

(10) 申请公布号 CN 113168408 A

(43) 申请公布日 2021. 07. 23

(21) 申请号 201980077140.X

(22) 申请日 2019.10.02

(30) 优先权数据

16/156,440 2018.10.10 US

(85) PCT国际申请进入国家阶段日

2021.05.24

(86) PCT国际申请的申请数据

PCT/US2019/054243 2019.10.02

(87) PCT国际申请的公布数据

W02020/076580 EN 2020.04.16

(71) 申请人 美光科技公司

地址 美国爱达荷州

(72) 发明人 A·汤姆林森 G·A·贝克尔

G·S·拉姆达西

(74) 专利代理机构 北京律盟知识产权代理有限公司 11287

代理人 王艳娇

(51) Int.Cl.

G06F 16/22 (2006.01)

G06F 16/28 (2006.01)

G06F 16/2455 (2006.01)

G06F 3/06 (2006.01)

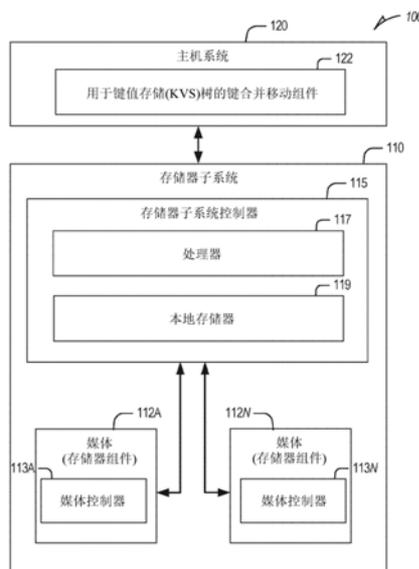
权利要求书3页 说明书24页 附图14页

(54) 发明名称

利用压缩的键值存储树数据块溢出

(57) 摘要

本公开的各方面提供对键值树数据结构的操作:通过合并且重写给定节点内的键值组的键块、同时基于所述给定节点的一或多个子节点是否包括叶节点而重写或推迟重写经合并键值组的值块来合并所述键值组;以及将所述经合并键值组的一或多个部分移动到所述给定节点的一或多个子节点。



1. 一种系统,其包括:

一组存储器组件,其存储键值存储树数据结构,所述键值存储树数据结构包括一组节点,其中所述一组节点中的节点包括键值组序列;以及

处理装置,其以操作方式耦合到所述一组存储器组件,且经配置以执行操作,所述操作包括:

检测将所述键值组序列合并且从所述键值存储树数据结构的所述节点移动到所述节点的一组子节点的条件;以及

响应于检测到所述条件:

确定所述节点的所述一组子节点是否包括叶节点;且

基于确定所述一组子节点是否包括所述叶节点,将所述键值组序列移动到所述一组子节点。

2. 根据权利要求1所述的系统,其中所述将所述键值组序列移动到所述一组子节点包括:

响应于确定所述一组子节点不包括所述叶节点:

合并所述键值组序列以产生经合并键值组,所述经合并键值组包括参考所述键值组序列的一组现有值块的一组新键块,且所述一组新键块基于所述键值组序列的一组现有键块而生成;以及

将所述经合并键值组移动到所述节点的所述一组子节点中。

3. 根据权利要求2所述的系统,其中所述将所述经合并键值组移动到所述节点的所述一组子节点中包括:

将所述经合并键值组划分成一组分离键值组,每个分离键值组被分配给所述一组子节点中的不同子节点;以及

将所述一组分离键值组中的每个分离键值组移动到所述一组子节点中的经分配子节点。

4. 根据权利要求2所述的系统,其中合并所述键值组序列以产生所述经合并键值组包括在生成所述经合并键值之后:

响应于确定所述一组子节点不包括所述叶节点,从所述节点删除所述键值组序列中的每个特定键值组,删除所述特定键值组包括删除所述特定键值组的一或多个键块,同时保留所述特定键值组的一或多个值块。

5. 根据权利要求2所述的系统,其中基于所述键值组序列的所述一组现有键块,通过复制所述一组现有键块以使得所述一组新键块包括对所述一组现有值块的一或多个参考来生成所述一组新键块。

6. 根据权利要求1所述的系统,其中所述节点的所述一组值块中的每个特定值块被分配数据生成编号,所述数据生成编号指示针对所述键值存储树结构初始生成所述特定值块的序列次序,且所述将所述键值组序列移动到所述一组子节点包括:

响应于确定所述一组子节点包括所述叶节点:

合并所述键值组序列以产生包括参考一组新值块的一组新键块的经合并键值组,所述一组新键块基于所述键值组序列的一组现有键块而生成,所述一组新值块基于所述键值组序列的一组现有值块而生成,且所述一组新值块被分配有分配给所述一组现有值块中的任

一值块的特定最大数据生成编号;以及

将所述经合并键值组移动到所述节点的所述一组子节点中。

7. 根据权利要求6所述的系统,其中所述将所述经合并键值组移动到所述节点的所述一组子节点中包括:

将所述经合并键值组划分成一组分离键值组,每个分离键值组被分配给所述一组子节点中的不同子节点;以及

将所述一组分离键值组中的每个分离键值组移动到所述一组子节点中的经分配子节点。

8. 根据权利要求6所述的系统,其中所述合并所述键值组序列以产生所述经合并键值组包括在生成所述经合并键值组之后:

响应于确定所述一组子节点确实包括所述叶节点:

针对所述键值存储树数据结构的每个特定叶节点,通过确定分配给与所述特定叶节点相关联的任何值块的最大数据生成编号,确定一组最大数据生成编号;

确定所述一组最大数据生成编号中的最小数据生成编号;以及

从所述节点删除所述键值组序列中的每个特定键值组,删除所述特定键值组包括:

删除具有小于所述最小数据生成编号的特定数据生成编号的由所述特定键值组的现有键块参考的任何现有值块;以及

删除所述特定键值组的一或多个现有键块。

9. 根据权利要求1所述的系统,其中所述节点的所述一组值块中的每个特定值块被分配数据生成编号,所述数据生成编号指示针对所述键值存储树结构初始生成所述特定值块的序列次序,且所述操作还包括:

确定所述一组最大数据生成编号中的最小数据生成编号;

确定给定值块的特定数据生成编号小于所述最小数据生成编号;以及

响应于确定所述特定数据生成编号小于所述最小数据生成编号,删除所述键值存储树数据结构的给定值块。

10. 根据权利要求1所述的系统,其中所述系统是存储器子系统。

11. 根据权利要求1所述的系统,其中主机系统包括所述处理装置,且存储器子系统包括所述一组存储器组件。

12. 一种方法,其包括:

在一组存储器组件上生成键值存储树数据结构,所述键值存储树数据结构包括一组节点,其中所述一组节点中的节点包括键值组序列;

由处理装置检测将所述键值组序列合并且从所述节点移动到所述节点的一组子节点的条件;以及

响应于检测到所述条件:

由所述处理装置确定所述节点的所述一组子节点是否包括叶节点;且

由所述处理装置基于确定所述一组子节点是否包括所述叶节点,将所述键值组序列移动到所述一组子节点。

13. 根据权利要求12所述的方法,其中所述将所述键值组序列移动到所述一组子节点包括:

响应于确定所述一组子节点不包括所述叶节点：

合并所述键值组序列以产生经合并键值组，所述经合并键值组包括参考所述键值组序列的一组现有值块的一组新键块，且所述一组新键块基于所述键值组序列的一组现有键块而生成；以及

将所述经合并键值组移动到所述节点的所述一组子节点中。

14. 根据权利要求12所述的方法，其中节点的所述一组值块中的每个特定值块被分配数据生成编号，所述数据生成编号指示针对所述键值存储树结构初始生成所述特定值块的序列次序，且所述将所述键值组序列移动到所述一组子节点包括：

响应于确定所述一组子节点包括所述叶节点：

合并所述键值组序列以产生包括参考一组新值块的一组新键块的经合并键值组，所述一组新键块基于所述键值组序列的一组现有键块而生成，所述一组新值块基于所述键值组序列的一组现有值块而生成，且所述一组新值块被分配有分配给所述一组现有值块中的任一值块的特定最大数据生成编号；以及

将所述经合并键值组移动到所述节点的所述一组子节点中。

15. 一种包括指令的非暂时性机器可读存储媒体，所述指令在由处理装置执行时使所述处理装置：

在一组存储器组件上存取键值存储树数据结构，所述键值存储树数据结构包括一组节点，其中所述一组节点中的节点包括键值组序列；

检测将所述键值组序列合并且从所述节点移动到所述节点的一组子节点的条件；以及

响应于检测到所述条件：

确定所述节点的所述一组子节点是否包括叶节点；且

基于确定所述一组子节点是否包括所述叶节点，将所述键值组序列移动到所述一组子节点。

利用压缩的键值存储树数据块溢出

[0001] 优先权申请

[0002] 本申请要求2018年10月10日提交的第16/156,440号美国申请的优先权益,所述美国申请以全文引用的方式并入本文中。

技术领域

[0003] 本公开的实施例大体上涉及存储器子系统,且更具体来说,涉及键值存储(key-value store,KVS)树数据结构的操作。

背景技术

[0004] 存储器子系统可以是存储系统,如固态硬盘(SSD),且可包含存储数据的一或多个存储器组件。存储器组件可例如为非易失性存储器组件和易失性存储器组件。一般来说,主机系统可利用存储器子系统以在存储器组件处存储数据以及从存储器组件检索数据。

附图说明

[0005] 根据下文给出的详细描述和本公开的各种实施例的附图,将更充分地理解本公开。

[0006] 图1是说明根据本公开的一些实施例的包含存储器子系统的实例计算环境的框图。

[0007] 图2是根据本公开的一些实施方案的用于键值存储(KVS)树的实例键合并移动的框图。

[0008] 图3到5是根据本公开的一些实施方案的用于键合并移动的实例方法的流程图。

[0009] 图6是说明根据本公开的一些实施方案的可通过键合并移动进行操作的实例KVS树的框图。

[0010] 图7A和7B是说明根据本公开的一些实施方案的在一组子节点不包括叶节点时对一或多个键值组执行的实例键合并移动的框图。

[0011] 图8A和8B是说明根据本公开的一些实施方案的在一组子节点仅包括一或多个叶节点时对一或多个键值组执行的实例键合并移动的框图。

[0012] 图9A到9C提供说明在其中执行用于键合并移动的方法的实例实施例的上下文中的计算环境的组件之间的交互的交互图。

[0013] 图10是说明根据本公开的一些实施例的呈计算机系统形式的机器的图形表示的框图,在所述计算机系统内可执行指令集以引起机器执行本文中所论述的方法中的任何一种或多种。

具体实施方式

[0014] 本公开的各方面涉及合并和移动可供存储器子系统使用或结合存储器子系统使用的键值树数据结构中的键值组。存储器子系统在下文也称为“存储器装置”。存储器子系

统的实例是存储系统,例如SSD。在一些实施例中,存储器子系统是混合式存储器/存储子系统。通常,主机系统可利用包含一或多个存储器组件的存储器子系统。主机系统可提供数据(例如,经由写入请求)以存储于存储器子系统处,且可请求从存储器子系统检索数据(例如,经由读取请求)。

[0015] 存储器子系统可包含可存储来自主机系统的数据的多个存储器组件。存储器子系统可进一步包含存储器子系统控制器,所述存储器子系统控制器可与存储器组件中的每一者通信,以响应于从主机系统接收到的请求而在存储器组件处执行例如读取数据、写入数据或擦除数据的操作。存储器子系统的存储器组件中的任何一或多个存储器组件可包含媒体控制器以管理存储器组件的存储器单元、与存储器子系统控制器通信以及执行从存储器子系统控制器接收的存储器请求(例如,读取或写入)。

[0016] 在例如数据库存储和卷数据存储(volume data storage)(例如,云存储)的一些应用中,键值数据结构用以将数据存储于数据存储媒体上,所述数据存储媒体例如由一或多个存储器装置实施且呈现为包括一或多个媒体数据块(媒体块)的单个逻辑数据存储卷的数据存储媒体池(媒体池)。键值存储(KVS)可包括一或多个键值数据结构以存储和搜索键值对。键值数据结构可准许高效搜索键值对的所存储数据、准许高效存储稀疏数据或准许高效存储可搜索的数据。通常,键值数据结构接受键值对以进行存储,且经配置以对基于键的值查询作出响应。键值数据结构可包括例如树数据结构之类的结构,其实例包含日志结构的合并树(LSM树)和键值存储(KVS)树(这里也称为键值存储树数据结构或KVS树数据结构)。

[0017] 本公开的各方面提供各种实施例,用于通过合并键值对数据中的键数据、同时推迟键值对数据中的值数据的重写来合并(例如压缩)键值树数据结构(KVS树)的节点的键值对数据,且接着将经合并(例如经压缩)键数据移动(例如溢出)到节点的子节点中的至少一者中。这样,此类实施例可准许值数据在KVS树的一或多个内部节点之间共享,同时减少在键值对数据沿着KVS树(从根节点到叶节点)向下流动时在KVS树数据中重写值数据的次数。例如,一些实施例限制特定值数据在KVS树中仅被写两次(例如,当所述特定值数据在KVS树的根节点处初始地写入时以及当所述特定值数据到达叶节点且在叶节点处重写时)。取决于实施例,本文所描述的操作可在主机系统上、在存储器子系统上或这两者的某种组合上执行。在不使用本文所描述的实施例的情况下,值数据可能被写入H次,其中H为KVS树的高度。值重写次数的减少可有益于KVS树和使用KVS树的存储器子系统的性能,在值数据通常在数据大小方面比键数据大许多的情况下尤其如此。减少在KVS树中的值重写次数不仅可减少关于KVS树的读取和写入放大,且还可减少关于用于存储KVS树的数据存储装置的装置输入/输出(I/O)。

[0018] 如本文所使用,键合并移动操作共同地指代将节点的键值对数据中的键数据(例如,键块)合并(例如,压缩)、同时推迟键值对数据中的值数据(例如,值块)的重写且接着将经合并(例如,经压缩)键数据移动(例如,溢出)到节点的子节点中的至少一者中的各种实施例的操作。键合并移动操作还可称为k溢出压缩操作,其可对键值组进行k压缩且接着溢出。根据各种实施例,KVS树的节点内的数据包括键值对的一组键块(用于存储键)和一组值块(用于存储值)。以此方式,KVS树的节点可存储与其对应的值分开的键,这会提供优于LSM树的KVS树性能益处。对于此类实施例,本文所描述的操作可实现在节点内合并(例如,压

缩) 键块, 且将经合并 (例如, 经压缩) 键块从节点移动 (例如溢出) 到一或多个子节点, 同时将值块原样保持在节点内且准许那些值块被KVS树的其它节点共享。通过合并键块 (例如, 作为对KVS树执行的无用单元收集处理的部分), 一些实施例回收 (例如, 释放) 较旧的或删除的键块在数据存储媒体 (例如, 媒体池) 上占用的存储器空间 (例如, 媒体块), 且可进一步减小KVS树的总体数据大小, 这可提高KVS树上的可搜索性。通过将键块移动 (例如, 溢出) 到节点的子节点之一, 一些实施例将较旧的数据 (例如, 较旧的键块和其参考的值块) 推入KVS树中更深, 同时释放节点中的存储器空间以接收较新的数据 (例如, 键块)。

[0019] 如本文所使用, KVS树包括树数据结构, 所述树数据结构包括父节点与子节点之间具有基于键的预定派生 (例如, 而非树的内容) 的连接。每个节点可包括经排序 (例如按时间排序) 的键值组 (在本文中也称为kvset) 序列。在按时间排序的情况下, 序列中靠后的键值组可表示较旧键值组。所述kvset可包括一或多个键值对, 其准许值连同参考所述值的对应键一起存储在KVS树中。因此, 在KVS树内, kvset用作在KVS树的节点中组织的个别键和值存储区。给定kvset内的键值对可按键分类。给定kvset内的每个键对于所述kvset中的其它键可以是唯一的; 然而, KVS树内的键可能不是唯一的 (例如, 单个节点内或KVS树的不同节点中的两个不同kvset可包含相同的键)。每个kvset在写到节点后可以是不变的 (例如, 在放置/存储在节点中后, kvset不会改变)。虽然节点内的kvset可不变, 但kvset可被删除或kvset的一些或全部数据内容可被添加到新kvset。

[0020] kvset可包括键树以存储kvset的键值对的键条目, 其中给定键条目可包括键和对值的参考两者。多种数据结构可用于高效存储和检索例如二叉搜索树、B树等键树 (例如, 其可能甚至不是数据树) 中的唯一键。举例来说, 键存储在键树的叶节点中, 其中键树的任何子树中的最大键可在最右子节点的最右条目中, 键树的第一节点的最右边缘链接到键树的子节点, 且以键树的所述子节点为根的子树中的所有键可大于键树的第一节点中的所有键。替代地, 在本文所描述的另一实例中, 可基于以基数为基础的键分布 (例如使用键的散列的分布) 存储键条目。

[0021] 对于一些实施例, kvset的键条目存储在可包括主键块和零或多个扩展键块的一组键数据块 (也称为键块或kblock) 中。所述一组键块的成员可对应于用于由例如SSD、硬盘驱动器等存储器装置实施的数据存储媒体的媒体数据块 (媒体块)。每个键块可包括用以将其标识为键块的标头, 且kvset的主键块可包括用于kvset的一或多个扩展键块的媒体块标识列表。

[0022] 主键块可包括到kvset的键树的标头。所述标头可包括辅助或促进与键或kvset的交互的许多值。举例来说, 主键块或其中存储的标头可包括kvset的键树中的最低键的副本或kvset的键树中的最高键的副本。主键块可包括用于kvset的键树的媒体块标识列表。另外, 主键块可包括用于kvset的布隆过滤器 (bloom filter) 的布隆过滤器标头, 且主键块可包括用于kvset的布隆过滤器的媒体块标识列表。

[0023] 对于一些实施例, kvset的值存储在在一组值数据块 (在本文中也称为值块或vblock) 中。KVS树中的每个特定值块可具有与其相关联的数据生成编号, 所述数据生成编号指示针对KVS树初始生成所述特定值块的序列次序。以此方式, 特定值块的数据生成编号可充当关于何时初始生成所述特定值块的时间戳。例如, 对于生成且添加到KVS树的根节点 (例如, 其kvset) 的第一值块, 数据生成编号可以值“1”开始, 且生成且添加到KVS树的根节

点(例如,其kvset)的第二值块将具有数据生成编号“2”。数据生成编号随着生成且通过根节点添加到KVS树的每个新值块而增加。

[0024] 所述一组值块的成员可对应于用于由存储器装置实施的数据存储媒体的媒体数据块(媒体块),如本文中所提及,所述存储器装置可包括SSD、硬盘驱动器等。每个值块可包括用以将其标识为值块的标头。值块可包括针对其间无分隔的一或多个值的存储区段,其中第一值的位可能延到数据存储媒体上的第二值的位,而两值之间没有防护、容器或其它分隔符。对于各种实施例,kvset的主键块包括用于kvset的一组值块中的值块的媒体块标识列表。以此方式,主键块可管理对kvset内的值块的存储参考。

[0025] 对于一些实施例,与键相关联的数据标记(下文称作铭碑(tombstone))用于指示对应于键的值已被删除。铭碑可驻存在与键相关联的键条目中,且可不针对键值对消耗值块空间。根据一些实施例,铭碑标记与键相关联的值的删除,同时避免可能昂贵的从KVS树清除所述值的操作。对于一些实施例,当针对给定键按时间排序搜索KVS树期间遇到铭碑时,搜索过程会知道对应于给定键的值已被删除,即使与给定键相关联的键值对的到期版本驻存在KVS树内的较低(例如,较旧)位置处也是如此。

[0026] 对于一些实施例,主键块包含用于kvset的一组度量。举例来说,所述一组度量可包括以下中的一或多个:存储在kvset中的键的总数目;或具有存储在kvset中的铭碑值的键的数目;存储在kvset中的键的所有键长度的总和;存储在kvset中的键的所有值长度的总和。最末两个度量可提供kvset消耗的至少大概(如果不精确)的存储量。所述一组度量还可包括例如kvset的值块中未被参考的数据的量(例如未被参考值)。最后的这个度量可提供可在维护操作(例如对KVS树执行的无用单元收集操作)中回收的空间的估计值。

[0027] 响应于多种触发条件,例如与给定节点中符合指定或计算的准则的一或多个kvset相关的条件,可执行各种实施例的键合并移动操作。此类kvset相关准则的实例包括但不限于:给定节点内kvset的数目;给定节点添加(例如获取)新的kvset(例如由于将kvset从给定节点的父节点移动(例如溢出)到给定节点);释放关于给定节点的资源(例如媒体块);给定节点内一或多个kvset的总大小;或可供用于无用单元收集的一或多个kvset中的数据量。kvset中可供用于无用单元收集的数据的一个实例包括kvset中例如通过较新kvset中的键值对或铭碑而呈现为过时的一或多个键值对或铭碑,或已违反约束(例如存留时间约束)的键值对。另一实例包括用于对KVS树执行维护(例如无用单元收集)的条件。又一实例条件包括(例如从主机系统的软件应用程序或操作系统)接收请求以发起关于KVS树的一或多个节点的键合并移动操作,其中所述请求还可指定是对整个kvset序列还是对kvset子序列进行操作。

[0028] 对于一些实施例,本文所描述的键合并移动操作包括以下的组合:通过合并kvset的键块(例如键压缩)同时推迟对kvset的值块的重写来合并给定节点内的kvset(例如kvset序列);以及遍历KVS树(到给定节点的一或多个子节点)以将所得经合并kvset的部分放置到所述一或多个子节点中。通过键合并移动操作接收(以进行操作的)kvset可包括给定节点中的一些或全部kvset,且另外可包括给定节点的两个或更多个kvset的在时间上连续的序列。举例来说,键合并移动操作接收以进行操作的kvset可包括给定节点的整个kvset序列或仅较大kvset序列末端处的kvset子序列。因此,如本文所使用,kvset序列可表示给定节点的整个kvset序列或给定节点的kvset子序列。

[0029] 根据一些实施例,生成KVS树且将其存储在由存储器子系统实施的数据存储媒体上,其中所述KVS树可用于将数据作为一或多个键值对存储在所述数据存储媒体上。对于一些实施例,在经触发以相对于KVS树的给定节点执行后(例如基于触发条件),键合并移动确定给定节点的一组子节点(例如所有子节点)是否包括叶节点,基于一或多个子节点是否包括叶节点而合并给定节点的kvset(例如kvset序列),且接着将所得经合并kvset移动到子节点中的一或多个者。

[0030] 响应于确定所述一组子节点不包括叶节点(例如,所述组中没有子节点是叶节点),键合并移动操作可合并节点的kvset序列(例如时间序列)以产生经合并kvset,其中所得经合并kvset包括参考kvset序列的一组现有值块的一组新键块。可基于kvset序列的一组现有键块(例如从其复制)来生成所述一组新键块。举例来说,可基于kvset序列的一组现有键块,通过将所述一组现有键块的值(例如,键值以及对现有值块的参考)复制到所述一组新键块来生成所述一组新键块。kvset序列中未被所述一组新键值块参考的那些现有值块被保留(例如,未删除),但被视为未被参考且可由不同kvset(例如,KVS树的内部节点的两个kvset)的两个键块共享。在已生成所述一组新键块之后,键合并移动操作可从节点删除键值组序列中的每个特定键值组,且删除每个特定键值组的所有键块,同时保留每个特定键值组的所有值块(例如,原样保留)。保留的值块可包括所述一组新键块所参考的值块、未被所述一组新键块中的任何一者参考的值块,或这两者。kvset序列的(被原样保留的)所有值块可移动到经合并kvset。前文操作可在本文中统称为键压缩(k压缩),其可视为用以移除过时的键块和被那些过时的键块占用的空闲资源(例如,数据存储媒体的媒体块)的无用单元收集形式。

[0031] 替代地,响应于确定所述一组子节点仅包括一或多个叶节点,键合并移动操作可合并节点的kvset序列(例如,时间序列)以产生经合并kvset,其中所得经合并kvset包括参考一组新值块的一组新键块,其中所述一组新键块基于所述kvset序列的一组现有键块而生成,且其中所述一组新值块基于所述kvset序列的一组现有值块而生成。举例来说,可基于kvset序列的所述一组现有键块,通过将所述一组现有键块的值(例如,键值)复制到所述一组新键块且使(所述一组新键块中的)新键块分别参考对应于所述一组现有键块所参考的现有块的(所述一组新值块中的)新值块来生成所述一组新键块。可基于kvset序列的所述一组现有值块,通过将所述一组现有值块的值复制到所述一组新值块来生成所述一组新值块。所述一组新值块可被分配(例如,承袭)分配给所述一组现有值块中的任何值块的最大(例如,最大值)数据生成编号。

[0032] 另外,对于一些实施例,响应于确定所述一组子节点包括至少一个叶节点和至少一个非叶节点(例如,KVS树不平衡且因此所述一组子节点可包括叶节点和非叶节点的混合),键合并移动操作可合并节点的kvset序列以产生第一经合并kvset和第二经合并kvset。对于一些此类实施例,第一经合并kvset包括参考kvset序列的一组现有值块的第一组新键块,其中第一组新键块内的键映射(例如,基于确定性映射)到一或多个非叶节点,且第二经合并kvset包括参考一组新值块的第二组新键块,其中第二组新键块内的键映射(例如,基于确定性映射)到一或多个叶节点。如在本文中所提及,用于第二经合并kvset的一组新值块可被分配(例如,承袭)最大的(例如,最大值)分配给所述一组现有值块中的任何值块的数据生成编号。最终,第一经合并kvset可分离成第一组分离kvset且分布到一或多个

非叶节点,第二经合并kvset可分离成第二组分离kvset且分布到一或多个叶节点。

[0033] 在已生成所述一组新键块和所述一组新值块之后,键合并移动操作可从节点删除kvset序列中的每个特定kvset且删除每个特定kvset的所有键块。另外,在已生成所述一组新键块和所述一组新值块之后,kvset序列的一或多个现有值块(例如,所有现有值块)可基于其个别数据生成编号而被删除。如在本文中所提及,值块的个别数据生成编号可在初始创建时(例如,在添加到KVS树的根节点时)分配给块,或可能从KVS树的另一操作承袭。根据一些实施例,基于数据生成编号删除(例如,特定kvset的现有键块中的条目所参考的)现有值块可包括:通过针对键值存储树数据结构的每个特定叶节点确定分配给与所述特定叶节点相关联(例如,其kvset所包含)的任何值块的最大数据生成编号来确定一组最大数据生成编号;确定所述一组最大数据生成编号中的最小数据生成编号;以及如果值块具有小于所述最小数据生成编号的特定数据生成编号,则删除所述值块。前文操作可在本文中统称为键值压缩(kv压缩),其可视为用以移除过时的键块和值块以及被那些过时的键块和值块占用的空闲资源(例如,数据存储媒体的媒体块)的无用单元收集形式。

[0034] 在kvset序列的合并之后,键合并移动操作可将经合并kvset划分成一组分离kvset,其中每个分离kvset旨在用于一组子节点中的不同子节点。举例来说,可基于经合并kvset的键块到节点的一或多个子节点的决定性映射来划分经合并kvset。因此,可划分经合并kvset,使得针对将从经合并kvset接收(例如,添加)包括一或多个键块的kvset的每个子节点生成分离kvset,以及使得分配给(例如,将添加到)给定子节点的分离kvset将仅包括基于决定性映射(例如,本文所描述的映射值)映射到给定子节点的键块。如此,取决于经合并kvset所包含的键块,划分经合并kvset可产生一组分离kvset,基于键块的决定性映射,所述一组分离kvset包括用于节点的不到全部子节点的分离kvset(例如,一个子节点仅有一个分离kvset)。最终,键合并移动操作可将分离kvset移动到节点的一或多个子节点(例如,根据决定性映射)。将特定分离kvset移动到特定子节点可包括将分离kvset作为特定子节点的新kvset添加到所述特定子节点。根据其中经合并kvset的所有键块映射到单个子节点的一些实施例,可跳过划分步骤,且可在无需划分的情况下将经合并kvset移动(例如,添加)到所述单个子节点。

[0035] 实施例可在回收被与KVS树的叶节点相关联的一或多个过时的值块占用的存储器空间(例如,数据存储媒体空间)的操作(例如,无用单元收集操作)期间进行数据生成编号到一组新值块的分配,所述一组新值块针对经合并kvset而生成。举例来说,在无用单元收集过程期间,与KVS树的内部节点相关联的任何值块可在其相关联数据生成编号小于从KVS树的每个叶节点勘测到的一组最大(例如,最大值)数据生成编号中发现的最小(例如,最小值)数据退化编号的情况下被删除。因此,值块的删除可不限于键合并移动操作将kvset合并并且移动到叶节点时。

[0036] 根据一些实施例,(通过键合并移动操作)移动到一或多个子节点的分离kvset以原子方式替换且在逻辑上等同于从中生成所述分离kvset的经合并kvset。键合并移动操作可使用决定性技术来将分离kvset分布到含有经合并kvset的给定节点的一或多个子节点。键合并移动操作可使用任何此类键分布方法,使得对于给定节点和给定键K,键合并移动操作始终将具有键K的键值对(或铭牌)写入到所述节点的同子节点。实例分布方法可包含基于基数的键分布方法。

[0037] 对于一些实施例,键合并移动操作对键进行处理且产生对应于KVS树的决定性映射的映射值(例如,移动或溢出值)。对于一些实施例,键合并移动操作对键和当前树层级两者进行处理,且产生特定于在所述当前树层级处用于所述键的父节点或子节点的映射值。决定性映射可确保在给定键时,操作(例如,搜索或移动压缩操作)可知晓键值对将映射到哪个子节点而无需考虑KVS树的内容。

[0038] 本文公开执行与本文所描述的键合并移动相关的操作的系统的一些实例。

[0039] 图1说明根据本公开的一些实例的包含存储器子系统110的实例计算环境100。存储器子系统110可包含媒体,例如存储器组件112A到112N。存储器组件112A到112N可为易失性存储器装置、非易失性存储器装置或此类装置的组合。在一些实施例中,存储器子系统110是存储系统。存储系统的实例是SSD。在一些实施例中,存储器子系统110是混合式存储器/存储系统。通常,计算环境100可包含使用存储器子系统110的主机系统120。举例来说,主机系统120可将数据写入到存储器子系统110且从存储器子系统110读取数据。

[0040] 主机系统120可以是计算装置,例如台式计算机、笔记本电脑、网络服务器、移动装置,或包含存储器和处理装置的此类计算装置。主机系统120可包含或耦合到存储器子系统110,使得主机系统120可从存储器子系统110读取数据或将数据写入到所述存储器子系统。主机系统120可经由物理主机接口耦合到存储器子系统110。如本文所使用,“耦合到”通常是指组件之间的连接,其可以是间接通信连接或直接通信连接(例如不具有居间组件),无论有线或无线,包含例如电连接、光学连接、磁连接等连接。物理主机接口的实例包括但不限于串行高级技术附件(SATA)接口、外围组件互连高速(PCIe)接口、通用串行总线(USB)接口、光纤通道接口、串行连接的SCSI(SAS)等。物理主机接口可用于在主机系统120与存储器子系统110之间传输数据。当存储器子系统110通过PCIe接口与主机系统120耦合时,主机系统120还可利用NVM高速(NVMe)接口来存取存储器组件112A到112N。物理主机接口可提供用于在存储器子系统110与主机系统120之间传送控制、地址、数据以及其它信号的接口。

[0041] 存储器组件112A到112N可包含不同类型的非易失性存储器组件和/或易失性存储器组件的任何组合。非易失性存储器组件的实例包含“与非”(NAND)型快闪存储器。存储器组件112A到112N中的每一者可包含一或多个存储器单元阵列,例如单层级单元(SLC)或多层级单元(MLC)(例如,TLC或QLC)。在一些实施例中,特定存储器组件可包含存储器单元的SLC部分和MLC部分两者。存储器单元中的每一者可存储由主机系统120使用的一或多位的数据(例如,数据块)。尽管描述如NAND型快闪存储器的非易失性存储器组件,但存储器组件112A到112N可基于任何其它类型的存储器,例如易失性存储器。在一些实施例中,存储器组件112A到112N可以是但不限于随机存取存储器(RAM)、只读存储器(ROM)、动态随机存取存储器(DRAM)、同步动态随机存取存储器(SDRAM)、相变存储器(PCM)、磁阻随机存取存储器(MRAM)、或非(NOR)快闪存储器、电可擦除可编程只读存储器(EEPROM)以及非易失性存储器单元的交叉点阵列。非易失性存储器的交叉点阵列可结合可堆叠交叉网格化数据存取阵列基于体电阻的改变来执行位存储。另外,与许多基于快闪的存储器相比,交叉点非易失性存储器可执行就地写入操作,其中可在不预先擦除非易失性存储器单元的情况下对非易失性存储器单元进行编程。此外,存储器组件112A到112N的存储器单元可分组为存储器页或数据块,所述存储器页或数据块可指用于存储数据的存储器组件的单元。

[0042] 存储器子系统控制器115可与存储器组件112A到112N通信以执行操作,例如在存

存储器组件112A到112N处读取数据、写入数据或擦除数据,以及其它此类操作。存储器子系统控制器115可包含硬件,例如一或多个集成电路和/或离散组件、缓冲存储器,或其组合。存储器子系统控制器115可以是微控制器、专用逻辑电路系统(例如,现场可编程门阵列(FPGA)、专用集成电路(ASIC)等)或另一合适的处理器。存储器子系统控制器115可包含处理器(处理装置)117,其经配置以执行存储在本地存储器119中的指令。在所说明的实例中,存储器子系统控制器115的本地存储器119包含嵌入式存储器,其经配置以存储用于执行控制存储器子系统110的操作的各种过程、操作、逻辑流和例程的指令,包含处置存储器子系统110与主机系统120之间的通信。在一些实施例中,本地存储器119可包含存储存储器指针、提取的数据等的存储器寄存器。本地存储器119还可包含用于存储微码的只读存储器(ROM)。尽管将图1中的实例存储器子系统110说明为包含存储器子系统控制器115,但在本公开的另一实施例中,存储器子系统110可能不包含存储器子系统控制器115,且可能改为依靠(例如由外部主机或由与存储器子系统110分开的处理器或控制器提供的)外部控制。

[0043] 一般来说,存储器子系统控制器115可从主机系统120接收命令或操作,且可将命令或操作转换成指令或合适的命令,以实现存储对存储器组件112A到112N的所要存取。存储器子系统控制器115可负责其它操作,例如耗损均衡操作、无用单元收集操作、检错和纠错码(ECC)操作、加密操作、高速缓存操作,以及与存储器组件112A到112N相关联的逻辑块地址与物理块地址之间的地址转译。存储器子系统控制器115还可包含主机接口电路系统以经由物理主机接口与主机系统120通信。主机接口电路系统可将主机系统120接收到的命令转换成命令指令以存取存储器组件112A到112N,以及将与存储器组件112A到112N相关联的响应转换成用于主机系统120的信息。

[0044] 存储器子系统110还可包含未说明的额外电路系统或组件。在一些实施例中,存储器子系统110可包含高速缓存或缓冲器(例如DRAM)以及地址电路系统(例如行解码器和列解码器),所述地址电路系统可从存储器子系统控制器115接收地址且对地址进行解码以存取存储器组件112A到112N。

[0045] 存储器组件112A到112N中的任一者可包含媒体控制器(例如,媒体控制器113A和媒体控制器113N),以管理存储器组件的存储器单元、与存储器子系统控制器115通信以及执行从存储器子系统控制器115接收的存储器请求(例如,读取或写入)。

[0046] 主机系统120包含用于键值存储(KVS)树的键合并移动组件122,其可用于执行如本文关于KVS树所描述的键合并移动操作,所述KVS树存储在由存储器组件112A到112N中的一或多个者实施的数据存储媒体(例如,媒体池)上。存储在数据存储媒体上的KVS树可以由主机系统120、由存储器子系统110(例如,由存储器子系统控制器115在主机系统120的请求下)或其某一组合生成的KVS树。取决于实施例,键合并移动组件122可为主机系统120上的应用程序或操作系统(例如,用于存储器子系统110的操作系统软件驱动器)的部分。在一些实施例中,存储器子系统110包含键合并移动组件122的至少一部分。举例来说,存储器子系统控制器115可包含处理器117(例如,处理装置),其经配置以执行存储在本地存储器119中的指令以用于执行本文所描述的操作。如在本文中所提及,键合并移动操作可通过合并kvset的键块同时推迟kvset的值块的重写而合并KVS树的给定节点内的kvset,且遍历KVS树以将所得经合并kvset的部分放置到KVS树的一或多个子节点中。

[0047] 响应于检测到条件(例如,触发条件),这可包含与符合指定或计算的准则的给定

节点中的一或多个kvset相关的条件,键合并移动组件122可执行键合并移动操作。此类kvset相关准则的实例包括但不限于:给定节点内kvset的数目;给定节点添加(例如获取)新的kvset(例如由于将kvset从给定节点的父节点移动到给定节点);释放关于给定节点的资源(例如媒体块);给定节点内一或多个kvset的总大小;或可供用于无用单元收集的一或多个kvset中的数据量。其它条件实例包括但不限于(例如,从主机系统的软件应用程序或操作系统)接收请求以发起关于KVS树的一或多个节点的键合并移动操作,其中所述请求还可指定是对整个kvset序列还是对kvset子序列进行操作。

[0048] 取决于实施例,键合并移动组件122可包括使存储器子系统110(例如,存储器子系统控制器115)执行本文关于键合并移动组件122所描述的操作的逻辑(例如,一组机器指令,例如固件)或一或多个组件。键合并移动组件122可包括能够执行本文所描述的操作的有形单元。下文描述关于键合并移动组件122的操作的其它细节。

[0049] 图2是根据本公开的一些实施方案的用于KVS树200的实例键合并移动组件(下文称为键合并移动组件200)的框图。如所说明,键合并移动组件200包括基于键的键值组合并器210、键值组分离器220和键值组移动器230。对于一些实施例,键合并移动组件200可在组件或布置上与图2中所说明的有所不同(例如,更少或更多组件)。

[0050] 如在本文中所提及,由键合并移动组件200进行操作的KVS树可存储在存储器子系统(例如,110)上以促进数据(例如,用户数据)作为键值对的存储。所述KVS树可为形成键值存储(KVS)数据库的多个KVS树的部分,所述KVS数据库可包括多级树,所述多级树具有包括不均匀kvset的基础级和包括各自分别包括均匀kvset的两个或更多个KVS子树的根节点(且因此起点)的第二级。KVS树可由主机系统(例如,110)、存储器子系统(例如,110)或其某一组合生成。KVS树可在一组存储器组件上生成,使得KVS树包括一组节点,其中所述一组节点中的节点包括kvset序列,且其中kvset序列中的kvset包括用于存储一或多个键的一组键块和用于存储一或多个值的一组值块。kvset序列可按时间排序,使得相对于最近添加的kvset,序列中较早添加的kvset表示较旧数据(例如,键值对)。

[0051] 基于给定节点的一组子节点是否包括叶节点,基于键的键值组合并器210合并(例如,压缩)给定节点的kvset序列以生成经合并kvset。因此,在键合并移动组件200相对于给定节点发起键合并移动(例如,由键合并移动组件200基于将kvset序列合并(例如,压缩)且从给定节点移动(例如,溢出)到其子节点中的一者的条件的满足而触发)后,基于键的键值组合并器210(或在键合并移动组件200内部或外部的某一其它组件)可确定给定节点的一组子节点是否包括叶节点。

[0052] 响应于确定所述一组子节点不包括叶节点(例如,所述组中没有子节点是叶节点),基于键的键值组合并器210可合并节点的kvset序列以产生经合并kvset,其中所得经合并kvset包括参考kvset序列的一组现有值块(例如,先前生成的值块)的一组新键块。举例来说,如在本文中所提及,可通过将所述一组现有键块的值(例如,键值和对现有值块的参考)复制到所述一组新键块来生成所述一组新键块。所述kvset序列中未被所述一组新键值块参考的那些现有值块被保留(例如,未删除),但被视为未被参考且可由不同kvset(例如,KVS树的内部节点的两个kvset)的两个键块共享。在已生成所述一组新键块之后,键合并移动操作可从节点删除键值组序列中的每个特定键值组,且删除每个特定键值组的所有键块,同时保留每个特定键值组的所有值块(例如,原样保留)。保留的值块可包括所述一组

新键块所参考的值块、未被所述一组新键块中的任何一者参考的值块,或这两者。kvset序列的(被原样保留的)所有值块可移动到经合并kvset。

[0053] 替代地,响应于确定所述一组子节点仅包括一或多个叶节点,基于键的键值组合并器210可合并给定节点的kvset序列以产生经合并kvset,使得所得经合并kvset包括参考一组新值块的一组新键块,其中所述新键块基于所述kvset序列的一组现有键块而生成,且其中所述一组新值块基于所述kvset序列的一组现有值块而生成。举例来说,如在本文中所提及,可基于kvset序列的所述一组现有键块,通过将所述一组现有键块的值(例如,键值)复制到所述一组新键块且使(所述一组新键块中的)新键块分别参考对应于所述一组现有键块所参考的现有块的(所述一组新值块中的)新值块来生成所述一组新键块。所述一组新值块可被分配(例如,承袭)分配给所述一组现有值块中的任何值块的最大(例如,最大值)数据生成编号。

[0054] 另外,响应于确定所述一组子节点包括至少一个叶节点和至少一个非叶节点(例如,KVS树不平衡且因此所述一组子节点可包括叶节点和非叶节点的混合),基于键的键值组合并器210可合并节点的kvset序列以产生第一经合并kvset和第二经合并kvset。对于一些此类实施例,第一经合并kvset包括参考kvset序列的一组现有值块的第一组新键块,其中第一组新键块内的键映射(例如,基于确定性映射)到一或多个非叶节点,且第二经合并kvset包括参考一组新值块的第二组新键块,其中第二组新键块内的键映射(例如,基于确定性映射)到一或多个叶节点。如在本文中所提及,用于第二经合并kvset的一组新值块可被分配(例如,承袭)最大的(例如,最大值)分配给所述一组现有值块中的任何值块的数据生成编号。

[0055] 在已生成所述一组新键块和所述一组新值块之后,对于一些实施例,基于键的键值组合并器210从给定节点删除kvset序列中的每个特定kvset,这包括删除特定kvset的所有键块(例如,所有现有值块)且还包括基于值块的个别数据生成编号而删除值块(例如,特定kvset的现有键块中的条目所参考的一或多个现有值块)。关于删除值块,基于键的键值组合并器210可基于值块的个别数据生成编号而通过以下操作来删除值块:通过针对键值存储树数据结构的每个特定叶节点确定分配给与所述特定叶节点相关联(例如,其kvset所包含)的任何值块的最大数据生成编号来确定一组最大数据生成编号;确定所述一组最大数据生成编号中的最小数据生成编号;以及如果值块具有小于所述最小数据生成编号的特定数据生成编号,则删除所述值块。取决于实施例,基于键的键值组合并器210可在删除给定kvset序列时确定所述一组最大数据生成编号仅一次。

[0056] 键值组分离器220可将(由基于键的键值组合并器210生成的)经合并kvset划分成一组分离kvset,其中每个分离kvset旨在用于一组子节点中的不同子节点。如在本文中所提及,可基于经合并kvset的键块(例如,键块的基于基数的键分布)到节点的一或多个子节点的决定性映射来划分经合并kvset。因此,键值组分离器220可划分经合并kvset,使得针对将从经合并kvset接收(例如,添加)包括一或多个键块的kvset的每个子节点生成分离kvset,以及使得分配给(例如,将添加到)给定子节点的分离kvset将仅包括基于决定性映射而映射到给定子节点的键块。举例来说,键值组分离器220可处理来自经合并kvset的特定键块的键和给定节点的当前树层级,且产生特定于在当前树层级处用于所述键的(给定节点的)子节点中的一者的映射值(例如,移动或溢出值)。因此,使用映射值,键值组分离器

220可确定哪个分离kvset(各自对应于不同子节点)从经合并kvset接收哪些键块。对于其中基于键的键值组合并器210生成旨在用于非叶节点的第一经合并kvset和旨在用于叶节点的第二经合并kvset的一些实施例,键值组分离器220将第一经合并kvset划分成第一组分离kvset且将第二经合并kvset划分成第二组分离kvsets,其中每个分离kvset旨在用于一组子节点中的不同子节点。

[0057] 根据键块的决定性映射,键值组移动器230将(由键值组分离器220生成的)一组分离kvset中的每个分离kvset移动(例如,溢出)到给定节点的其对应的子节点。这可类似于如上文所描述的当键值组分离器220生成第一组分离kvset和第二组分离kvset时。对于一些实施例,将给定分离kvset从所述一组分离kvset移动到其对应的子节点包括将给定分离kvset作为新kvset添加到对应的子节点(例如,到子节点的kvset序列)。

[0058] 如在本文中提及,对于其中经合并kvset的所有键块映射到节点的一个子节点的一些实施例,跳过由键值组分离器220对经合并kvset的划分,且键值组移动器230基于KVS树的决定性映射将经合并kvset移动到所述一个子节点。

[0059] 图3到5是根据本公开的一些实施方案的用于键合并移动的实例方法的流程图。图3到5的方法300、400、500中的任一者可由处理逻辑执行,所述处理逻辑可包含硬件(例如,处理装置、电路系统、专用逻辑、可编程逻辑、微码、装置的硬件、集成电路等)、软件(例如,在处理装置上运行或执行的指令),或其组合。在一些实施例中,图3到5的一或多种方法300、400、500由图1的主机系统120执行。在这些实施例中,方法300、400、500可至少部分地由键合并移动组件122执行。替代地,图3到5的一或多个方法由图1的存储器子系统110(例如,存储器子系统控制器115的处理器)执行。尽管以特定顺序或次序来展示过程,但除非另有指定,否则可修改所述过程的次序。因此,所说明实施例应仅作为实例理解,且所说明过程可以不同次序执行,且一些过程可并行地执行。另外,在各种实施例中可省去一或多个过程。因此,并非在每个实施例中都需要所有过程。其它过程流是可能的。图3到5的方法300、400、500的操作可相对于KVS树的两个或更多个节点同时执行。

[0060] 现参考图3的方法300,在操作305处,主机系统(例如,120)的处理装置在存储器子系统(例如,110)上生成包括一组节点的键值存储树数据结构(KVS树),其中每个节点包括键值组(kvset)序列,且kvset序列中的kvset包括用于存储一或多个键的一组键块和用于存储一或多个值的一组值块。替代地,可已经(例如,通过另一过程)生成KVS树,且对于一些实施例,在操作305处存取KVS树,其中存取KVS树可辅助方法300的后续操作(例如,操作310)。

[0061] 在操作310处,主机系统的处理装置检测条件(例如,触发条件)以将kvset序列合并并且从给定节点移动到给定节点的一组子节点。如在本文中所提及,实例条件可包含但不限于涉及符合指定的或计算的kvset相关准则的给定节点的条件,例如给定节点内kvset的数目、给定节点添加(例如获取)新的kvset、释放关于给定节点的资源(例如媒体块)、给定节点内一或多个kvset的总大小,或可供用于无用单元收集的一或多个kvset中的数据量。在操作315处,如果主机系统的处理装置(在操作310处)检测到将kvset序列合并并且从给定节点移动到给定节点的一组子节点的条件,则方法300进行到操作320;否则方法300返回到操作310。在操作320处,主机系统的处理装置确定给定节点的一组子节点是否包括叶节点。在一些实施例中,KVS树经结构化以使得如果给定节点的一个子节点包括叶节点,则给定节

点的所有子节点包括叶节点。

[0062] 在操作325处,主机系统的处理装置基于操作320对所述一组子节点是否包括叶节点的确定而将kvset序列移动到所述一组子节点。对于一些实施例,基于确定所述一组子节点是否包括叶节点而将kvset序列移动到所述一组子节点包括:主机系统的处理装置响应于(在操作320处)确定所述一组子节点不包括叶而执行图4的方法400,且主机系统的处理装置响应于(在操作320处)确定所述一组子节点确实包括叶而执行图5的方法500。

[0063] 现参考图4的方法400,在操作405处,主机系统的处理装置合并kvset序列以产生经合并kvset,所述经合并kvset包括参考kvset序列的一组现有值块的一组新键块,其中所述一组新键块基于kvset序列的一组现有键块而生成。如在本文中所提及,此组新键块内的键可映射(例如,基于确定性映射)到给定节点的一或多个非叶节点。对于一些实施例,基于kvset序列的所述一组现有键块,通过将所述一组现有键块的值(例如,键值和对现有值块的参考)复制到所述一组新值块来生成所述一组新键块,由此使所述一组新键块包括与所述一组现有键块相同的对所述一组现有值块的参考。另外,对于一些实施例,将kvset序列的所有现有值块(其在操作410处原样保留)移动到经合并kvset。

[0064] 在操作410处,主机系统的处理装置通过删除特定kvset的所有键块同时原样保留所述特定kvset的所有值块(例如,现有值块)而从给定节点删除kvset序列中的每个特定kvset。如在本文中所提及,特定kvset的值块可包含所述一组新键块所参考的值块、未被所述一组新键块参考的值块,或这两者的组合。

[0065] 在操作415处,主机系统的处理装置将(在操作405处生成的)经合并kvset移动(例如,溢出)到给定节点的所述一组子节点中。对于一些实施例,将经合并kvset移动到所述一组子节点中包括首先将经合并kvset划分成一组分离kvset,其中每个分离kvset被分配到(所述一组子节点中的)不同子节点,所述子节点将基于KVS树的决定性映射(例如,本文中所描述的映射值)接收经合并kvset的一部分。如在本文中所提及,取决于经合并kvset所包含的键块,划分经合并kvset可产生一组分离kvset,基于键块的决定性映射,所述一组分离kvset包括用于节点的不到全部子节点的分离kvset(例如,一个子节点仅有一个分离kvset)。最终,主机系统的处理装置将所述一组分离kvset中的每个分离kvset移动到所述一组子节点中的所分配子节点。对于一些实施例,将特定分离kvset移动到特定子节点可包括将分离kvset作为特定子节点的新kvset添加到所述特定子节点。根据其中经合并kvset的所有键块映射到单个子节点的一些实施例,可跳过划分操作,且可在无需划分的情况下将经合并kvset移动(例如,添加)到所述单个子节点。

[0066] 现参考图5的方法500,在操作505处,主机系统的处理装置合并kvset序列以产生经合并kvset,所述经合并kvset包括参考一组新值块的一组新键块,其中所述一组新键块基于kvset序列的一组现有键块而生成,且所述一组新值块基于kvset序列的一组现有值块而生成。如在本文中所提及,此组新键块内的键可映射(例如,基于确定性映射)到给定节点的一或多个叶节点。对于一些实施例,基于kvset序列的所述一组现有值块,通过将所述一组现有值块的值复制到所述一组新值块来生成所述一组新值块。另外,对于一些实施例,所述一组新值块被分配最大的分配给所述一组现有值块中的任何值块的数据生成编号。关于所述一组新键块,根据一些实施例,基于kvset序列的所述一组现有键块,通过将所述一组现有键块的值(例如,键值)复制到所述一组新键块且使(所述一组新键块中的)新键块分别

参考对应于所述一组现有键块所参考的现有块的(所述一组新值块中的)新值块来生成所述一组新键块。以此方式,所述一组新键块可包括与所述一组现有键块相同的数据内容(例如,键),同时参考对应于所述一组现有键块的新值块。

[0067] 在操作510处,主机系统的处理装置通过针对KVS树的每个特定叶节点确定分配给与所述特定叶节点相关联的任何值块的最大数据生成编号来确定一组最大数据生成编号(对于KVS树的叶节点)。在操作515处,主机系统的处理装置确定在操作510处所确定的所述一组最大数据生成编号中的最小数据生成编号。

[0068] 在操作520处,主机系统的处理装置通过删除kvset序列的现有键块且通过删除kvset序列的具有小于在操作515处所确定的最小数据生成编号的数据生成编号的现有值块而从给定节点删除kvset序列。这样,kvset序列中具有不小于(在操作515处确定的)最小数据生成编号的数据生成编号的那些现有值块可原样留下,且可表示仍由仍存在于KVS树内的至少一个kvset参考(例如,与其共享)的现有值块。可最终在KVS树的未来操作(例如,键合并移动的未来执行)中删除原样留下的那些现有值块。对于一些实施例,作为删除由kvset序列包含的每个kvset的操作(例如,在操作515处执行)的部分,执行kvset序列的现有键块和现有值块的删除。

[0069] 在操作525处,主机系统的处理装置将(在操作505处生成的)经合并kvset移动(例如,溢出)到给定节点的所述一组子节点中。对于各种实施例,操作525类似于上文关于图4的方法400所描述的操作415。

[0070] 图6是说明根据本公开的一些实施方案的可通过键合并移动进行操作的实例KVS树600的框图。如在本文中所提及,KVS树600包括被组织为树的键值数据结构。作为键值数据结构,值连同参考所述值的对应键一起存储在KVS树600中。具体地说,键条目可用于含有键和额外信息,例如对值的参考。键自身可具有KVS树600内的总排序,且因此,键可在彼此当中分类。键还可分成子键,其中所述子键是键的不重叠部分。对于一些实施例,键的总排序基于在多个键之间比较类似子键(例如,将键的第一子键与另一键的第一子键相比较)。另外,对于一些实施例,键前缀包括键的开始部分。当使用时,键前缀可由一或多个子键构成。

[0071] KVS树600包括一或多个节点,例如节点610,其中每一者包含一或多个键值组(kvset)。对于一些实施例,一或多个节点(例如,节点610)各自包括按时间排序的kvset序列。如所说明,kvset 615包括‘N’标记以指示其为序列的最新组,而kvset 620包括‘0’标记以指示其为序列的最旧组。Kvset 625包括‘I’标记以指示其在序列的中间。这些标记始终用以标记kvset;然而,另一标记(例如‘X’)表示特定kvset而非其在序列中的位置(例如,新、中间、旧等),除非其为波浪符‘~’,在此情况下其仅为不记名kvset。如下文更详细地解释,较旧kvset(具有较旧键块)在KVS树600中较低处出现。因此,将kvset沿树层级向下推动(例如,溢出),例如从L1到L2,会使来自父节点的至少一个新kvset被添加到所述父节点接收方子节点中的最新位置。

[0072] KVS树600包括给定节点(例如,节点610)的kvset中(由键块和值块存储)的键值对到给定节点的任何一个子节点(例如,表示为L1处的所有节点的节点610中的任一个子节点)的决定性映射。KVS树600的决定性映射可意指给定某个键,外部实体可在不知道KVS树600的内容的情况下寻踪通过KVS树600的节点的路径到所述键的(键值对的)键块和值块。

举例来说,这不同于B树,例如在B树中,树的内容将决定给定键的值将落在何处,以便维持树的搜索优化结构。相比之下,KVS树600的决定性映射可提供规则以使得例如给定某个键,可计算键将映射到的L3处的子节点,即使最大树层级(例如,树深度)当时仅处于L1处也如此。对于一些实施例,决定性映射包括所述键的一部分的散列的一部分。子键可经散列以到达映射集,且映射集的一部分可用于树的任何给定层级。取决于实施例,键的所述部分可包括整个键。

[0073] 对于一些实施例,散列包括包含所述散列的所述部分的多个不重叠部分。例如,多个不重叠部分中的每一者可对应于树的一级。可通过节点的层级从所述多个不重叠部分确定散列的部分。因此,节点的子节点的最大数目可由散列的所述部分的大小限定,其中散列的所述部分的所述大小可以是一定数目的位。举例来说,关于产生八个位的键的散列,所述八个位可划分成三个组,包括前两个位、位三到六以及位七和八。子节点可基于此组位编索引,使得处于第一级(例如,L1)的子节点具有两位名(基于位一和二),第二级(例如,L2)上的子节点具有四位名(基于位三到六),且第三级(例如,L3)上的子节点具有两位名(基于位七和八)。

[0074] 对于一些实施例,节点610表示KVS树600的根节点。KVS树600可存储在由存储器子系统(例如,110)实施的数据存储媒体上,其中KVS树600可存储在数据存储媒体的媒体块中。数据存储媒体的媒体块可以是可寻址的块。

[0075] 图7A和7B是说明根据本公开的一些实施方案的在一组子节点不包括叶节点时对一个或多个键值组执行的实例键合并移动的框图。具体地说,图7A相对于父节点700说明kvset序列的一或多个现有键块的读取和合并(同时未读取kvset序列的现有值块)、将所得经合并键块作为一或多个新键块写入到新kvset(经合并kvset)中、删除现有键块以及删除kvset序列中的kvset,同时原样保留现有值块。图7B继续树遍历到父节点700的一或多个子节点以将经合并(例如,经压缩)kvset的一或多个部分移动(例如,溢出)到所述一或多个子节点中。对于一些实施例,图7A和7B的操作说明后跟着实例溢出操作的实例键压缩操作。

[0076] 现参考图7A,父节点700包括KVSET 1、KVSET 2和KVSET 3,其可表示kvset序列(例如,随着时间推移,向父节点700首先添加KVSET 1,其次添加KVSET 2,且最后添加KVSET 3)。取决于实施例,KVSET 1、2、3可表示父节点700的所有KVSET,或可仅表示父节点700内较大kvset序列的尾端。所说明的KVSET包括键块705、710、725、730、745、755、760、765和值块715、720、735、740、750。键块705包括键A且参考值ID 10;键块710包括键B和对值ID 11的值参考;键块725包括键B和对值ID 20的值参考;键块730包括键C和对值ID 21的值参考;键块745包括键A和对值ID 30的值参考;键块755包括键A和对值ID 10的值参考;键块760包括键B和对值ID 11的值参考;且键块765包括键C和对值ID 21的值参考。虽然在图7A和7B中将键块(例如,705、710、725、730、745、755、760、765)说明为包括单个键和对值ID的相关联参考,但对于一些实施例,给定键块可包括两个或更多个键和其值ID的相关联参考。例如,新键块755、760、765的数据内容可含于KVSET 3的单个新键块内。

[0077] 值块715包括值ID (VID) 10和数据生成编号 (DGEN) 5;值块720包括VID 11和DGEN 6;值块735包括VID 20和DGEN 2;值块740包括VID 21和DGEN 3;且值块750包括VID 30和DGEN 1。键块705、710、725、730、745表示kvset序列的现有键块,值块715、720、735、740、750表示kvset序列的现有值块,且键块755、760、765表示针对新KVSET 3生成的新键块。

[0078] 对于一些实施例,由合并kvset序列产生的经合并kvset承袭与kvset序列中的任一kvset相关联的最大序数。举例来说,对于一些实施例,KVS树中的个别kvset(如值块)可与数据生成编号相关联,且所述序数可包括与kvset相关联的相关数据生成编号。因此,在kvset序列包括KVSET 10、KVSET 8、KVSET 7和KVSET 5的情况下,由合并kvset序列产生的经合并kvset可被视为KVSET 10,因为KVSET 10具有序列中的最大序数。另外,以此方式,经合并kvset可被视为对kvset序列的替换。

[0079] 如所说明,将KVSET 3(最新)、KVSET 2和KVSET 1(最旧)合并(例如,压缩)成父节点700内的新KVSET 3。在合并期间,新KVSET 3生成为添加到父节点700的新kvset,且基于现有键块705、710、725、730、745的合并针对新KVSET 3生成新键块755、760、765。现有键块705、710、725、730、745的合并导致在键A上(相对于现有键块705和745)的冲突和在键B上(相对于现有键块710和725)的冲突。根据一些实施例,给定kvset针对每个键仅包括一个键条目(跨给定kvset的键块),且经合并的多个键块之间的键冲突可以有利于多个键块中的最近(例如,最新)键条目(例如,保留最近键条目且舍弃多个键块中的键条目的其余部分)的方式解决。因此,键A和B的冲突以有利于键A和B的最近项的方式解决,所述最近项(在图7A中)存储在键A和B的最左边键块中——用于键A的现有键块705和用于键B的现有键块710。键C不具有冲突,且因此用于键C的现有键块730将相对于新的新KVSET 3使用。因此,合并现有键块705、710、725、730、745会使存储在现有键块705、710、730中的键A、B和C的键条目被选择用于合并,且基于这些选择的键条目生成用于新KVSET 3的新键块755、760、765。最终删除现有键块705、710、725、730、745,kvset序列中的每一个kvset(即,KVSET 1、2、3)也如此。在删除KVSET 1、2、3的情况下,新KVSET 3替换父节点700内的KVSET 1、2、3。

[0080] 因此,基于所选择的现有键块705、710和730,通过将值(例如,键值和来自现有值块中的条目的参考)从现有键块705、710、730复制到新键块755、760、765来生成新键块755、760、765。因此,与现有键块705、710、730相同,新键块755、760、765分别参考VID 10、11和21(分别对应于现有值块715、720、740)。以此方式,新键块755、760、765继续参考kvset序列(KVSET 1、2、3)的现有键块705、710、730,且需要为新KVSET 3创建新值块。还如所展示,现有键块705、710、730被移动到新KVSET 3。在删除现有键块705、710、725、730、745之后,kvset序列中未被新KVSET 3的新值块参考的那些现有值块(735、750)继续存留,但未被移动到新KVSET 3且被视为未被参考,这在图7A和7B中利用虚线表示。如在本文中所提及,尽管未被新KVSET 3参考,但现有块735、750可继续存在且被KVS树内的一或多个其它kvset参考(且因此被共享)。

[0081] 对于一些实施例,生成和添加新KVSET 3包括在父节点700的kvset序列内相对于KVSET 3(例如,向左)将新KVSET 3添加到较新位置,由此确保新KVSET 3的生成和添加是KVS树内的非分块(non-blocking)操作。

[0082] 现参考图7B,在删除kvset序列(KVSET 1、2、3)、包含删除现有键块705、710、725、730、745之后,新KVSET 3仍在父节点700内,带有新键块755、760、765。将新KVSET 3划分(或分离)成包括KVSET X、Y、Z的一组新kvset(在本文中也称为分离kvset)。如在本文中所提及,对于一些实施例,KVS树使用决定性映射(例如,映射值)以用于将来自父节点(例如,700)的经合并kvset(例如,新KVSET 3)的键块的键条目和(键块所参考的)值块分布到所述父节点的一或多个子节点。因此,根据新键块755、760、765的键条目基于其相应映射值映射

哪些子节点来划分新KVSET 3。在图7B中,键块的给定键条目的映射值包括给定键条目的键的散列。如所说明,这会产生以下情况:键A的键条目(散列(键A)=00F)通过新键块770包含在新KVSET X中,所述新KVSET X经分配以移动到与映射值00F相关联(例如,映射到所述映射值)的非叶子节点;键C的键条目(散列(键C)=00F)通过新键块775包含在新KVSET Y中,所述新KVSET Y经分配以移动到与映射值11C相关联的非叶子节点;以及键B的键条目(散列(键B)=00F)通过新键块780包含在新KVSET Z中,所述新KVSET Z经分配以移动到与映射值0X2相关联的非叶子节点。

[0083] 如图7B中所展示,新KVSET 3的划分并未产生与映射值1X0相关联的非叶节点的新kvset,因为新KVSET 3没有一个键具有映射值1X0。当KVSET X、Y、Z移动到其相应非叶子节点时,根据一些实施例,所述KVSET作为最新kvset被添加到那些节点(例如,通过放置在节点内的最左边位置处来表示)。还如图7B中说明,现有值块735、750继续存留为未被参考的值块(如虚线所表示)。

[0084] 图8A和8B是说明根据本公开的一些实施方案的在一组子节点仅包括一或多个叶节点时对一或多个键值组执行的实例键合并移动的框图。具体地说,图8A相对于父节点800说明kvset序列的一或多个现有键块和一或多个现有值块的读取和合并、将所得经合并键块和值块作为一或多个新键块和一或多个新值块写入到新kvset(经合并kvset)中、删除现有键块、删除kvset序列中的kvset,以及基于现有值块的相应数据生成编号而删除所述现有值块。图8B继续树遍历到父节点800的一或多个子节点以将经合并(例如,经压缩)kvset的一或多个部分移动(例如,溢出)到所述一或多个子节点中。不同于图7A和7B,父节点800的子节点是叶子节点。如在本文中所提及,通过在KVS树的叶节点层级(而不是在非叶节点层级)生成新值块,一些实施例可避免或延迟在执行KVS树的内部节点之间的键合并移动时写入值块。对于一些实施例,图8A和8B的操作说明后跟着实例溢出操作的实例键值压缩操作。

[0085] 现参考图8A,父节点800包括KVSET 1、KVSET 2和KVSET 3,其可表示kvset序列(例如,随着时间推移,向父节点800首先添加KVSET 1,其次添加KVSET 2,且最后添加KVSET 3)。取决于实施例,KVSET 1、2、3可表示父节点800的所有KVSET,或可仅表示父节点800内较大kvset序列的尾端。所说明的KVSET包括键块805、810、825、830、845、855、860、865和值块815、820、835、840、850、870、875、880。键块805包括键A且参考值ID 10;键块810包括键B和对值ID 11的值参考;键块825包括键B和对值ID 20的值参考;键块830包括键C和对值ID 21的值参考;键块845包括键A和对值ID 30的值参考;键块855包括键A和对值ID 31的值参考;键块860包括键B和对值ID 32的值参考;且键块865包括键C和对值ID 33的值参考。虽然在图8A和8B中将键块(例如,805、810、825、830、845、855、860、865)说明为包括单个键和对值ID的相关联参考,但对于一些实施例,给定键块可包括两个或更多个键和其对值ID的相关联参考。例如,新键块855、860、865的数据内容可含于KVSET 3的单个新键块内。

[0086] 值块815包括值ID (VID) 10和数据生成编号 (DGEN) 5;值块820包括VID 11和DGEN 6;值块835包括VID 20和DGEN 2;值块840包括VID 21和DGEN 3;值块850包括VID 30和DGEN 1;值块870包括VID 31和DGEN 6;值块875包括VID 32和DGEN 6;且值块880包括VID 33和DGEN 6。

[0087] 键块805、810、825、830、845表示kvset序列的现有键块,值块815、820、835、840、850表示kvset序列的现有值块,键块855、860、865表示针对新KVSET 3生成的新键块,且值

块870、875、880表示针对新KVSET 3生成的新值块。

[0088] 如所说明,将KVSET 3(最新)、KVSET 2和KVSET 1(最旧)合并(例如,压缩)成父节点800内的新KVSET 3。在合并期间,新KVSET 3生成为添加到父节点800的新kvset,基于现有键块805、810、825、830、845的合并而生成用于新KVSET3的新键块855、860、865,且基于现有值块815、820、835、840、850的合并(鉴于现有键块805、810、825、830、845的合并)而生成用于新KVSET 3的新值块870、875、880。现有键块805、810、825、830、845的合并导致在键A上(相对于现有键块805和845)的冲突和在键B上(相对于现有键块810和825)的冲突。如在本文中提及,根据一些实施例,给定kvset针对每个键仅包括一个键条目(跨给定kvset的键块),且经合并的多个键块之间的键冲突可以有利于多个键块中的最近(例如,最新)键条目(例如,保留最近键条目且舍弃多个键块中的键条目的其余部分)的方式解决。因此,键A和B的冲突以有利于键A和B的最近键块的方式解决,所述最近键块(在图8A中)存储在键A和B的最左边键块中--用于键A的现有键块805和用于键B的现有键块810。键C不具有冲突,且因此用于键C的现有键块830将相对于新的新KVSET 3使用。因此,合并现有键块805、810、825、830、845会使存储在现有键块805、810、830中的键A、B和C的键条目被选择用于合并,且基于这些选择的键条目生成用于新KVSET 3的新键块855、860、865。另外,基于针对键A、B和C选择现有键块805、810、830,对应于现有键块805、810、830的现有值块(815、820、840)用于生成新值块870、875、880。最终删除现有键块805、810、825、830、845,kvset序列中的每一个kvset(即,KVSET 1、2、3)也如此。在删除KVSET1、2、3的情况下,新KVSET 3替换父节点800内的KVSET 1、2、3。

[0089] 如所展示,基于现有值块815、820、840,通过将现有值块815、820、840的值复制到对应新值块870、875、880中来生成新值块870、875、880。另外,新值块870、875、880中的每一者被分配(例如,承袭)现有值块815、820、835、840、850中的任一者的最大数据生成编号,所述最大数据生成编号在图8A的情况下为6。基于所选择的现有键块805、810、830生成新键块855、860、865,使得新键块855、860、865分别包括与现有键块805、810、830相同的键,同时参考对应于现有值块815、820、840的新值块870、875、880。因此,新键块855、860、865分别包括对VID 31、32、33的值参考。

[0090] 可基于其相应数据生成编号来删除现有值块815、820、835、840、850中的每一者。根据一些实施例,对于父节点800的每个叶子节点,确定所述叶子节点包含(例如,由其kvset参考)的任何值块的最大数据生成编号。在图8B中,这将产生一组最大数据生成编号{39,30,14,21}。接着,根据一些实施例,确定所述一组最大数据生成编号包含的最小数据生成编号。在图8B的情况下,这将是14。随后,根据一些实施例,删除具有小于所确定的最小数据生成编号的数据生成编号的任何现有键块。关于图8B,现有值块815、820、835、840、850中的每一者具有小于最小数据生成编号14的数据生成编号(分别为5、6、2、3、1)。

[0091] 对于一些实施例,生成和添加新KVSET 3包括在父节点800的kvset序列内相对于KVSET 3(例如,向左)将新KVSET 3添加到较新位置,由此确保新KVSET 3的生成和添加是KVS树内的非分块操作。

[0092] 现参考图8B,在删除kvset序列(KVSET 1、2、3)、包含删除现有键块805、810、825、830、845之后,新KVSET 3仍在父节点800内,带有分别参考新值块870、875、880的新键块855、860、865。将新KVSET 3划分(或分离)成包括KVSET X、Y、Z的一组新kvset(在本文中)

称为分离kvset)。如在本文中所提及,对于一些实施例,KVS树使用决定性映射(例如,映射值)以用于将来自父节点(例如,800)的经合并kvset(例如,新KVSET 3)的键块的键条目和(键块所参考的)值块分布到所述父节点的一或多个子节点。因此,根据新键块855、860、865的键条目基于其相应映射值映射哪些子节点来划分新KVSET 3。在图8B中,键块的给定键条目的映射值包括给定键条目的键的散列。如所说明,这会产生以下情况:键A的键条目(散列(键A)=00F)通过新键块870包含在新KVSET X中,所述新KVSET X经分配以移动到与映射值00F相关联(例如,映射到所述映射值)的非叶子节点;键C的键条目(散列(键C)=00F)通过新键块875包含在新KVSET Y中,所述新KVSET Y经分配以移动到与映射值11C相关联的非叶子节点;以及键B的键条目(散列(键B)=11C)通过新键块880包含在新KVSET Z中,所述新KVSET Z经分配以移动到与映射值0X2相关联的非叶子节点。

[0093] 如图8B中所展示,新KVSET 3的划分并未产生与映射值1X0相关联的叶节点的新kvset,因为新KVSET 3没有一个键块具有映射值1X0。当KVSET X、Y、Z移动到其相应非叶子节点时,根据一些实施例,将所述KVSET X、Y、Z作为最新kvset添加到那些节点(例如,这通过放置在节点内的最左位置处来表示)。还如图8B中所说明,现有值块835、850并不继续存留。

[0094] 图9A到9C提供说明在其中执行用于键合并移动的方法的实例实施例的上下文中的计算环境100的组件之间的交互的交互图。方法的操作可由处理逻辑执行,所述处理逻辑可包含硬件(例如,处理装置、电路系统、专用逻辑、可编程逻辑、微码、装置的硬件、集成电路等)、软件(例如,在处理装置上运行或执行的指令),或其组合。在一些实施例中,所述方法由主机系统120执行。尽管以特定顺序或次序来展示操作,但除非另有指定,否则可修改过程次序。因此,所说明实施例应仅作为实例理解,且所说明过程可以不同次序执行,且一些过程可并行地执行。另外,在各种实施例中可省去一或多个过程。因此,并非每个实施例中都需要所有过程。

[0095] 在图9A到9C中说明的实例的上下文中,主机系统可包括主机系统120,且存储器子系统可包括存储器子系统110,其中存储器组件112A到112N中的一或多个者可实施用于存储由主机系统120操作的KVS树的数据存储媒体。

[0096] 如所展示,在操作902处,主机系统120生成键值存储树数据结构(KVS树),主机系统120将其写入到存储器子系统110以供存储。作为响应,在操作910处,存储器子系统110将KVS树存储在数据存储媒体上。

[0097] 在操作904处,主机系统120检测将父节点的键值组序列合并且移动到所述父节点的一组子节点的条件。在主机系统120检测到所述条件后,在操作906处,主机系统120确定所述一组子节点是否包括叶节点。为实现这一点,主机系统120读取存储在存储器子系统110的数据存储媒体上的KVS树(例如,以遍历到父节点和所述一组子节点),且存储器子系统110在操作912处提供对KVS树的存取。如果主机系统120确定一组子节点仅包括叶节点,则主机系统120进行到操作924,而如果主机系统120确定一组子节点不包括叶节点,则主机系统120进行到操作908。

[0098] 虽然未说明,但主机系统120可确定一组子节点包括至少一个叶节点和至少一个非叶节点(例如,KVS树不平衡)。如果主机系统120确定一组子节点包括至少一个叶节点和至少一个非叶节点,如本文所描述,则主机系统120针对映射到非叶节点的键条目将所述键

值组序列合并成第一经合并键值组,且针对映射到叶节点的键条目将所述键值组序列合并成第二经合并键值组。随后,可通过操作920和此后跟随的操作来对第一经合并键值组进行操作,同时可通过操作926和此后跟随的操作来对第二经合并键值组进行操作。

[0099] 在操作908处,主机系统120合并父节点的键值组序列以产生经合并键值组,所述经合并键值组包括参考所述键值组序列所包含的(例如,其键值组所参考的)一组现有值块的一组新键块。如在本文中所提及,基于所述键值组序列所包含的(例如,其键值组所包含的)一组现有键块来生成所述一组新键块。为了合并父节点的键值组序列,主机系统120读取存储在存储器子系统110的数据存储媒体上的KVS树(例如,以读取所述序列的键值组),且存储器子系统110在操作914处提供对KVS树的存取。

[0100] 在操作908之后,在操作920处,主机系统120通过删除所述键值组序列的所有现有值块同时原样保留所述键值组序列的所有现有值块而删除所述键值组序列。主机系统120将这些改变写入到存储在存储器子系统110的数据存储媒体上的KVS树,所述存储器子系统在操作930处将这些改变提交给存储的KVS树。随后,在操作922处,主机系统120将(在操作908处所得的)经合并键值组移动到所述父节点的一组子节点。同样,主机系统120将这些改变写入到存储器子系统110,所述存储器子系统在操作932处将这些改变提交给存储的KVS树。

[0101] 在操作924处,主机系统120合并父节点的键值组序列以产生经合并键值组,所述经合并键值组包括参考一组新值块的一组新键块。如在本文中所提及,所述一组新键块基于所述键值组序列所包含的(例如,其键值组所包含的)一组现有键块而生成,且所述一组新值块基于所述一组现有键块所参考的一组现有值块而生成。为了合并父节点的所述键值组序列,主机系统120读取存储在存储器子系统110的数据存储媒体上的KVS树,且存储器子系统110在操作934处提供对KVS树的存取。

[0102] 在操作924之后,在操作926处,主机系统120确定用于KVS树的叶节点的一组最大数据生成编号。为实现这一点,主机系统120读取存储在存储器子系统110的数据存储媒体上的KVS树(例如,以勘测KVS树的叶节点),且存储器子系统110在操作936处提供对KVS树的存取。在操作928处,主机系统120确定在操作926处所确定的所述一组最大数据生成编号中的最小数据生成编号。在操作940处,主机系统120通过删除键值组序列的所有现有值块来删除所述键值组序列,且删除所述键值组序列的具有小于在操作928处所确定的最小数据生成编号的数据生成编号的任何现有值块。主机系统120将这些改变写入到存储在存储器子系统110的数据存储媒体上的KVS树,所述存储器子系统在操作944处将这些改变提交给存储的KVS树。随后,在操作942处,主机系统120将(在操作924处所得的)经合并键值组移动到所述父节点的一组子节点。同样,主机系统120将这些改变写入到存储器子系统110,所述存储器子系统在操作946处将这些改变提交给存储的KVS树。

[0103] 图10说明呈计算机系统1000的形式的实例机器,在所述计算机系统内可执行指令集,以用于使所述机器执行本文中所论述的任何一或多个方法。在一些实施例中,计算机系统1000可对应于主机系统(例如图1的主机系统120),所述主机系统包含、耦合到或利用存储器子系统(例如图1的存储器子系统110),或可用于执行控制器的操作(例如,执行操作系统以执行对应于图1的键合并移动组件122的操作)。在替代实施例中,所述机器可连接(例如联网)到局域网(LAN)、内联网、外联网和/或互联网中的其它机器。所述机器可作为对等

(或分布式)网络环境中的对等机器或作为云计算基础设施或环境中的服务器或客户端机器而以客户端-服务器网络环境中的服务器或客户端机器的容量进行操作。

[0104] 所述机器可以是个人计算机(PC)、平板PC、机顶盒(STB)、个人数字助理(PDA)、蜂窝电话、网络设备、服务器、网络路由器、网络交换机、网桥,或能够(循序或以其它方式)执行指定待由所述机器采取的动作的指令集的任何机器。此外,尽管说明了单个机器,但还应认为术语“机器”包含分别或共同地执行一组(或多组)指令以执行本文所论述的任何一或多个方法的任何机器集合。

[0105] 实例计算机系统1000包含经由总线1030彼此通信的处理装置1002、主存储器1004(例如,只读存储器(ROM)、快闪存储器、动态随机存取存储器(DRAM),例如同步DRAM(SDRAM)或Rambus DRAM(RDRAM)等)、静态存储器1006(例如,快闪存储器、静态随机存取存储器(SRAM)等)以及数据存储装置1018。

[0106] 处理装置1002表示一或多个通用处理装置,例如微处理器、中央处理单元等。更具体地说,处理装置1002可以是复杂指令集计算(CISC)微处理器、精简指令集计算(RISC)微处理器、超长指令字(VLIW)微处理器、实施其它指令集的处理器,或实施指令集的组的处理器。处理装置1002还可以是一或多个专用处理装置,例如专用集成电路(ASIC)、现场可编程门阵列(FPGA)、数字信号处理器(DSP)、网络处理器等。处理装置1002经配置以执行用于执行本文所论述的操作和步骤的指令1026。计算机系统1000还可包含网络接口装置1008以通过网络1020通信。

[0107] 数据存储装置1018可包含机器可读存储媒体1024(也称为计算机可读媒体),其上存储有体现本文所描述的任何一或多个方法或功能的一或多组指令1026或软件。指令1026还可在其由计算机系统1000执行的期间完全或至少部分地驻存在主存储器1004内和/或处理装置1002内,主存储器1004和处理装置1002也构成机器可读存储媒体。机器可读存储媒体1024、数据存储装置1018和/或主存储器1004可对应于图1的存储器子系统110。

[0108] 在一个实施例中,指令1026包含用于实施对应于具有部分计算跟踪的奇偶校验计算器(例如,图1的键合并移动组件122)的功能的指令。尽管在实例实施例中将机器可读存储媒体1024展示为单个媒体,但术语“机器可读存储媒体”应被认为包含存储一组或多组指令的单个媒体或多个媒体。术语“机器可读存储媒体”还应被认为包含能够存储或编码供机器执行且使机器执行本公开的任何一或多个方法的指令集的任何媒体。因此,应认为术语“机器可读存储媒体”包含但不限于固态存储器、光学媒体以及磁性媒体。

[0109] 已在针对计算机存储器内的数据位的操作的算法和符号表示方面呈现了先前详细描述的一些部分。这些算法描述和表示是数据处理领域中的技术人员用以将其工作的主旨最有效地传达给所属领域的其他技术人员的方式。算法在此处以及通常被认为是产生期望的结果的操作的自洽序列。所述操作是要求对物理量进行物理操控的操作。这些量通常但未必呈能够被存储、组合、比较和以其它方式操控的电或磁信号的形式。有时,主要出于通用的原因,已证明将这些信号称为位、值、元素、符号、字符、项、数字等是方便的。

[0110] 然而,应牢记,所有这些和类似术语应与适当物理量相关联,且仅仅是应用于这些量的方便标签。本公开可指计算机系统或类似电子计算装置的动作和过程,所述计算机系统或类似电子计算装置操控且将计算机系统的寄存器和存储器内表示为物理(电子)量的数据变换成类似地表示为计算机系统存储器或寄存器或其它此类信息存储系统内的物理

量的其它数据。

[0111] 本公开还涉及用于执行本文中的操作的设备。此设备可出于既定目的而专门构造,或其可包含由存储在计算机中的计算机程序选择性地激活或重新配置的通用计算机。此类计算机程序可存储在计算机可读存储媒体中,例如但不限于任何类型的盘(包含软盘、光盘、CD-ROM和磁性光盘)、只读存储器(ROM)、随机存取存储器(RAM)、可擦除可编程只读存储器(EPROM)、EEPROM、磁卡或光卡或适合存储电子指令的任何类型的媒体,各个媒体耦合到计算机系统总线。

[0112] 本文中呈现的算法和显示在本质上不与任何特定计算机或其它设备相关。各种通用系统可与根据本文中的教示的程序一起使用,或可证明构建更专用设备以执行所述方法是方便的。如上文描述中所阐述的那样来呈现多种这些系统的结构。另外,未参考任何特定编程语言来描述本公开。应了解,可使用各种编程语言来实施本文中所描述的本公开的教示。

[0113] 本公开可提供为计算机程序产品或软件,其可包含机器可读媒体,所述机器可读媒体上存储有指令,所述指令可用于编程计算机系统(或其它电子装置)以执行根据本公开的过程。机器可读媒体包含用于以机器(例如,计算机)可读的形式存储信息的任何机制。在一些实施例中,机器可读(例如计算机可读)媒体包含机器可读(例如计算机可读)存储媒体,例如只读存储器(ROM)、随机存取存储器(RAM)、磁盘存储媒体、光学存储媒体、快闪存储器组件等。

[0114] 在前述说明书中,已参考其特定实例实施例描述了本公开的实施例。应显而易见的是,可在不脱离如所附权利要求书中阐述的本公开的更广泛实施例的情况下对本公开进行各种修改。因此,应在说明性意义上而非限制性意义上看待说明书和附图。

[0115] 实例

[0116] 实例1是一种系统,其包括:一组存储器组件,其存储键值存储树数据结构,所述键值存储树数据结构包括一组节点,其中所述一组节点中的节点包括键值组序列;以及处理装置,其以操作方式耦合到所述一组存储器组件,且经配置以执行操作,所述操作包括:检测将所述键值组序列合并且从所述键值存储树数据结构的所述节点移动到所述节点的一组子节点的条件;以及响应于检测到所述条件:确定所述节点的所述一组子节点是否包括叶节点;且基于确定所述一组子节点是否包括所述叶节点,将所述键值组序列移动到所述一组子节点。

[0117] 在实例2中,实例1的主题任选地包含其中所述将所述键值组序列移动到所述一组子节点包括:响应于确定所述一组子节点不包括所述叶节点:合并所述键值组序列以产生经合并键值组,所述经合并键值组包括参考所述键值组序列的一组现有值块的一组新键块,且所述一组新键块基于所述键值组序列的一组现有键块而生成;以及将所述经合并键值组移动到所述节点的所述一组子节点中。

[0118] 在实例3中,实例1或实例2的主题任选地包含其中将所述经合并键值组移动到所述节点的所述一组子节点中包括:将所述经合并键值组划分成一组分离键值组,每个分离键值组被分配给所述一组子节点中的不同子节点;以及将所述一组分离键值组中的每个分离键值组移动到所述一组子节点中的经分配子节点。

[0119] 在实例4中,实例1到3中的任一者的主题任选地包含其中合并所述键值组序列以

产生所述经合并键值组包括在生成所述经合并键值之后:响应于确定所述一组子节点不包括所述叶节点,从所述节点删除所述键值组序列中的每个特定键值组,删除所述特定键值组包括删除所述特定键值组的一或多个键块,同时保留所述特定键值组的一或多个值块。

[0120] 在实例5中,实例1到4中任一者的主题任选地包含其中基于所述键值组序列的所述一组现有键块,通过复制所述一组现有键块以使得所述一组新键块包括对所述一组现有值块的一或多个参考来生成所述一组新键块。

[0121] 在实例6中,实例1到5中的任一者的主题任选地包含其中所述节点的所述一组值块中的每个特定值块被分配数据生成编号,所述数据生成编号指示针对所述键值存储树结构初始生成所述特定值块的序列次序,且所述将所述键值组序列移动到所述一组子节点包括:响应于确定所述一组子节点包括所述叶节点:合并所述键值组序列以产生包括参考一组新值块的一组新键块的经合并键值组,所述一组新键块基于所述键值组序列的一组现有键块而生成,所述一组新值块基于所述键值组序列的一组现有值块而生成,且所述一组新值块被分配有分配给所述一组现有值块中的任一值块的特定最大数据生成编号;以及将所述经合并键值组移动到所述节点的所述一组子节点中。

[0122] 在实例7中,实例1到6中的任一者的主题任选地包含其中将所述经合并键值组移动到所述节点的所述一组子节点中包括:将所述经合并键值组划分成一组分离键值组,每个分离键值组被分配给所述一组子节点中的不同子节点;以及将所述一组分离键值组中的每个分离键值组移动到所述一组子节点中的经分配子节点。

[0123] 在实例8中,实例1到7中的任一者的主题任选地包含其中所述合并所述键值组序列以产生所述经合并键值组包括在生成所述经合并键值之后:响应于确定所述一组子节点确实包括所述叶节点:针对所述键值存储树数据结构的每个特定叶节点,通过确定分配给与所述特定叶节点相关联的任何值块的最大数据生成编号,确定一组最大数据生成编号;确定所述一组最大数据生成编号中的最小数据生成编号;以及从所述节点删除所述键值组序列中的每个特定键值组,删除所述特定键值组包括:删除具有小于所述最小数据生成编号的特定数据生成编号的由所述特定键值组的现有键块参考的任何现有值块;以及删除所述特定键值组的一或多个现有键块。

[0124] 在实例9中,实例1到8中的任一者的主题任选地包含其中所述节点的所述一组值块中的每个特定值块被分配数据生成编号,所述数据生成编号指示针对所述键值存储树结构初始生成所述特定值块的序列次序,且所述操作还包括:确定所述一组最大数据生成编号中的最小数据生成编号;确定给定值块的特定数据生成编号小于所述最小数据生成编号;以及响应于确定所述特定数据生成编号小于所述最小数据生成编号,删除所述键值存储树数据结构的给定值块。

[0125] 在实例10中,实例1到9中的任一者的主题任选地包含其中所述系统是存储器子系统。

[0126] 在实例11中,实例1到10中的任一者的主题任选地包含其中主机系统包括处理装置,且存储器子系统包括所述一组存储器组件。

[0127] 实例12是一种方法,其包括:在一组存储器组件上生成键值存储树数据结构,所述键值存储树数据结构包括一组节点,其中所述一组节点中的节点包括键值组序列;由处理装置检测将所述键值组序列合并并且从所述节点移动到所述节点的一组子节点的条件;以及

响应于检测到所述条件:由所述处理装置确定所述节点的所述一组子节点是否包括叶节点;且由所述处理装置基于确定所述一组子节点是否包括所述叶节点,将所述键值组序列移动到所述一组子节点。

[0128] 在实例13中,实例12的主题任选地包含其中所述将所述键值组序列移动到所述一组子节点包括:响应于确定所述一组子节点不包括所述叶节点:合并所述键值组序列以产生经合并键值组,所述经合并键值组包括参考所述键值组序列的一组现有值块的一组新键块,且所述一组新键块基于所述键值组序列的一组现有键块而生成;以及将所述经合并键值组移动到所述节点的所述一组子节点中。

[0129] 在实例14中,实例12或实例13的主题任选地包含其中将所述经合并键值组移动到所述节点的所述一组子节点中包括:将所述经合并键值组划分成一组分离键值组,每个分离键值组被分配给所述一组子节点中的不同子节点;以及将所述一组分离键值组中的每个分离键值组移动到所述一组子节点中的经分配子节点。

[0130] 在实例15中,实例12到14中的任一者的主题任选地包含其中合并所述键值组序列以产生所述经合并键值组包括在生成所述经合并键值之后:响应于确定所述一组子节点不包括所述叶节点,从所述节点删除所述键值组序列中的每个特定键值组,删除所述特定键值组包括删除所述特定键值组的一或多个键块,同时保留所述特定键值组的一或多个值块。

[0131] 在实例16中,实例12到15中任一者的主题任选地包含其中基于所述键值组序列的所述一组现有键块,通过复制所述一组现有键块以使得所述一组新键块包括对所述一组现有值块的一或多个参考来生成所述一组新键块。

[0132] 在实例17中,实例12到16中的任一者的主题任选地包含其中所述节点的所述一组值块中的每个特定值块被分配数据生成编号,所述数据生成编号指示针对所述键值存储树结构初始生成所述特定值块的序列次序,且所述将所述键值组序列移动到所述一组子节点包括:响应于确定所述一组子节点包括所述叶节点:合并所述键值组序列以产生包括参考一组新值块的一组新键块的经合并键值组,所述一组新键块基于所述键值组序列的一组现有键块而生成,所述一组新值块基于所述键值组序列的一组现有值块而生成,且所述一组新值块被分配有分配给所述一组现有值块中的任一值块的特定最大数据生成编号;以及将所述经合并键值组移动到所述节点的所述一组子节点中。

[0133] 在实例18中,实例12到17中的任一者的主题任选地包含其中将所述经合并键值组移动到所述节点的所述一组子节点中包括:将所述经合并键值组划分成一组分离键值组,每个分离键值组被分配给所述一组子节点中的不同子节点;以及将所述一组分离键值组中的每个分离键值组移动到所述一组子节点中的经分配子节点。

[0134] 在实例19中,实例12到18中的任一者的主题任选地包含其中所述合并所述键值组序列以产生所述经合并键值组包括在生成所述经合并键值之后:响应于确定所述一组子节点确实包括所述叶节点:针对所述键值存储树数据结构的每个特定叶节点,通过确定分配给与所述特定叶节点相关联的任何值块的最大数据生成编号,确定一组最大数据生成编号;确定所述一组最大数据生成编号中的最小数据生成编号;以及从所述节点删除所述键值组序列中的每个特定键值组,删除所述特定键值组包括:删除具有小于所述最小数据生成编号的特定数据生成编号的由所述特定键值组的现有键块参考的任何现有值块;以及删

除所述特定键值组的一或多个现有键块。

[0135] 实例20是一种包括指令的非暂时性机器可读存储媒体,所述指令在由处理装置执行时使所述处理装置:在一组存储器组件上存取键值存储树数据结构,所述键值存储树数据结构包括一组节点,其中所述一组节点中的节点包括键值组序列;检测将所述键值组序列合并并且从所述节点移动到所述节点的一组子节点的条件;以及响应于检测到所述条件:确定所述节点的所述一组子节点是否包括叶节点;且基于确定所述一组子节点是否包括所述叶节点,将所述键值组序列移动到所述一组子节点。

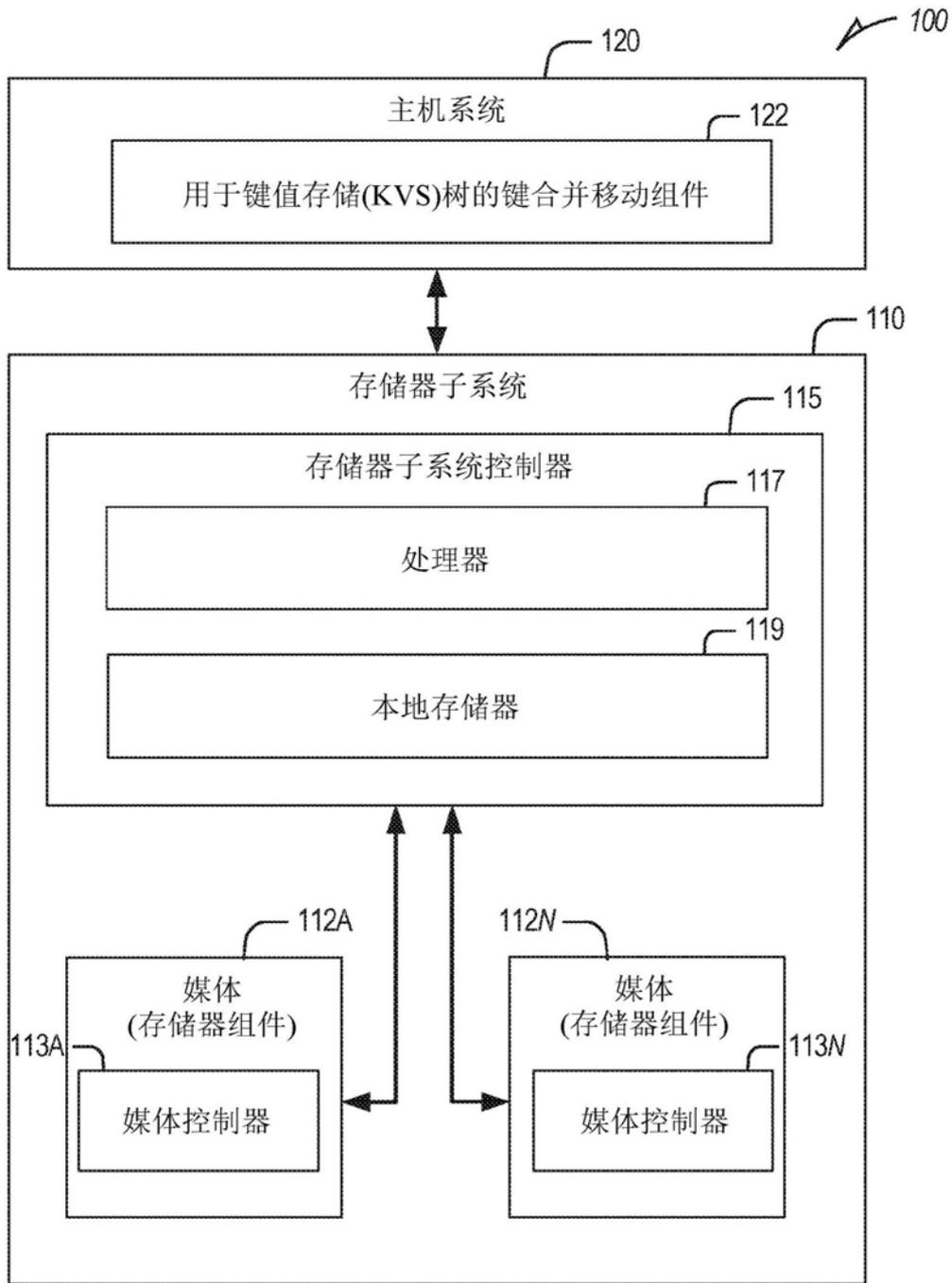


图1

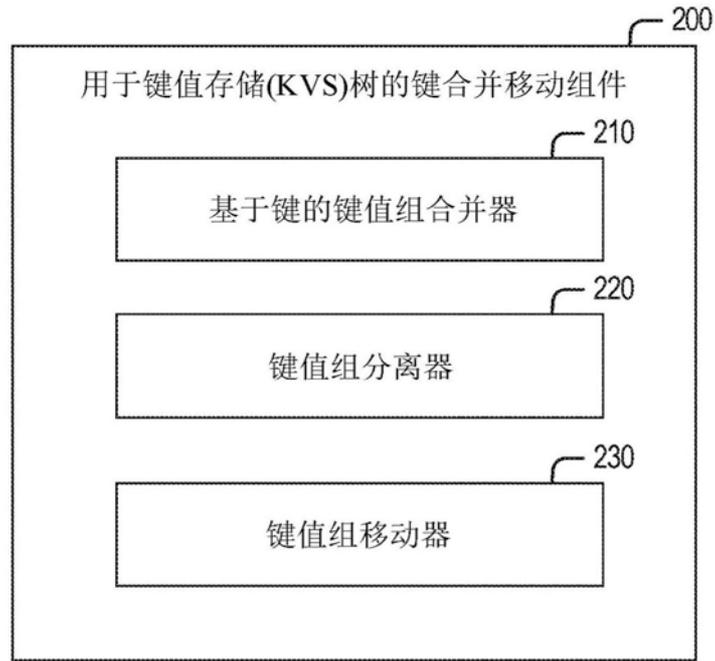


图2

300 ↗

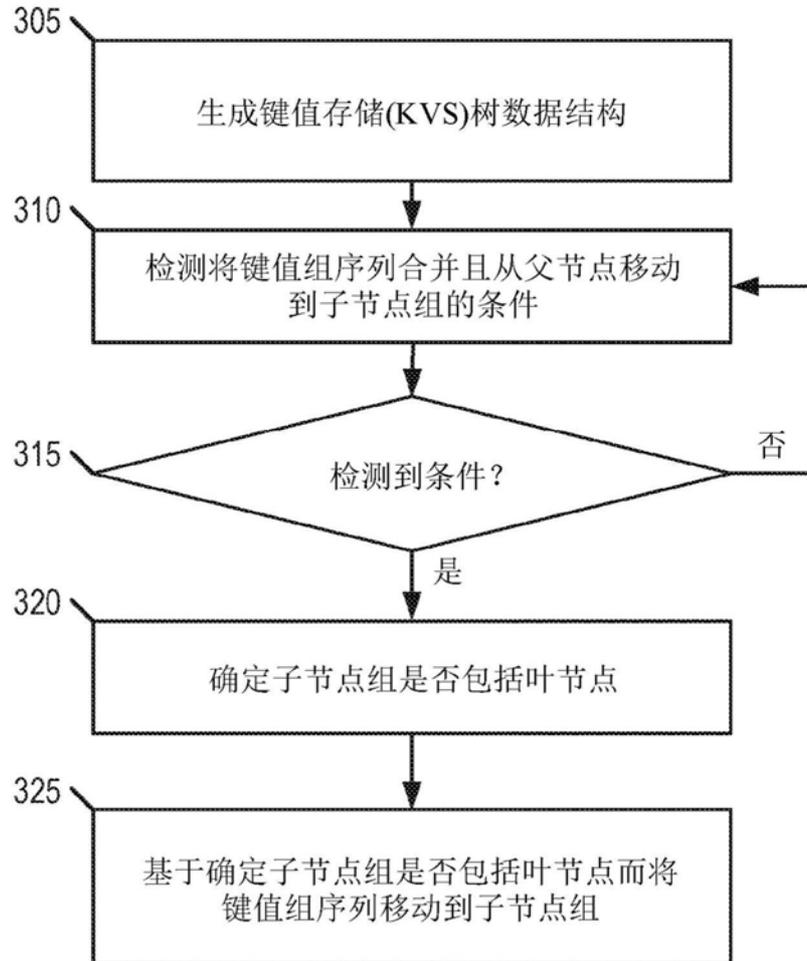


图3

400 ↘

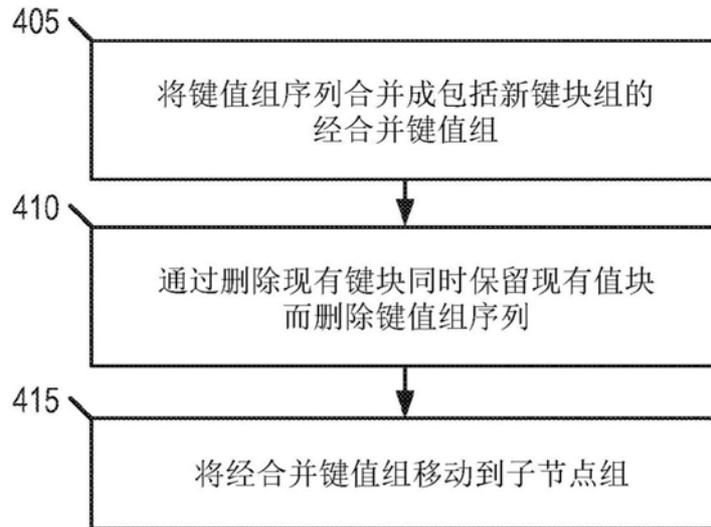


图4

500 ↘

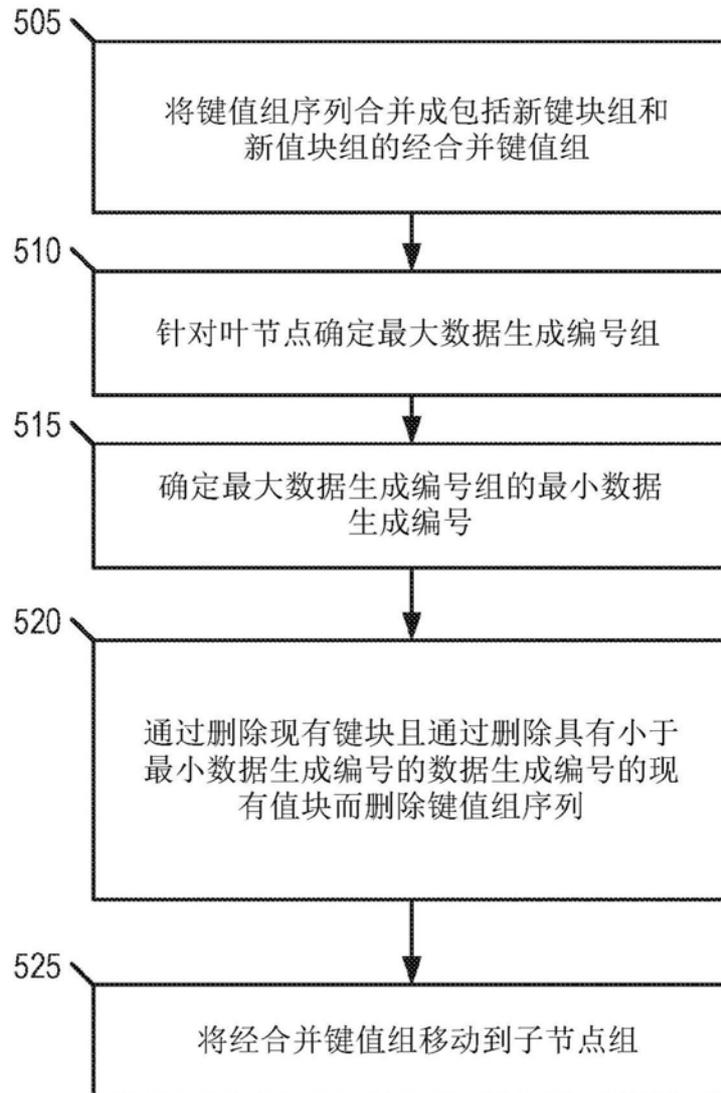


图5

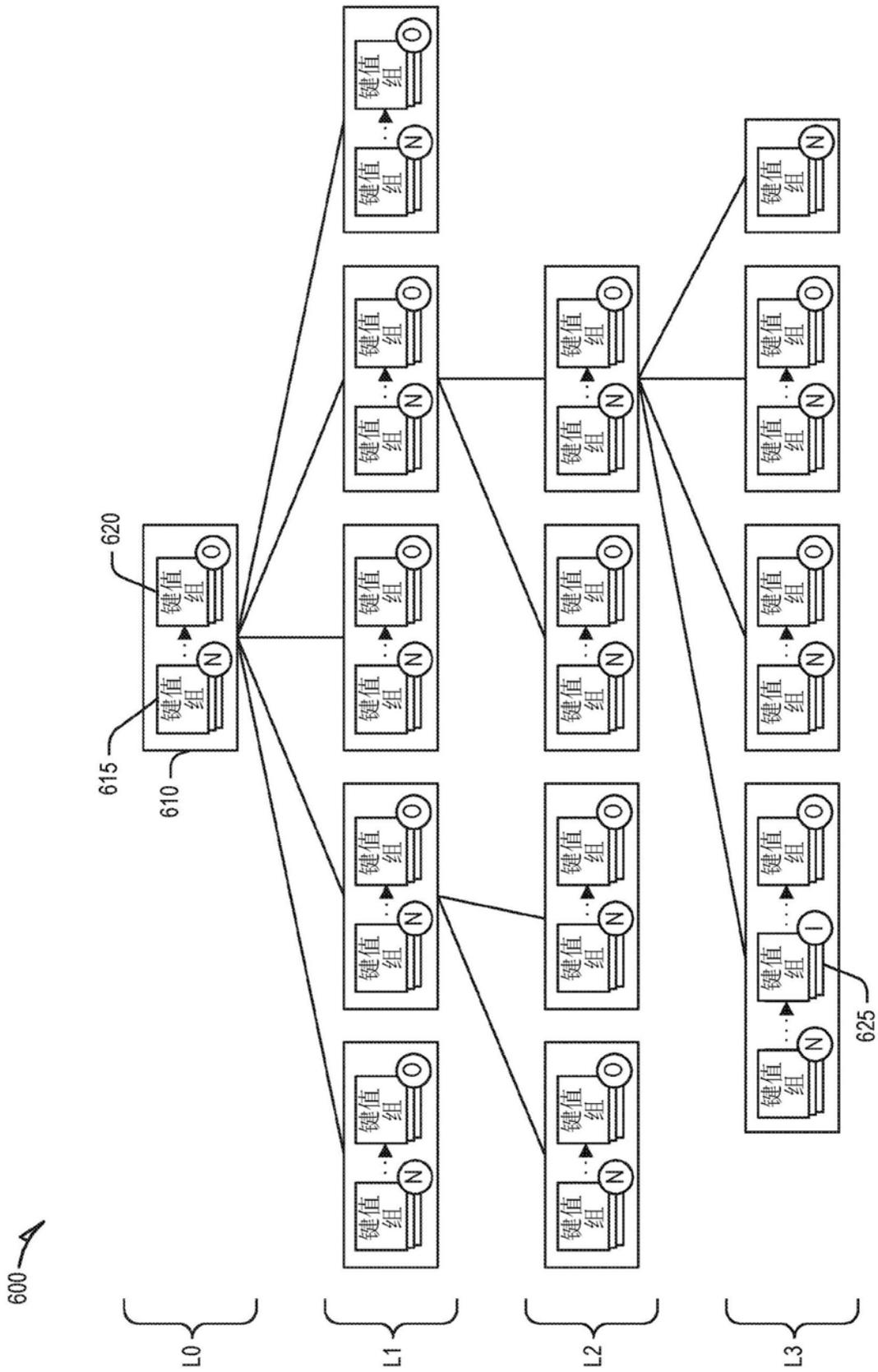


图6

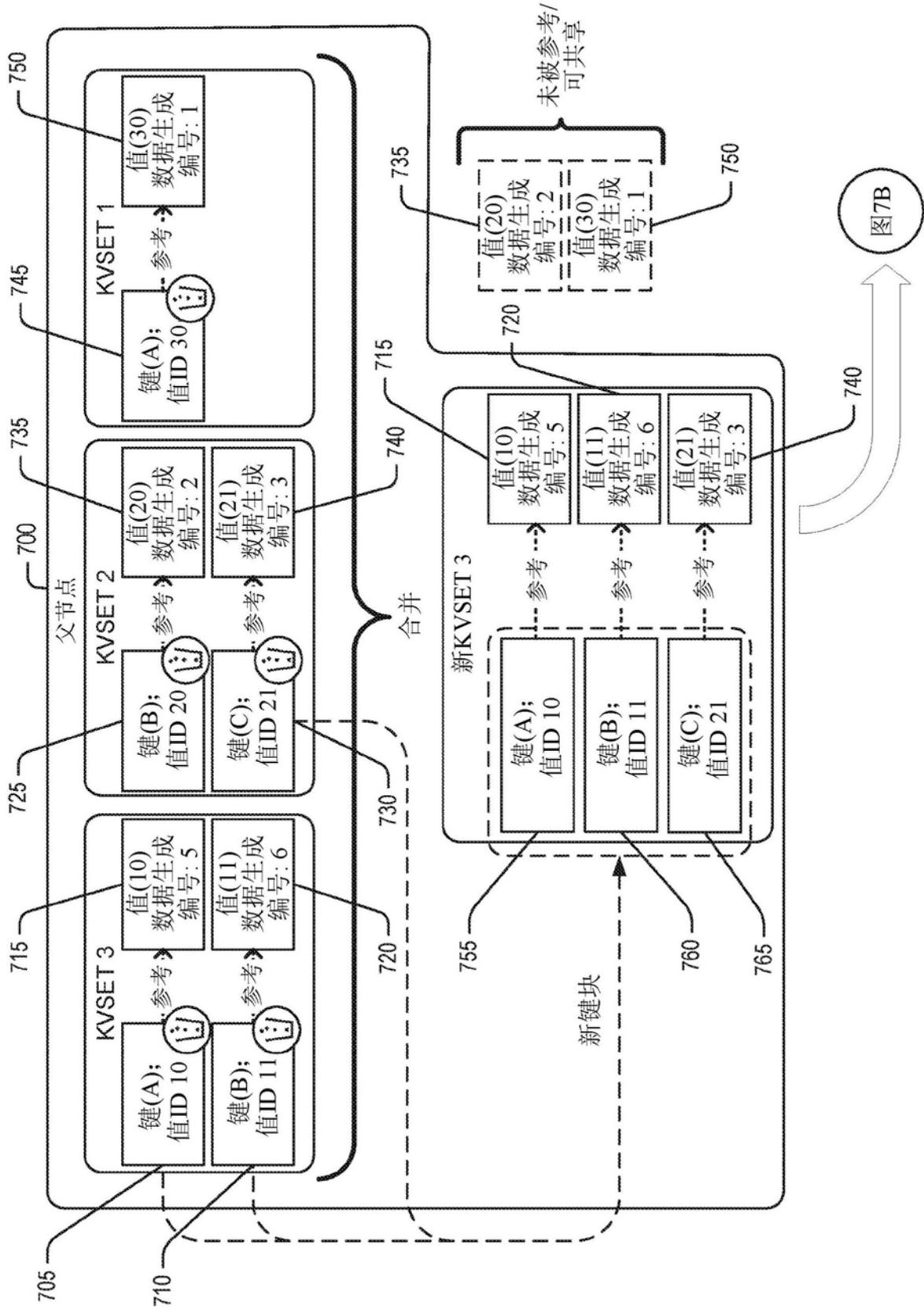


图7A

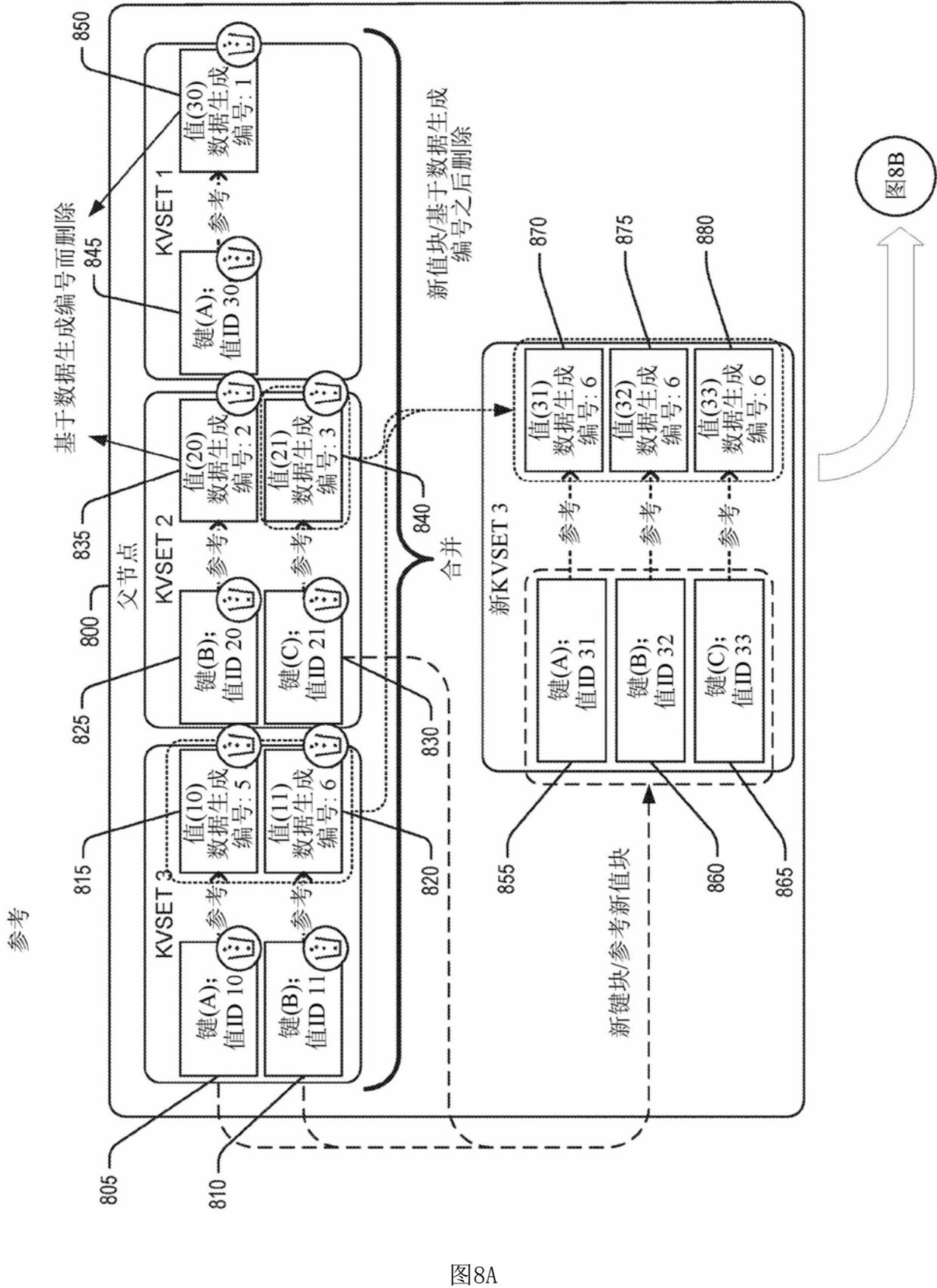


图8A

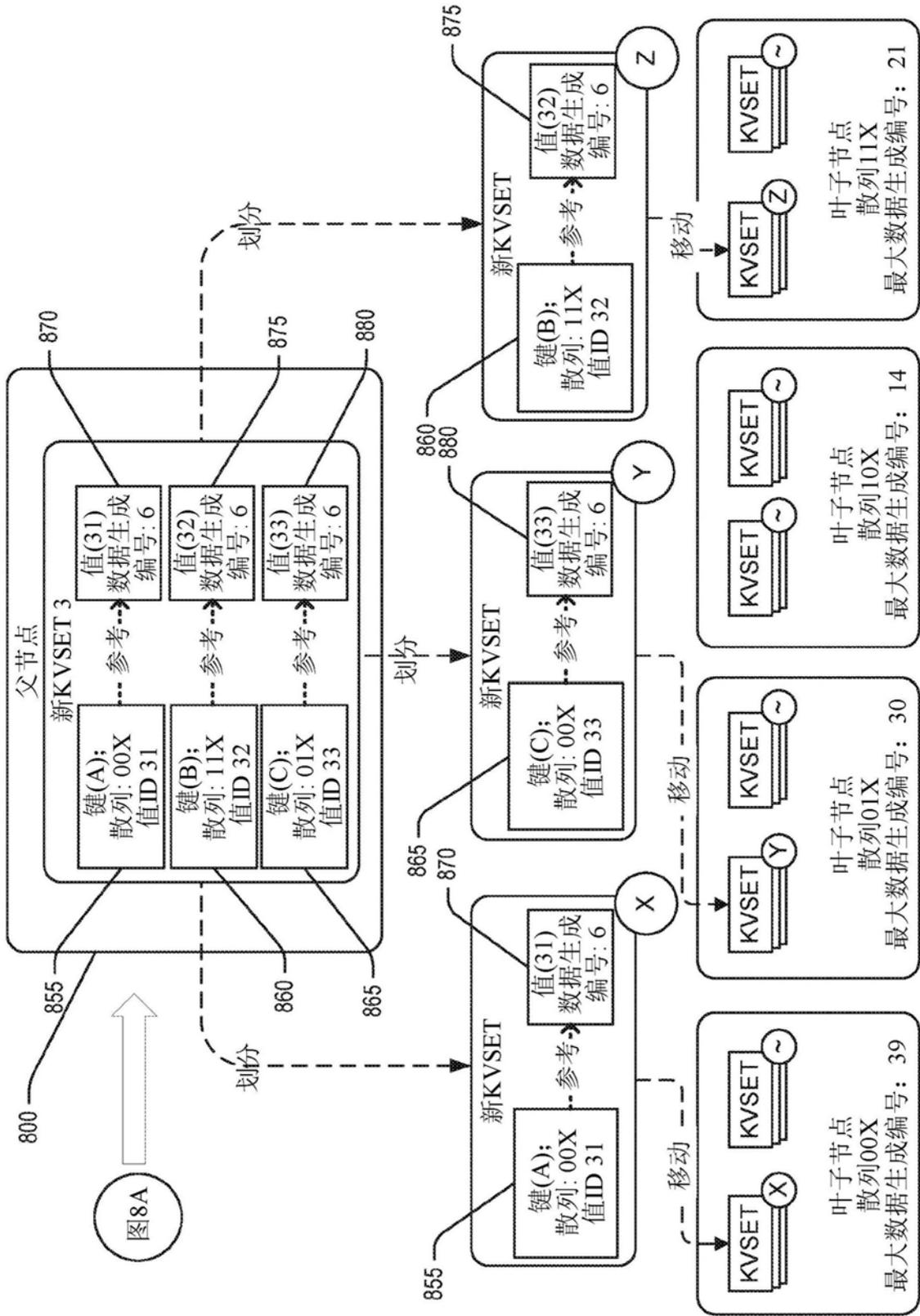


图8B

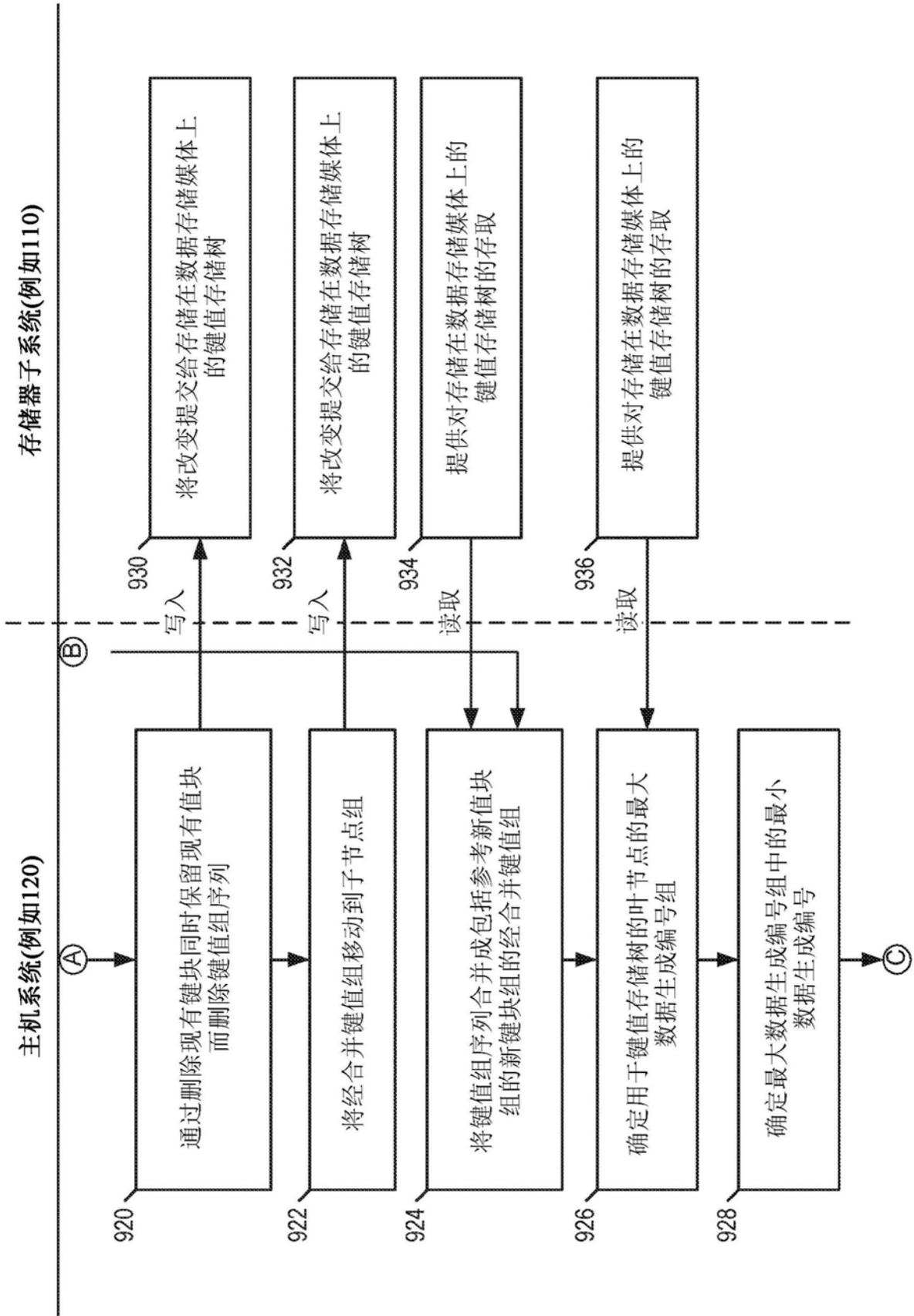


图9B

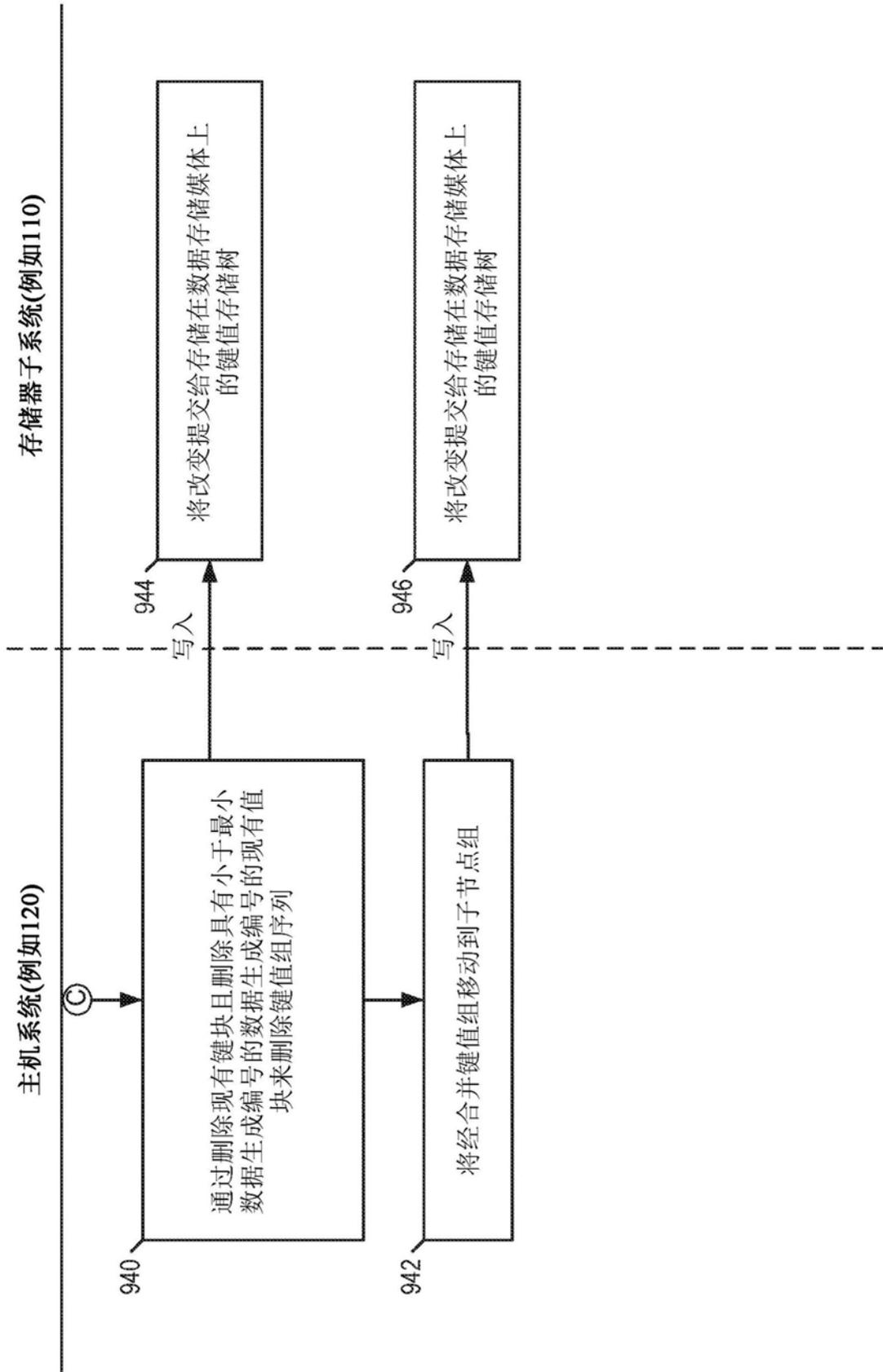


图9C

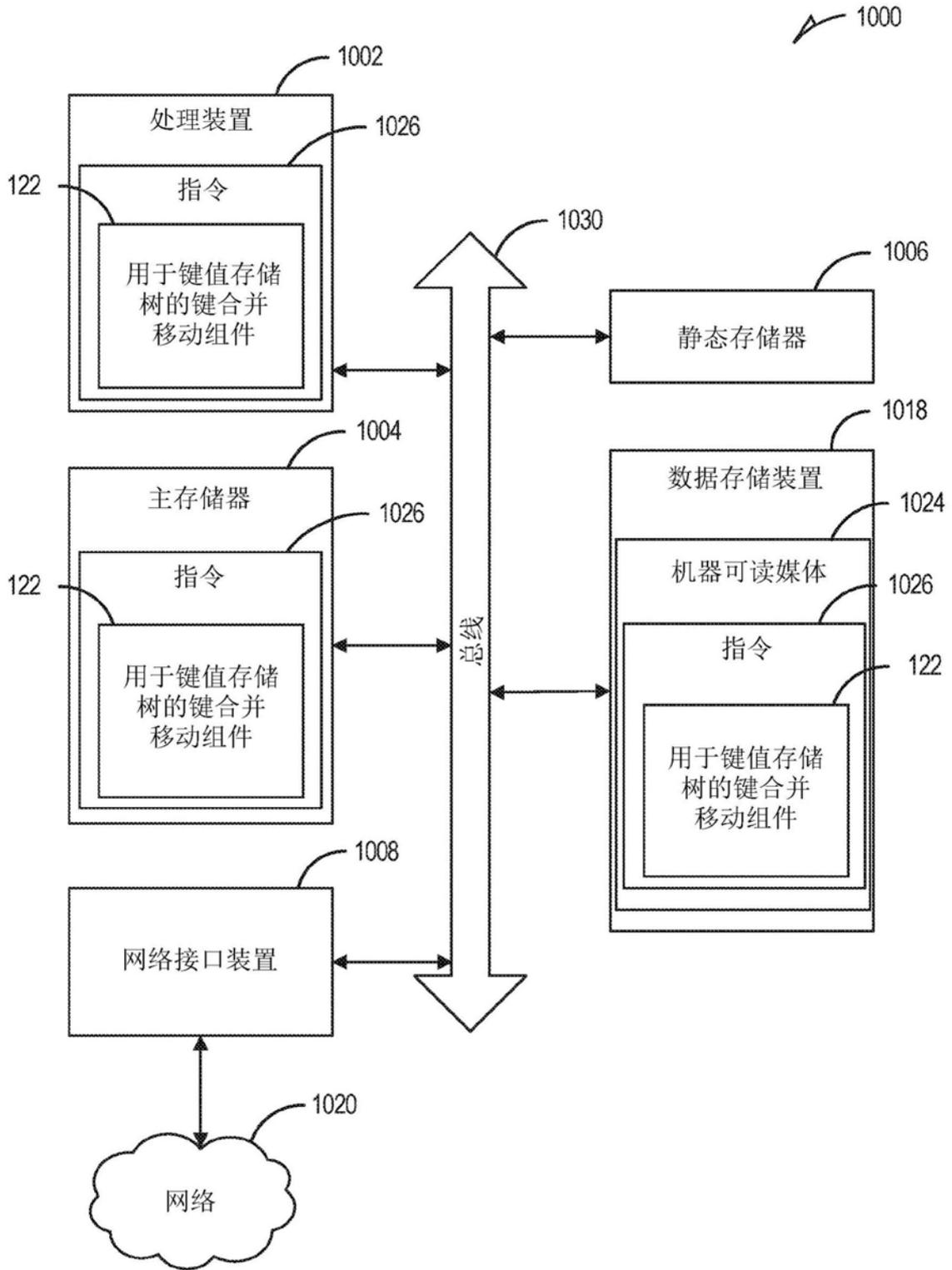


图10