



(12)发明专利申请

(10)申请公布号 CN 110268399 A

(43)申请公布日 2019.09.20

(21)申请号 201880011043.6

(22)申请日 2018.02.06

(30)优先权数据

15/428,951 2017.02.09 US

(85)PCT国际申请进入国家阶段日

2019.08.08

(86)PCT国际申请的申请数据

PCT/US2018/017043 2018.02.06

(87)PCT国际申请的公布数据

W02018/148198 EN 2018.08.16

(71)申请人 美光科技公司

地址 美国爱达荷州

(72)发明人 D·博尔斯 J·M·格罗韦斯

S·莫耶 A·汤姆林森

(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 王龙

(51)Int.Cl.

G06F 16/901(2019.01)

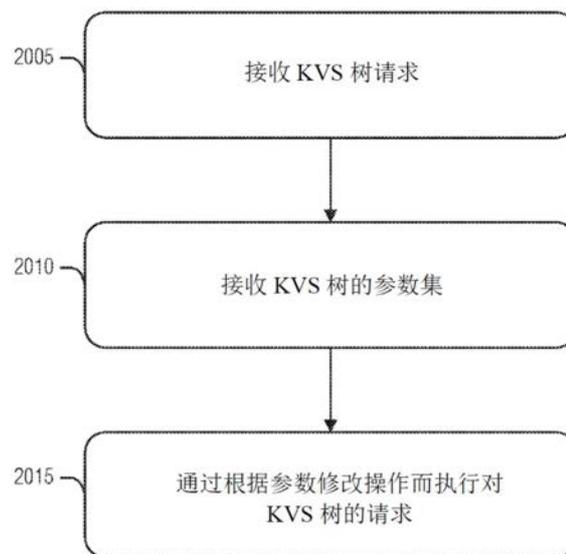
权利要求书4页 说明书41页 附图26页

(54)发明名称

用于维护操作的合并树修改

(57)摘要

在本文中描述用于维护操作的合并树修改的系统及技术。接收对KVS树的请求。在此处,所述KVS树为包含节点的数据结构,且所述节点包含时间上经定序的kvset序列,所述kvset以经排序次序存储键。接收所述KVS树的参数集。通过根据参数修改所述KVS树的操作而执行对所述KVS树的所述请求。



1. 一种系统,其包括经配置以进行以下操作的处理电路:
接收对KVS树的请求,所述KVS树为包含节点的数据结构,所述节点包含时间上经定序的kvset序列,所述kvset以经排序次序存储键;
接收所述KVS树的参数集;且
通过根据参数修改所述KVS树的操作而执行对所述KVS树的所述请求。
2. 根据权利要求1所述的系统,其中所述请求包含键前缀及逻辑删除,其中所述参数集具有在所述请求中将所述逻辑删除定义为前缀逻辑删除的成员,且其中为执行对所述KVS树的所述请求,所述处理电路经配置以将所述前缀逻辑删除写入到所述KVS树的kvset。
3. 根据权利要求1所述的系统,其中所述请求包含键,其中所述参数集包含规定逻辑删除加速度的成员;且其中为执行对所述KVS树的所述请求,所述处理电路经配置以将逻辑删除写入于通过对所述键执行溢出函数而规定的至少一个子节点中。
4. 根据权利要求3所述的系统,其中将所述逻辑删除写入到通过对所述键执行所述溢出函数而规定的所有现存子节点。
5. 根据权利要求1所述的系统,其中所述请求包含键、逻辑删除及在所述KVS树中与所述键对应的值的存储大小,其中所述参数集具有规定无用单元收集统计数据存储区的成员,且其中为执行对所述KVS树的所述请求,所述处理电路经配置以将所述键及所述存储大小存储于所述KVS树的数据结构中。
6. 根据权利要求1所述的系统,其中所述参数集包含规定所述KVS树为不可变的成员,其中为执行对所述KVS树的所述请求,所述处理电路经配置以将所述请求写入到所述KVS树的根节点。
7. 根据权利要求6所述的系统,其中当所述KVS树为不可变的时,所述KVS树排他地使用键压缩。
8. 根据权利要求7所述的系统,其中所述处理电路进一步经配置以:
响应于所述KVS树为不可变的而存储键搜索统计数据;且
响应于所述键搜索统计数据满足阈值而执行键压缩。
9. 根据权利要求8所述的系统,其中所述处理电路进一步经配置以响应于以下情形中的至少一者而将所述键搜索统计数据复位:压缩、引入、规定数目次搜索之后或规定时间间隔之后。
10. 根据权利要求6所述的系统,其中所述参数集的第二成员规定所述KVS树在先进先出基础上移除元素,其中所述参数集的第三成员规定所述KVS树的保留约束,其中所述KVS树基于所述保留约束而对kvset执行键压缩,且其中所述KVS树在违反所述保留约束时移除最旧kvset。
11. 至少一个机器可读媒体,其包含在由机器执行时致使所述机器执行包括以下各项的操作的指令:
接收对KVS树的请求,所述KVS树为包含节点的数据结构,所述节点包含时间上经定序的kvset序列,所述kvset以经排序次序存储键;
接收所述KVS树的参数集;及
通过根据参数修改所述KVS树的操作而执行对所述KVS树的所述请求。
12. 根据权利要求11所述的至少一个机器可读媒体,其中所述请求包含键前缀及逻辑

删除,其中所述参数集具有在所述请求中将所述逻辑删除定义为前缀逻辑删除的成员,且其中执行对所述KVS树的所述请求包含将所述前缀逻辑删除写入到所述KVS树的kvset。

13.根据权利要求12所述的至少一个机器可读媒体,其中在将键进行比较的KVS树操作时,前缀逻辑删除匹配具有与所述前缀逻辑删除的所述键前缀相同的前缀的任何键。

14.根据权利要求11所述的至少一个机器可读媒体,其中所述请求包含键,其中所述参数集包含规定逻辑删除加速度的成员;且其中所述执行对所述KVS树的所述请求包含将逻辑删除写入于通过对所述键执行溢出函数而规定的至少一个子节点中。

15.根据权利要求14所述的至少一个机器可读媒体,其中将所述逻辑删除写入到通过对所述键执行所述溢出函数而规定的所有现存子节点。

16.根据权利要求14所述的至少一个机器可读媒体,其中所述请求包含逻辑删除。

17.根据权利要求14所述的至少一个机器可读媒体,其中所述请求包含值。

18.根据权利要求11所述的至少一个机器可读媒体,其中所述请求包含键、逻辑删除及在所述KVS树中与所述键对应的值的存储大小,其中所述参数集具有规定无用单元收集统计数据存储区的成员,且其中执行对所述KVS树的所述请求包含将所述键及所述存储大小存储于所述KVS树的数据结构中。

19.根据权利要求18所述的至少一个机器可读媒体,其中所述逻辑删除为前缀逻辑删除。

20.根据权利要求11所述的至少一个机器可读媒体,其中所述参数集包含规定所述KVS树为不可变的成员,其中执行对所述KVS树的所述请求包含将所述请求写入到所述KVS树的根节点。

21.根据权利要求20所述的至少一个机器可读媒体,其中当所述KVS树为不可变的时,所述KVS树排他地使用键压缩。

22.根据权利要求21所述的至少一个机器可读媒体,其中所述操作包括:

响应于所述KVS树为不可变的而存储键搜索统计数据;及

响应于所述键搜索统计数据满足阈值而执行键压缩。

23.根据权利要求22所述的至少一个机器可读媒体,其中所述键搜索统计数据为最小、最大、平均数或平均值搜索时间中的至少一者。

24.根据权利要求22所述的至少一个机器可读媒体,其中所述键搜索统计数据为所述根节点中的kvset数目。

25.根据权利要求22所述的至少一个机器可读媒体,其中所述操作包括响应于以下情形中的至少一者而将所述键搜索统计数据复位:压缩、引入、规定数目次搜索之后或规定时间间隔之后。

26.根据权利要求20所述的至少一个机器可读媒体,其中所述参数集的第二成员规定所述KVS树在先进先出基础上移除元素,其中所述参数集的第三成员规定所述KVS树的保留约束,其中所述KVS树基于所述保留约束而对kvset执行键压缩,且其中所述KVS树在违反所述保留约束时移除最旧kvset。

27.根据权利要求26所述的至少一个机器可读媒体,其中基于所述保留约束而对kvset执行键压缩包含:

将连续kvset分组以产生群组集,来自所述群组集中的每一成员的经求和指标约计所

述保留约束的分数;及

对所述群组集的每一成员执行键压缩。

28. 根据权利要求26所述的至少一个机器可读媒体,其中所述保留约束为最大键值对数目。

29. 根据权利要求26所述的至少一个机器可读媒体,其中所述保留约束为键值对的最大年龄。

30. 根据权利要求26所述的至少一个机器可读媒体,其中所述保留约束为由键值对消耗的最大存储值。

31. 一种机器实施的方法,其包括:

接收对KVS树的请求,所述KVS树为包含节点的数据结构,所述节点包含时间上经定序的kvset序列,所述kvset以经排序次序存储键;

接收所述KVS树的参数集;及

通过根据参数修改所述KVS树的操作而执行对所述KVS树的所述请求。

32. 根据权利要求31所述的方法,其中所述请求包含键前缀及逻辑删除,其中所述参数集具有在所述请求中将所述逻辑删除定义为前缀逻辑删除的成员,且其中执行对所述KVS树的所述请求包含将所述前缀逻辑删除写入到所述KVS树的kvset。

33. 根据权利要求31所述的方法,其中所述请求包含键,其中所述参数集包含规定逻辑删除加速度的成员;且其中所述执行对所述KVS树的所述请求包含将逻辑删除写入于通过对所述键执行溢出函数而规定的至少一个子节点中。

34. 根据权利要求31所述的方法,其中所述参数集包含规定所述KVS树为不可变的成员,其中执行对所述KVS树的所述请求包含将所述请求写入到所述KVS树的根节点。

35. 根据权利要求34所述的方法,其中当所述KVS树为不可变的时,所述KVS树排他地使用键压缩。

36. 根据权利要求35所述的方法,其包括:

响应于所述KVS树为不可变的而存储键搜索统计数据;及

响应于所述键搜索统计数据满足阈值而执行键压缩。

37. 一种系统,其包括:

用于接收对KVS树的请求的构件,所述KVS树为包含节点的数据结构,所述节点包含时间上经定序的kvset序列,所述kvset以经排序次序存储键;

用于接收所述KVS树的参数集的构件;及

用于通过根据参数修改所述KVS树的操作而执行对所述KVS树的所述请求的构件。

38. 根据权利要求37所述的系统,其中所述请求包含键前缀及逻辑删除,其中所述参数集具有在所述请求中将所述逻辑删除定义为前缀逻辑删除的成员,且其中执行对所述KVS树的所述请求包含将所述前缀逻辑删除写入到所述KVS树的kvset。

39. 根据权利要求37所述的系统,其中所述请求包含键,其中所述参数集包含规定逻辑删除加速度的成员;且其中所述执行对所述KVS树的所述请求包含将逻辑删除写入于通过对所述键执行溢出函数而规定的至少一个子节点中。

40. 根据权利要求37所述的系统,其中所述参数集包含规定所述KVS树为不可变的成员,其中执行对所述KVS树的所述请求包含将所述请求写入到所述KVS树的根节点。

41. 根据权利要求40所述的系统,其中当所述KVS树为不可变的时,所述KVS树排他地使用键压缩。

42. 根据权利要求41所述的系统,其包括:

用于响应于所述KVS树为不可变的而存储键搜索统计数据的构件;及

用于响应于所述键搜索统计数据满足阈值而执行键压缩的构件。

用于维护操作的合并树修改

[0001] 优先权申请案

[0002] 本申请案主张2017年2月9日提出申请的第15/428,951号美国申请案的优先权权益,所述美国申请案以其全文引用方式并入本文中。

技术领域

[0003] 本文中所描述的实施例一般来说涉及一种键值数据存储,且更具体来说涉及用于维护操作的合并树修改。

背景技术

[0004] 数据结构为准许用各种方式与存储于其中的数据相互作用的数据组织。数据结构可经设计以尤其准许(例如)在二进制搜索树中对数据进行高效搜索,准许(例如)利用链接列表对稀疏数据进行高效存储,或准许(例如)利用B树对可搜索数据进行高效存储。

[0005] 键值数据结构接受键值对且经配置以对对键的查询做出响应。键值数据结构可包含例如字典(例如,映射、散列映射等)的结构,其中键存储于链接(或含有)相应值的列表中。虽然这些结构在内存中(例如,在与存储区相对的主要或系统状态存储器中)是可用的,但这些结构在持久存储区中(例如,在磁盘上)的存储表示可为低效的。因此,已引入一类基于日志的存储结构。实例为日志结构化合并树(LSM树)。

[0006] 已存在各种LSM树实施方案,但许多LSM树实施方案符合其中将键值对接受到经键排序的内存中结构中的设计。当所述内存中结构填满时,数据分配在若干子节点当中。所述分配使得子节点中的键在子节点自身内而且在子节点之间经定序。举例来说,在具有三个子节点的第一树层级处,最左子节点内的最大键小于来自中间子节点的最小键且中间子节点中的最大键小于来自最右子节点的最小键。此结构准许对数据结构中的键以及键范围两者进行高效搜索。

附图说明

[0007] 在图式(其未必按比例绘制)中,相似编号可在不同视图中描述类似组件。具有不同字母后缀的相似编号可表示类似组件的不同例子。图式一般以实例方式而非以限制方式图解说明本文件中所论述的各种实施例。

[0008] 图1图解说明根据实施例的KVS树的实例。

[0009] 图2是根据实施例的图解说明对多流存储装置的写入的实例的框图。

[0010] 图3图解说明根据实施例的用以促进对多流存储装置进行写入的方法的实例。

[0011] 图4是根据实施例的图解说明用于键及值的存储组织的实例的框图。

[0012] 图5是根据实施例的图解说明键块及值块的配置的实例的框图。

[0013] 图6图解说明根据实施例的KB树的实例。

[0014] 图7是根据实施例的图解说明KVS树引入的框图。

[0015] 图8图解说明根据实施例的用于KVS树引入的方法的实例。

- [0016] 图9是根据实施例的图解说明键压缩的框图。
- [0017] 图10图解说明根据实施例的用于键压缩的方法的实例。
- [0018] 图11是根据实施例的图解说明键值压缩的框图。
- [0019] 图12图解说明根据实施例的用于键值压缩的方法的实例。
- [0020] 图13图解说明根据实施例的溢出值及其与树的关系的实例。
- [0021] 图14图解说明根据实施例的用于溢出值函数的方法的实例。
- [0022] 图15是根据实施例的图解说明溢出压缩的框图。
- [0023] 图16图解说明根据实施例的用于溢出压缩的方法的实例。
- [0024] 图17是根据实施例的图解说明提升压缩的框图。
- [0025] 图18图解说明根据实施例的用于提升压缩的方法的实例。
- [0026] 图19图解说明根据实施例的用于对KVS树执行维护的方法的实例。
- [0027] 图20图解说明根据实施例的用于修改KVS树操作的方法的实例。
- [0028] 图21是根据实施例的图解说明键搜索的框图。
- [0029] 图22图解说明根据实施例的用于执行键搜索的方法的实例。
- [0030] 图23是根据实施例的图解说明键扫描的框图。
- [0031] 图24是根据实施例的图解说明键扫描的框图。
- [0032] 图25是根据实施例的图解说明前缀扫描的框图。
- [0033] 图26是图解说明机器的实例的框图,可在所述机器上实施一或多个实施例。

具体实施方式

[0034] LSM树已成为数据的受欢迎存储结构,其中预期高容量写入而且预期对数据的高效存取。为支持这些特征,LSM的若干部分针对上面保持有所述部分的媒体经调谐且后台进程一般解决使数据在不同部分之间移动(例如,从内存中部分到磁盘上部分)。在本文中,内存中是指随机存取且字节可寻址装置(例如,静态随机存取存储器(SRAM)或动态随机存取存储器(DRAM)),且磁盘上是指块可寻址装置(例如,硬盘驱动器、光盘、数字多功能光盘或固态驱动器(SSD),例如基于快闪存储器的装置),其还称为媒体装置或存储装置。LSM树利用由内存中装置提供的就绪存取来将传入数据按键排序,以提供对对应值的就绪存取。当将数据合并到磁盘上部分上时,驻存于磁盘上数据与新数据合并且以块形式回写到磁盘。

[0035] 虽然LSM树已成为构成若干个数据库与容量存储(例如,云存储)设计的基础的受欢迎结构,但其确实具有一些缺点。首先,新数据与旧数据不断合并以使内部结构保持按键排序会引起显著写入放大率。写入放大率为由给定存储技术强加的最小数据写入次数的增加。举例来说,为存储数据,将数据写入到磁盘至少一次。此可(举例来说)通过仅仅将最新数据片附加到已经写入数据的末尾上而完成。然而,此结构的搜索速度缓慢(例如,其随数据量而线性增长),且可在改变或删除数据时引起低效。LSM树增加写入放大率,因为其从磁盘读取将与新数据合并的数据且接着将所述数据往回重写到磁盘。写入放大率问题可在包含存储装置活动(例如对硬盘驱动器进行碎片整理或SSD的无用单元收集)时加剧。SSD上的写入放大率可为尤其有害的,因为这些装置可随着若干次写入而“耗损”。也就是说,SSD具有以写入来测量的有限寿命。因此,关于SSD的写入放大率使得缩短基础硬件的可用寿命。

[0036] 关于LSM树的第二问题包含在执行合并时可消耗的大量空间。LSM树确保将磁盘上

部分按键排序。如果驻存于磁盘上的数据量太大,那么可消耗大量临时或暂存空间来执行合并。此可通过将磁盘上部分划分成非重叠结构以准许对数据子集的合并来得到稍微缓解,但可难以实现结构开销与性能之间的平衡。

[0037] 关于LSM树的第三问题包含可能有限的写入吞吐量。此问题起源于LSM数据的全部的基本上始终经排序本质。因此,压倒内存中部分的大容量写入必须等待直到利用可能耗时合并操作清除内存中部分为止。为解决此问题,已提议写入缓冲器(WB)树,其中操纵较小数据插入以在此情景中避免合并问题。具体来说,WB树对传入键进行散列以散布数据,且将键散列与值组合存储于较小摄入集中。这些集可在各种时间处经合并或基于键散列值而写入到子节点。此避免LSM树的昂贵合并操作同时在查找特定键时为高性能的。然而,按键散列排序的WB树引起昂贵整树扫描以对未直接由键散列引用的值进行定位,例如发生在对键范围进行搜索时。

[0038] 为解决上文所述的问题,在本文中描述KVS树及对应操作。KVS树为树数据结构,所述树数据结构包含基于键的预定导出而非树的内容而在母节点与子节点之间具有连接的节点。所述节点包含时间上经定序的键值集(kvset)序列。所述kvset含有在经键排序结构中的键值对。Kvset一旦经写入便也为不可变的。KVS树实现WB树的写入吞吐量同时通过维护节点中的kvset而对WB树搜索进行改进以提供对kvset的高效搜索,所述kvset包含经排序键以及(在实例中)键指标(例如布隆过滤器、最小键及最大键等)。在许多实例中,KVS树可通过将键与值分隔且合并较小kvset集合而对LSM树的临时存储问题进行改进。另外,所描述KVS树可通过对kvset的各种维护操作来降低写入放大率。此外,当节点中的kvset为不可变的时,例如SSD上的写入耗损的问题可由数据结构管理,从而减少装置自身的无用单元收集活动。此具有如下额外益处:释放内部装置资源(例如,总线带宽、处理循环等)从而引起更佳外部驱动性能(例如,读取或写入速度)。在下文描述KVS树及其上的操作的额外细节及实例性实施方案。

[0039] 图1图解说明根据实施例的KVS树100的实例。KVS树100为组织成树的键值数据结构。作为键值数据结构,值与引用所述值的对应键一起存储于树100中。具体来说,键条目用于含有键及额外信息(例如对值的引用)两者,然而,除非另有规定,否则键条目为了简单而简称为键。键自身在树100内具有总定序。因此,键可在彼此当中进行排序。键还可划分成子键。一般来说,子键为键的非重叠部分。在实例中,键的总定序基于在多个键之间将相似子键进行比较(例如,将一键的第一子键与另一键的所述第一子键进行比较)。在实例中,键前缀为键的开始部分。所述键前缀可由一或多个子键构成(在使用所述子键时)。

[0040] 树100包含一或多个节点,例如节点110。节点110包含时间上经定序的不可变键值集(kvset)序列。如所图解说明,kvset 115包含‘N’徽章以指示其为序列中的最新者,而kvset 120包含‘O’徽章以指示其为序列中的最旧者。Kvset 125包含‘I’徽章以指示其为序列中的中间者。通篇使用这些徽章来给kvset加标签,然而,另一徽章(例如‘X’)表示特定kvset而非其在序列中的位置(例如,新的、中间的、旧的等),除非其为波形符‘~’,在所述情形中其仅仅为匿名kvset。如下文更详细地阐释,较旧键值条目出现在树100中较低处。因此,使值上升一树层级(例如从L2到L1)会在接收方节点中的最旧位置中产生新kvset。

[0041] 节点110还包含节点的kvset中的键值对到节点110的任何一个子节点的确定性映射。如本文中所使用,所述确定性映射意味:给定键值对,外部实体可在不知晓树100的内容

的情况下追踪可能子节点的穿过树100的路径。此(举例来说)与B树相当不同,举例来说,其中树的内容将确定给定键的值将落在何处以便维护树的搜索经优化结构。替代地,在此处,所述确定性映射提供规则,使得(举例来说)给定键值对,可计算此对将映射的L3处的子节点,即使最大树层级(例如,树深度)仅在L1处。在实例中,所述确定性映射包含键的一部分的散列的一部分。因此,可对子键进行散列以到达映射集。可针对树的任一给定层级使用此集的一部分。在实例中,所述键的所述部分为整个键。没有理由可不使用整个键。

[0042] 在实例中,所述散列包含多个非重叠部分,所述多个非重叠部分包含所述散列的所述部分。在实例中,所述多个非重叠部分中的每一者对应于树的层级。在实例中,由节点的层级依据所述多个非重叠部分确定所述散列的所述部分。在实例中,所述节点的最大子节点数目由所述散列的所述部分的大小定义。在实例中,所述散列的所述部分的所述大小为位数目。这些实例可通过采用产生8个位的键的散列来图解说明。可将这八个位划分成如下三个集:前两个位;第三个到第六个位(产生四个位);及第七个及第八个位。子节点可为基于位集的指标,使得第一层级(例如,L1)处的子节点具有两位名称,第二层级(例如,L2)上的子节点具有四位名称,且第三层级(例如,L3)上的子节点具有两位名称。下文关于图13及14包含经展开论述。

[0043] Kvset为组织在树100的节点中的键与值存储区。Kvset的不可变性意味:kvset一旦放置于节点中便不改变。然而,可删除kvset,可将其内容中的一些或所有内容添加到新kvset等。在实例中,kvset的不可变性还扩展到含纳于kvset内的任何控制或元数据。此一般为可能的,因为元数据所应用到的内容为不变的且因此元数据在那时通常也将为静态的。

[0044] 还注意,KVS树100不需要遍及树100的键之间的唯一性,但kvset具有键的仅一者。也就是说,给定kvset中的每个键不同于所述kvset的其它键。此最后陈述对于特定kvset成立,且因此在(举例来说)kvset经版本化时不可适用。Kvset版本化对于形成数据的快照可为有帮助的。在经版本化kvset的情况下,通过kvset标识(ID)与版本的组合来确定kvset中的键的唯一性。然而,两个不同kvset(例如,kvset 115及kvset 120)可各自包含相同键。

[0045] 在实例中,所述kvset包含键树以存储所述kvset的键值对的键条目。各种数据结构可用于高效地存储且检索例如二进制搜索树、B树等键树(其甚至可并非树)中的唯一键。在实例中,所述键存储于所述键树的叶节点中。在实例中,所述键树的任一子树中的最大键在最右子节点的最右条目中。在实例中,所述键树的第一节点的最右边缘链接到所述键树的子节点。在实例中,生根于所述键树的所述子节点处的子树中的所有键大于所述键树的所述第一节点中的所有键。此最后几个实例图解说明KB树的特征,如下文关于图6所论述。

[0046] 在实例中,所述kvset的键条目存储于包含主要键块及零个或更多个扩展键块的键块集中。在实例中,所述键块集的成员对应于存储媒体(例如SSD、硬盘驱动器等)的媒体块。在实例中,每一键块包含用以将其识别为键块的标头。在实例中,所述主要键块包含所述kvset的所述一或多个扩展键块的媒体块标识列表。

[0047] 在实例中,所述主要键块包含所述kvset的键树的标头。所述标头可包含若干个值以使与所述键或一般来说所述kvset相互作用更容易。在实例中,所述主要键块或标头包含所述kvset的键树中的最低键的副本。在此处,所述最低键通过所述树的预设定排序次序(例如,树100中的键的总定序)来确定。在实例中,所述主要键块包含所述kvset的键树中的

最高键的副本,所述最高键通过所述树的预设定排序次序来确定。在实例中,所述主要键块包含所述kvset的键树的媒体块标识列表。在实例中,所述主要键块包含所述kvset的布隆过滤器的布隆过滤器标头。在实例中,所述主要键块包含所述kvset的布隆过滤器的媒体块标识列表。

[0048] 在实例中,所述kvset的值存储于值块集中。在此处,所述值块集的成员对应于所述存储媒体的媒体块。在实例中,每一值块包含用以将其识别为值块的标头。在实例中,值块包含一或多个值的存储区段,在所述一或多个值之间不具有间隔。因此,第一值的位在所述存储媒体上运行成第二值的位,在其之间不具有防护、容器或其它分隔符。在实例中,所述主要键块包含所述值块集中的值块的媒体块标识列表。因此,所述主要键块管理对值块的存储引用。

[0049] 在实例中,所述主要键块包含所述kvset的指标集。在实例中,所述指标集包含存储于所述kvset中的键的总数目。在实例中,所述指标集包含存储于所述kvset中的具有逻辑删除(tombstone)值的键的数目。如本文中所使用,逻辑删除为指示已删除与键对应的值的数据标记(data marker)。一般来说,逻辑删除将驻存于键条目中且针对此键值对将不消耗值块空间。逻辑删除的目的为将值的删除加标记同时避免清除来自树100的值的的可能昂贵操作。因此,当使用时间上经定序搜索遇到逻辑删除时,知晓对应值经删除,即使键值对的过期版本驻存于树100内的较旧位置处。

[0050] 在实例中,存储于所述主要键块中的所述指标集包含存储于所述kvset中的键的所有键长度的和。在实例中,所述指标集包含存储于所述kvset中的键的所有值长度的和。这最后两个指标给出由所述kvset消耗的存储区的大致(或精确)量。在实例中,所述指标集包含所述kvset的值块中的未经引用数据(例如,未经引用值)的量。此最后指标给出对可在维护操作中回收的空间的估计。在下文关于图4及5论述键块及值块的额外细节。

[0051] 在实例中,树100包含在至少一个机器可读媒体的第一计算机可读媒体中的第一根105及在至少一个计算机可读媒体的第二计算机可读媒体中的第二根110。在实例中,所述第二根为所述第一根的仅有子根。在实例中,所述第一计算机可读媒体为可字节寻址的且其中所述第二计算机可读为可块寻址的。此在图1中经图解说明,其中节点105在MEM树层级中以暗示其内存中位置,而节点110在L0处以暗示其在树100的根磁盘上元件中。

[0052] 上文论述证明KVS树100的各种组织属性。下文关于图7到25论述用以与树100相互作用的操作,例如树维护(例如,优化、无用单元收集等)、搜索等。在继续进行到这些主题之前,图2及3图解说明利用KVS树100的结构来实施多流存储装置的有效使用的技术。

[0053] 包括快闪存储器的存储装置或SSD可更高效地操作且在将具有类似寿命的数据分组在快闪擦除块中的情况下具有更大耐久性(例如,将不“耗损”)。包括其它非易失性媒体的存储装置还可受益于将具有类似寿命的数据分组,例如叠瓦式磁记录(SMR)硬盘驱动器(HDD)。在此上下文中,如果在相同时间或在相对小时间间隔内删除数据,那么数据具有类似寿命。用于删除存储装置上的数据的方法可包含对存储装置上的数据进行明确地解除分配、逻辑上覆写或物理上覆写。

[0054] 由于存储装置可能一般不知晓将存储于其内的各种数据的寿命,因此所述存储装置可为识别与数据相关联的逻辑寿命群组的数据存取命令(例如,读取或写入)提供接口。举例来说,工业标准SCSI及所提议NVMe存储装置接口规定写入命令,所述写入命令包括将

写入到存储装置的数据及与所述数据对应的称为流的寿命群组的数值流标识符(流ID)。支持多个流的存储装置为多流存储装置。

[0055] 温度为用以将数据分类的稳定性值,借此所述值对应于将在任一给定时间间隔中删除数据的相对概率。举例来说,可预期在一分钟内删除(或改变)HOT数据,同时可预期COLD数据持续一小时。在实例中,有限稳定性值集可用于规定此分类。在实例中,所述稳定性值集可为{热,温,冷},其中在给定时间间隔中,经分类为热的数据比经分类为温的数据具有更高的经删除概率,经分类为温的数据又比经分类为冷的数据具有更高的经删除概率。

[0056] 图2及3解决基于给定稳定性值以及关于一或多个KVS树的数据的一或多个属性而将不同流ID指派给不同写入。因此,继续先前实例,对于给定存储装置,第一流标识符集可与经分类为热的数据的写入命令一起使用,第二流标识符集可与经分类为温的数据的写入命令一起使用,且第三流标识符集可与经分类为冷的数据的写入命令一起使用,其中流标识符在这三个集中的至多一者中。

[0057] 为了便于论述图2及3的多流存储装置系统及技术而提供以下术语:

[0058] DID为存储装置的唯一装置标识符。

[0059] SID为给定存储装置上的流的流标识符。

[0060] TEMPSET为有限温度值集。

[0061] TEMP为TEMPSET的元素。

[0062] FID为KVS树集合的唯一森林标识符。

[0063] TID为KVS树的唯一树标识符。KVS树100具有TID。

[0064] LNUM为给定KVS树中的层级编号,其中为了方便,将KVS树的根节点视为在树层级0处,将根节点(如果存在)的子节点视为在树层级1处,依此类推。因此,如所图解说明,KVS树100包含树层级L0(包含节点110)到L3。

[0065] NNUM为给定KVS树中的给定层级处的给定节点的编号,其中为了方便,NNUM可为在范围零到(NodeCount(LNUM)-1)中的编号,其中NodeCount(LNUM)为树层级处的总节点数目LNUM,使得KVS树100中的每个节点由元组(LNUM,NNUM)唯一地识别。如图1中所图解说明,在节点110处开始且从顶部到底部、从左到右而进展的节点元组的完整列出将为:

[0066] L0(根):(0,0)

[0067] L1:(1,0)、(1,1)、(1,2)、(1,3)、(1,4)

[0068] L2:(2,0)、(2,1)、(2,2)、(2,3)

[0069] L3:(3,0)、(3,1)、(3,2)、(3,3)

[0070] KVSETID为唯一kvset标识符。

[0071] WTYPE为如下文所论述的值KBLOCK或VBLOCK。

[0072] WLAST为如下文所论述的布尔值(TRUE或FALSE)。

[0073] 图2是图解说明根据实施例的对多流存储装置(例如,装置260或265)的写入的实例的框图。图2图解说明多个KVS树,KVS树205及KVS树210。如所图解说明,每一树分别执行写入操作215及220。这些写入操作由存储子系统225处置。所述存储子系统可为例如用于装置260的装置驱动器,可为用以管理多个装置(例如,装置260及装置265)(例如存在于操作系统中的那些装置、网络附接存储装置等)的存储产品。存储子系统225将分别在操作250及

255中及时地完成对存储装置的写入。流映射电路230提供给定写入215的流ID以在装置写入250中使用。

[0074] 在KVS树205中,kvset的不可变性致使一次写入或删除全部kvset。因此,包括kvset的数据具有类似寿命。可使用例如擦除编码或RAID的技术将包括新kvset的数据写入到单个存储装置或写入到数个存储装置(例如,装置260及装置265)。此外,由于kvset的大小可大于任何给定装置写入250,因此写入kvset可涉及将多个写入命令引导到给定存储装置260。为促进流映射电路230的操作,可提供以下各项中的一或多者以用于针对每一此类写入命令250选择流ID:

[0075] A) 经写入的kvset的KVSETID;

[0076] B) 存储装置的DID;

[0077] C) KVS树所属的森林的FID;

[0078] D) KVS树的TID;

[0079] E) 含有kvset的KVS树中的节点的LNUM;

[0080] F) 含有kvset的KVS树中的节点的NNUM;

[0081] G) 如果写入命令是针对DID上的KVSETID的键块,那么WTYPE为KBLOCK,或如果写入命令是针对DID上的KVSETID的值块,那么WTYPE为VBLOCK

[0082] H) 如果写入命令对于DID上的KVSETID为最后的,那么WLAST为TRUE,且否则为FALSE

[0083] 在实例中,对于每一此类写入命令,称为流映射元组的元组(DID,FID,TID,LNUM,NNUM,KVSETID,WTYPE,WLAST)可发送到流映射电路230。流映射电路230接着可以存储子系统225的流ID将与写入命令250一起使用来做出响应。

[0084] 流映射电路230可包含电子硬件实施的控制器235、可存取流ID(A-SID)表240及选定流ID(S-SID)表245。控制器235经布置以接受流映射元组作为输入且以流ID做出响应。在实例中,控制器235经配置到存储多个KVS树205及210的多个存储装置260及265。控制器235经布置以获得(例如,通过配置、查询等)可存取装置的配置。控制器235还经布置以配置稳定性值集TEMPSET,且针对TEMPSET中的每一值TEMP而配置给定存储装置上的流的分数、数目或所述数目的其它决定性因子以供通过所述值分类的数据使用。

[0085] 在实例中,控制器235经布置以获得(例如,经由配置、消息等接收,从配置装置、固件等检索)温度指派方法。在此实例中,所述温度指派方法将用于将稳定性值指派给写入请求215。在实例中,流映射元组可包含DID、FID、TID、LNUM、NNUM、KVSETID、WTYPE或WLAST中的任何一或多者且用作由控制器235执行以从TEMPSET选择稳定性值TEMP的温度指派方法的输入。在实例中,KVS树范围为特定于写入KVS树组件(例如,kvset)的写入的参数集合。在实例中,所述KVS树范围包含FID、TID、LNUM、NNUM或KVSETID中的一或多者。因此,在此实例中,流映射元组可包含KVS树范围的组件以及装置特定或写入特定组件,例如DID、WLAST或WTYPE。在实例中,从流映射元组导出稳定性或温度范围元组TSCOPE。下文为可用于创建TSCOPE的实例性构成KVS树范围组件:

[0086] A) 经计算为(FID,TID,LNUM)的TSCOPE;

[0087] B) 经计算为(LNUM)的TSCOPE;

[0088] C) 经计算为(TID)的TSCOPE;

[0089] D) 经计算为(TID,LNUM)的TSCOPE;或

[0090] E) 经计算为(TID,LNUM,NNUM)的TSCOPE。

[0091] 在实例中,控制器235可实施静态温度指派方法。所述静态温度指派方法可(举例来说)从配置文件、数据库、KVS树元数据或者KVS树105TID或其它数据库中的元数据(包含存储于KVS树TID中的元数据)读取选定TEMP。在此实例中,这些数据源包含从TSCOPE到稳定性值的映射。在实例中,可对映射进行高速缓存(例如,基于控制器235的激活或在稍后操作期间动态地)以在写入请求到达时加速稳定性值的指派。

[0092] 在实例中,控制器235可实施动态温度指派方法。所述动态温度指派方法可基于将kvset写入到TSCOPE的频率而计算选定TEMP。举例来说,控制器235针对给定TSCOPE执行温度指派方法的频率可经测量且在TEMPSET中的TEMPS周围群集化。因此,此计算可(举例来说)定义频率范围集及从每一频率范围到稳定性值的映射,使得TEMP的值由含有将kvset写入到TSCOPE的频率的频率范围来确定。

[0093] 控制器235经布置以获得(例如,经由配置、消息等接收,从配置装置、固件等检索)流指派方法。所述流指派方法将消耗写入215的KVS树205方面以及稳定性值(例如,来自温度指派)以产生流ID。在实例中,控制器235可在流指派方法中使用流映射元组(例如,包含KVS树范围)来选择流ID。在实例中,可在由控制器235执行以选择流ID的流指派方法中使用DID、FID、TID、LNUM、NNUM、KVSETID、WTYPE或WLAST中的任一或多者以及稳定性值。在实例中,从流映射元组导出流范围元组SSCOPE。下文为可用于创建SSCOPE的实例性构成KVS树范围组件:

[0094] A) 经计算为(FID,TID,LNUM,NNUM)的SSCOPE

[0095] B) 经计算为(KVSETID)的SSCOPE

[0096] C) 经计算为(TID)的SSCOPE

[0097] D) 经计算为(TID,LNUM)的SSCOPE

[0098] E) 经计算为(TID,LNUM,NNUM)的SSCOPE

[0099] F) 经计算为(LNUM)的SSCOPE

[0100] 控制器235可经布置以在接受输入之前初始化A-SID表240及S-SID表245。A-SID表240为可针对元组(DID,TEMP,SID)而存储条目且可检索具有DID及TEMP的规定值的此类条目的数据结构(表、字典等)。记号A-SID(DID,TEMP)是指A-SID表240(如果存在)中具有DID及TEMP的规定值的所有条目。在实例中,A-SID表240可针对每一经配置存储装置260及265以及TEMPSET中的温度值经初始化。A-SID表240初始化可如下继续进行:对于每一经配置存储装置DID,控制器235可经布置以:

[0101] A) 获得DID上可用的流数目,称为SCOUNT;

[0102] B) 针对DID上的SCOUNT流中的每一者获得唯一SID;且

[0103] C) 针对TEMPSET中的每一值TEMP:

[0104] a) 根据TEMP的经配置决定性因子计算多少SCOUNT流将用于通过TEMP分类的数据,称为TCOUNT;且

[0105] b) 选择尚未输入于A-SID表240中的DID的TCOUNT SID,且针对DID的每一选定TCOUNT SID,在A-SID表240中创建针对(DID,TEMP,SID)的一个条目(例如,行)。

[0106] 因此,一旦经初始化,A-SID表240便针对每一经配置存储装置DID及TEMPSET中的

值TEMP包含经指派唯一SID的条目。用于获得对于经配置存储装置260可用的流数目及每一流的可用SID的技术因存储装置接口而不同,然而,这些流是经由多流存储装置的接口可容易地存取的。

[0107] S-SID表245维持已在使用中(例如,已成为给定写入的一部分)的流记录。S-SID表245为可存储元组(DID,TEMP,SSCOPE,SID,时间戳)的条目且可检索或删除具有DID、TEMP及任选地SSCOPE的规定值的此类条目的数据结构(表、字典等)。记号S-SID(DID,TEMP)是指S-SID表245(如果存在)中具有DID及TEMP的规定值的所有条目。与A-SID表240一样,S-SID表245可由控制器235初始化。在实例中,控制器235经布置以针对每一经配置存储装置260及265以及TEMPSET中的温度值初始化S-SID表245。

[0108] 如上文所述,S-SID表245中的条目表示用于写入操作的当前或已经经指派流。因此,一般来说,S-SID表245在初始化之后为空的,条目在指派流ID时由控制器235创建。

[0109] 在实例中,控制器235可实施静态流指派方法。所述静态流指派方法针对给定DID、TEMP及SSCOPE选择相同流ID。在实例中,所述静态流指派方法可确定S-SID(DID,TEMP)是否具有针对SSCOPE的条目。如果不存在符合条目,那么所述静态流指派方法从A-SID(DID,TEMP)选择流ID SID且在S-SID表245中创建针对(DID,TEMP,SSCOPE,SID,时间戳)的条目,其中时间戳为在选择之后的当前时间。在实例中,从A-SID(DID,TEMP)的选择为随机的,或为循环过程的结果。一旦找到或创建来自S-SID表245的条目,便将流ID SID传回到存储子系统225。在实例中,如果WLAST为真,那么删除S-SID表245中针对(DID,TEMP,SSCOPE)的条目。此最后实例证明使WLAST发信号通知原本树205已知但存储子系统225不知的kvset等的写入215的完成的有用性。

[0110] 在实例中,控制器235可实施最近最少使用(LRU)的流指派方法。所述LRU流指派方法在相对小时间间隔内针对给定DID、TEMP及SSCOPE选择相同流ID。在实例中,所述LRU指派方法确定S-SID(DID,TEMP)是否具有针对SSCOPE的条目。如果存在所述条目,那么所述LRU指派方法选择此条目中的流ID且将S-SID表245中的此条目中的时间戳设定到当前时间。

[0111] 如果SSCOPE条目不在S-SID(DID,TEMP)中,那么所述LRU流指派方法确定条目S-SID(DID,TEMP)数目是否等于条目A-SID(DID,TEMP)数目。如果此为真,那么所述LRU指派方法从S-SID(DID,TEMP)中具有最旧时间戳的条目选择流ID SID。在此处,用新条目(DID,TEMP,SSCOPE,SID,时间戳)替换S-SID表245中的条目,其中时间戳为在选择之后的当前时间。

[0112] 如果存在少于A-SID(DID,TEMP)条目的S-SSID(DID,TEMP)条目,那么所述方法从A-SID(DID,TEMP)选择流ID SID,使得S-SID(DID,TEMP)中不存在具有选定流ID的条目且在S-SID表245中创建针对(DID,TEMP,SSCOPE,SID,时间戳)的条目,其中时间戳为在选择之后的当前时间。

[0113] 一旦找到或创建来自S-SID表245的条目,便将流ID SID传回到存储子系统225。在实例中,如果WLAST为真,那么删除S-SID表245中针对(DID,TEMP,SSCOPE)的条目。

[0114] 在操作中,控制器235经配置以为经接收作为写入请求215的一部分的给定流映射元组指派稳定性值。一旦确定所述稳定性值,控制器235便经布置以指派SID。温度指派方法及流指派方法可各自引用且更新A-SID表240及S-SID表245。在实例中,控制器235还经布置以将SID提供给请求者,例如存储子系统225。

[0115] 基于KVS树范围而使用流ID准许相似数据共置在多流存储装置260上的擦除块270中。此减少装置上的无用单元收集且因此可增加装置性能及寿命。此益处可扩展到多个KVS树。KVS树可用于森林或小树林中，借此使用数个KVS树来实施单个结构，例如文件系统。举例来说，一个KVS树可使用块编号作为键且使用块中的位作为值，而第二KVS树可使用文件路径作为键且使用块编号列表作为所述值。在此实例中，由路径引用的给定文件的kvset与保存块编号的kvset很可能具有类似寿命。因此上文FID的包含。

[0116] 上文所描述的结构及技术提供实施KVS树及例如快闪存储装置的存储装置的系统中的若干个优点。在实例中，实施存储于一或多个存储装置上的数个KVS树的计算系统可使用KVS树的知识来更高效地选择多流存储装置中的流。举例来说，所述系统可经配置使得对KVS树执行的同步写入操作（例如，引入或压缩）数目是基于任一给定存储装置上为温度分类（指派给通过这些写入操作而写入的kvset数据）预留的流的数目而限定的。此为可能的，因为在kvset内，所述数据的寿命预期在kvset全部经写入及删除时为相同的。如别处所述，可将键与值分隔。因此，当执行下文所论述的键压缩时，键写入将具有可能比值寿命短的相同寿命。另外，树层级在实验上似乎为数据寿命、较旧数据及因此较大（例如，较深）树层级（具有比较高树层级处的较年轻数据长的寿命）的强烈指示。

[0117] 以下情景可进一步阐明流映射电路230限定写入的操作，考虑到：

[0118] A) 温度值{热,冷}，其中给定存储装置上的H流用于经分类为热的数据，且给定存储装置上的C流用于经分类为冷的数据。

[0119] B) 配置有经计算为(LNUM)的TSCOPE的温度指派方法，借此给写入到任一KVS树中的L0的数据指派热的温度值，且给写入到任一KVS树中的L1或更大层级的数据指派冷的温度值。

[0120] C) 配置有经计算为(TID,LNUM)的SSCOPE的LRU流指派方法。

[0121] 在此情形中，所有KVS树的同步引入及压缩操作（产生写入的操作）的总数目遵循这些条件：所有KVS树的同步引入操作至多为H，因为用于所有引入操作的数据经写入到KVS树中的层级0且因此将经分类为热，且所有KVS树的同步压缩操作至多为C，因为用于所有溢出压缩及大多数其它压缩操作的数据经写入到层级1或更大层级且因此将经分类为冷。

[0122] 其它此类限定为可能的且可取决于KVS树及控制器235的特定实施细节而为有利的。举例来说，给定如上文而配置的控制器的235，引入操作数目为H的分数（例如，二分之一）及压缩操作数目为C的分数（例如，四分之三）可为有利的，因为在SSCOPE经计算为(TID,LNUM)的情况下的LRU流指派可不利用流映射元组中的WLAST来在接收到TID中的给定KVSET的最后写入后即刻移除不需要S-SID表245条目，从而产生次优SID选择。

[0123] 尽管上文在KVS树的上下文中描述流映射电路230的操作，但例如LSM树实施方案的其它结构可同样地受益于本文中所呈现的概念。许多LSM树变体存储键值对与逻辑删除的集合，借此给定集合可通过引入操作或无用单元收集操作（通常称为压缩或合并操作）来创建，且接着稍后由于后续引入操作或无用单元收集操作而全部被删除。因此，与包括KVS树中的kvset的数据一样，包括此集合的数据具有类似寿命。因此，类似于上文的流映射元组的元组可针对大多数其它LSM树变体经定义，其中可由通过给定LSM树变体中的引入操作或无用单元收集操作创建的键值对或逻辑删除集合的唯一标识符替换KVSETID。流映射电路230接着可如所描述而用于为存储包括键值对与逻辑删除的此集合的数据的所述多个写

入命令选择流标识符。

[0124] 图3图解说明根据实施例的用以促进对多流存储装置进行写入的方法300的实例。方法300的操作利用例如在本申请案全篇(包含在下文关于图26)所描述的电子硬件(例如,电路)来实施。方法300提供若干个实例以实施上文关于图2的论述。

[0125] 在操作305处,接收对多流存储装置的KVS树写入请求的通知。在实例中,所述通知包含与所述写入请求中的数据对应的KVS树范围。在实例中,所述KVS树范围包含以下各项中的至少一者:与所述数据的kvset对应的kvset ID;与对应于所述数据的KVS树的节点对应的节点ID;与对应于所述数据的树层级对应的层级ID;所述KVS树的树ID;与所述KVS树所属的森林对应的森林ID;或与所述数据对应的类型。在实例中,所述类型为键块类型或值块类型。

[0126] 在实例中,所述通知包含所述多流装置的装置ID。在实例中,所述通知包含与将由所述kvset ID识别的kvset写入到所述多流存储装置的写入请求序列中的最后写入请求对应的WLAST标志。

[0127] 在操作310处,基于所述KVS树范围及所述写入请求的稳定性值而将流标识符(ID)指派给所述写入请求。在实例中,指派所述稳定性值包含:针对与树层级对应的层级ID维持稳定性值指派的频率集,所述频率集的每一成员对应于唯一层级ID;从所述频率集检索与所述KVS树范围中的层级ID对应的频率;及基于所述频率而从稳定性值与频率范围的映射选择稳定性值。

[0128] 在实例中,基于所述KVS树范围及所述写入请求的所述稳定性值而将所述流ID指派给所述写入请求包含依据所述KVS树范围创建流范围值。在实例中,所述流范围值包含所述数据的层级ID。在实例中,所述流范围值包含所述数据的树ID。在实例中,所述流范围值包含所述数据的层级ID。在实例中,所述流范围值包含所述数据的节点ID。在实例中,所述流范围值包含所述数据的kvset ID。

[0129] 在实例中,基于所述KVS树范围及所述写入请求的所述稳定性值而将所述流ID指派给所述写入请求还包含使用所述流范围值在选定流数据结构中执行查找。在实例中,在所述选定流数据结构中执行所述查找包含:未能在所述选定流数据结构中找到所述流范围值;使用所述稳定性值对可用流数据结构执行查找;接收包含流ID的所述查找的结果;及将条目添加到所述选定流数据结构,所述条目包含所述流ID、所述流范围值及在添加所述条目时的时间的戳。在实例中,所述可用流数据结构的多个条目对应于所述稳定性值,且其中所述查找的所述结果为来自所述多个条目的条目的循环或随机选择中的至少一者。在实例中,所述可用流数据结构可通过以下方式而初始化:获得可从所述多流存储装置得到的多个流;获得可从所述多流存储装置得到的所有流的流ID,每一流ID为唯一的;将流ID添加到稳定性值群组;及在所述可用流数据结构中创建针对每一流ID的记录,所述记录包含所述流ID、所述多流存储装置的装置ID及与所述流ID的稳定性值群组对应的稳定性值。

[0130] 在实例中,在所述选定流数据结构中执行所述查找包含:未能在所述选定流数据结构中找到所述流范围值;基于所述选定流数据结构的内容而从所述选定流数据结构或可用流数据结构定位流ID;及将条目创建到所述选定流数据结构,所述条目包含所述流ID、所述流范围值及在添加所述条目时的时间的戳。在实例中,基于所述选定流数据结构的内容而从所述选定流数据结构或可用流数据结构定位所述流ID包含:将来自所述选定流数

据结构的第一条目数目与来自所述可用流数据结构的第二条目数目进行比较,以确定所述第一条目数目与所述第二条目数目相等;从所述选定流数据结构定位与所述稳定性值对应的条目群组;及传回所述条目群组中具有最旧时间戳的条目的流ID。在实例中,基于所述选定流数据结构的内容而从所述选定流数据结构或可用流数据结构定位所述流ID包含:将来自所述选定流数据结构的第一条目数目与来自所述可用流数据结构的第二条目数目进行比较以确定所述第一条目数目与所述第二条目数目不相等;使用所述选定流数据结构的条目中的所述稳定性值及流ID对所述可用流数据结构执行查找;接收包含未在所述选定流数据结构的所述条目中的流ID的所述查找的结果;及将条目添加到所述选定流数据结构,所述条目包含所述流ID、所述流范围值及在添加所述条目时的时间的的时间戳。

[0131] 在实例中,基于所述KVS树范围及所述写入请求的所述稳定性值而将所述流ID指派给所述写入请求还包含从所述选定流数据结构传回与所述流范围对应的流ID。在实例中,从所述选定流数据结构传回与所述流范围对应的所述流ID包含更新所述选定流数据结构中与所述流ID对应的条目的时间戳。在实例中,所述写入请求包含WLAST标志,且其中从所述选定流数据结构传回与所述流范围对应的所述流ID包含从所述选定流数据结构移除与所述流ID对应的条目。

[0132] 在实例中,方法300可经扩展以包含从所述选定流数据结构移除具有超过阈值的时间戳的条目。

[0133] 在操作315处,传回所述流ID以管理对所述写入请求的流指派,其中所述流指派修改所述多流存储装置的写入操作。

[0134] 在实例中,方法300可任选地经扩展以包含基于所述KVS树范围而指派所述稳定性值。在实例中,所述稳定性值为预定义稳定性值集中的一者。在实例中,所述预定义稳定性值集包含HOT、WARM及COLD,其中HOT指示所述多流存储装置上的所述数据的最低预期寿命且COLD指示所述多流存储装置上的所述数据的最高预期寿命。

[0135] 在实例中,指派所述稳定性值包含使用所述KVS树范围的一部分从数据结构定位所述稳定性值。在实例中,所述KVS树范围的所述部分包含所述数据的层级ID。在实例中,所述KVS树范围的所述部分包含所述数据的类型。

[0136] 在实例中,所述KVS树范围的所述部分包含所述数据的树ID。在实例中,所述KVS树范围的所述部分包含所述数据的层级ID。在实例中,所述KVS树范围的所述部分包含所述数据的节点ID。

[0137] 图4是根据实施例的图解说明用于键及值的存储组织的实例的框图。Kvset可使用用以保存键(视需要以及逻辑删除)的键块及用以保存值的值块来存储。针对给定kvset,所述键块还可含有指标及其它信息(例如布隆过滤器)以用于高效地定位单个键、定位键范围、或产生所述kvset中的所有键(包含键逻辑删除)的总定序,且用于获得与所述键(如果存在)相关联的值。

[0138] 在图4中表示单个kvset。所述键块包含主要键块410(包含标头405)及扩展键块415(包含扩展标头417)。所述值块分别包含标头420及440以及值425、430、435及445。第二值块还包含自由空间450。

[0139] Kvset的树表示经图解说明为横跨键块410及415。在此图解说明中,叶节点含有对值425、430、435及445的值引用(VID)及具有逻辑删除的两个键。此图解说明:在实例中,逻

辑删除不具有在值块中的对应值,即使其可称为某一类型的键值对。

[0140] 对所述值块的图解说明证明每一值块可具有标头及在不进行划定的情况下彼此相邻的值。对例如值425的值的值块中的特定位置的引用一般(举例来说)以偏移且扩展格式存储于对应键条目中。

[0141] 图5是根据实施例的图解说明键块及值块的配置的实例的框图。图5的键块与值块组织图解说明扩展键块及值块的一般简单本质。具体来说,每一者一般为简单存储容器,所述简单存储容器具有用以识别其类型(例如,键块或值块)的标头及可能大小、在存储区上的位置或其它元数据。在实例中,值块包含指示其为值块的具有幻数的标头540及用以存储值位的存储区545。键扩展块包含指示其为扩展块的标头525且存储键结构的一部分530,例如KB树、B树等。

[0142] 主要键块除简单地存储键结构之外也为许多kvset元数据提供位置。主要键块包含键结构的根520。主要键块还可包含标头505、布隆过滤器510或键结构的一部分515。

[0143] 对主要键块的组件的引用包含在标头505、例如布隆过滤器510的块或根节点520中。例如kvset大小、值块地址、压缩性能或使用的指标也可含纳在标头505中。

[0144] 布隆过滤器510在创建kvset时经计算且提供就绪机制以在不对键结构执行搜索的情况下确定键是否不在kvset中。此进步允许如下文所述的扫描操作的更大效率。

[0145] 图6图解说明根据实施例的KB树600的实例。将在kvset的键块中使用的实例性键结构为KB树。KB树600具有与B+树的结构类似性。在实例中,KB树600具有4096字节节点(例如,节点605、610及615)。KB树的所有键驻存于叶节点(例如,节点615)中。内部节点(例如,节点610)具有选定叶节点键的副本以导航树600。键查找的结果为值引用,其可为(在实例中)针对值块ID、偏移及长度。

[0146] KB树600具有以下性质:

[0147] A) 生根于边缘键K的子节点处的子树中的所有键小于或等于K。

[0148] B) 任一树或子树中的最大键为最右叶节点中的最右条目。

[0149] C) 给定具有指向子节点R的最右边缘的节点N,生根于节点R处的子树中的所有键大于节点N中的所有键。

[0150] 可经由二进制搜索在根节点605中的键当中搜索KB树600以找到适当“边缘”键。可遵循到边缘键的子节点的链接。接着重复此过程直到在叶节点615中找到匹配或找不到匹配为止。

[0151] 由于kvset一旦经创建且不改变,因此创建KB树600可不同于随时间而变的其它树结构。可以自下而上方式创建KB树600。在实例中,首先创建叶节点615,后续接着其母节点610,依此类推,直到留下一个节点(根节点605)为止。在实例中,创建以单个空叶节点(当前节点)开始。将每一新键添加到当前节点。在当前节点变满时,创建新叶节点且其成为当前节点。当添加最后键时,所有叶节点为完整的。此时,使用来自每一叶节点的最大键作为输入流,以类似方式创建下一向上层级处的节点(即,叶节点的母节点)。当耗尽所述键时,所述层级为完整的。重复此过程直到最近创建的层级由单个节点(根节点605)组成为止。

[0152] 如果在创建期间当前键块变满,那么可将新节点写入到扩展键块。在实例中,从第一键块跨越到第二键块的边缘包含对第二键块的引用。

[0153] 图7是根据实施例的图解说明KVS树引入的框图。在KVS树中,将新kvset写入到根

节点730的过程称为引入。键值对705(包含逻辑删除)积累在所述KVS树的存储器710中,且经组织到从最新kvset 715到最旧kvset 720而定序的若干kvset中。在实例中,kvset 715可为可变的以同步地接受键值对。此为所述KVS树中的仅有可变kvset变体。

[0154] 引入725将主要存储器710中的最旧kvset 720中的键值对及逻辑删除写入到所述KVS树的根节点730中的新(且最新)kvset 735,且接着从主要存储器710删除所述kvset 720。

[0155] 图8图解说明根据实施例的用于KVS树引入的方法800的实例。方法800的操作利用例如在本申请案全篇(包含在下文关于图26)所描述的电子硬件(例如,电路)来实施。

[0156] 在操作805处,接收键值集(kvset)以存储于键值数据结构中。在此处,所述键值数据结构经组织为树且所述kvset包含唯一键到值的映射。所述kvset的键及值为不可变的且所述树的节点具有时间上经定序的kvset序列。

[0157] 在实例中,当将kvset写入到至少一个存储媒体时,所述kvset为不可变的。在实例中,其中所述kvset的键条目存储于包含主要键块及零个或更多个扩展键块的键块集中。在此处,所述键块集的成员对应于至少一个存储媒体的媒体块,其中每一键块包含用以将其识别为键块的标头。

[0158] 在实例中,所述主要键块包含所述kvset的所述一或多个扩展键块的媒体块标识列表。在实例中,所述主要键块包含所述值块集中的值块的媒体块标识列表。在实例中,所述主要键块包含所述kvset的键树中的最低键的副本,所述最低键通过所述树的预设定排序次序来确定。在实例中,所述主要键块包含所述kvset的键树中的最高键的副本,所述最高键通过所述树的预设定排序次序来确定。在实例中,所述主要键块包含所述kvset的键树的标头。在实例中,所述主要键块包含所述kvset的键树的媒体块标识列表。在实例中,所述主要键块包含所述kvset的布隆过滤器的布隆过滤器标头。在实例中,所述主要键块包含所述kvset的布隆过滤器的媒体块标识列表。

[0159] 在实例中,将值存储于值块集中,操作805。在此处,所述值块集的成员对应于所述至少一个存储媒体的媒体块,其中每一值块包含用以将其识别为值块的标头。在实例中,值块包含在值之间不具有间隔的一或多个值的存储区段。

[0160] 在实例中,所述主要键块包含所述kvset的指标集。在实例中,所述指标集包含存储于所述kvset中的键的总数目。在实例中,所述指标集包含存储于所述kvset中的具有逻辑删除值的键的数目。在实例中,所述指标集包含存储于所述kvset中的键的所有键长度的和。在实例中,所述指标集包含存储于所述kvset中的键的所有值长度的和。在实例中,所述指标集包含所述kvset的值块中的未经引用数据的量。

[0161] 在操作810处,将所述kvset写入到所述树的根节点的kvset序列。

[0162] 方法800可经扩展以包含操作815到825。

[0163] 在操作815处,接收将存储于所述键值数据结构中的键及对应值。

[0164] 在操作820处,将所述键及所述值放置在初步kvset中,所述初步kvset为可变的。在实例中,写入到所述初步根节点的速率超过阈值。在此实例中,可扩展方法800以抑制对所述键值数据结构的写入请求。

[0165] 在操作825处,在达到指标时将所述kvset写入到所述键值数据结构。在实例中,所述指标为初步根节点的大小。在实例中,所述指标为经过时间。

[0166] 一旦已发生引入,便可采用各种维护操作来维护KVS树。举例来说,如果键在一个时间以第一值写入且在稍后时间以第二值写入,那么移除第一键值对将释放空间或减少搜索时间。为解决这些问题中的一些问题,KVS树可使用压缩。下文关于图9到18论述数个压缩操作的细节。所图解说明压缩操作为无用单元收集形式,因为其可在合并期间移除过时数据,例如键或键值对。

[0167] 压缩发生在各种触发条件下,例如当节点中的kvset满足经规定或经计算准则时。此压缩准则的实例包含所述kvset的总大小或所述kvset中的无用单元量。Kvset中的无用单元的一个实例为(举例来说)因较新kvset中的键值对或逻辑删除或者已违反生存时间约束的键值对以及其它原因而变过时的一个kvset中的键值对或逻辑删除。Kvset中的无用单元的另一实例为由键压缩产生的值块中的未经引用数据(未经引用值)。

[0168] 一般来说,压缩操作的输入为在满足压缩准则时节点中的kvset中的一些或所有kvset。这些kvset称为合并集且包括两个或两个以上kvset的时间上连续序列。

[0169] 由于一般在引入新数据时触发压缩,因此可扩展方法800以支持压缩,然而,还可在(举例来说)存在自由处理资源或其它便利情景时触发以下操作以执行维护。

[0170] 因此,可压缩KVS树。在实例中,响应于触发而执行压缩。在实例中,所述触发为时间周期的到期。

[0171] 在实例中,所述触发为节点的指标。在实例中,所述指标为节点的kvset的总大小。在实例中,所述指标为节点的kvset数目。在实例中,所述指标为节点的未经引用值的总大小。在实例中,所述指标为未经引用值的数目。

[0172] 图9是根据实施例的图解说明键压缩的框图。键压缩从合并集读取键及逻辑删除而非值,移除所有过时键或逻辑删除,将所得键及逻辑删除写入到一或多个新kvset中(例如,通过写入到新键块中),从节点删除键存储区而非值。所述新kvset在内容方面且在节点中的kvset从最新到最旧的逻辑定序内的放置方面原子地替换且逻辑上等效于合并集。

[0173] 如所图解说明,kvset KVS3(最新)、KVS2及KVS1(最旧)经历节点的键压缩。当合并这些kvset的键存储区时,发生键A及B上的冲突。由于新kvset(KVS4(下文所图解说明))可仅含有每一经合并键的一者,因此解决所述冲突以支持最近(如所图解说明的最左)键,从而分别参考键A及B的值ID 10及值ID 11。键C不具有冲突且因此将包含于新kvset中。因此,将为新kvset(KVS4)的一部分的键条目在顶部节点中经加阴影。

[0174] 出于说明性目的,KVS4经绘制以横跨节点中的KVS1、KVS2及KVS3,且值条目经绘制于节点中的类似位置中。这些位置的目的证明在键压缩中不改变值,而是仅改变键。如下文所阐释,此通过减少在任一给定节点中搜索的kvset数目而提供更高效搜索且还可提供用以指导维护操作的有价值见解。还注意,以虚线来图解说明值20及30,从而表示其存留于节点中但不再由键条目引用,因为在压缩中移除了其相应键条目。

[0175] 当在压缩期间可将新kvset(例如,KVS5)放置在KVS3或KVS4的最新位置中(例如,在左边)时键压缩为非阻断的,因为按照定义,所添加kvset将逻辑上比由键压缩产生的kvset(例如,KVS4)新。

[0176] 图10图解说明根据实施例的用于键压缩的方法1000的实例。方法1000的操作利用例如在本申请案全篇(包含在下文关于图26)所描述的电子硬件(例如,电路)来实施。

[0177] 在操作1005处,选择来自节点的kvset序列的kvset子集。在实例中,所述kvset子

集为连续kvset且包含最旧kvset。

[0178] 在操作1010处,定位冲突键集。所述冲突键集的成员包含所述节点的所述kvset序列中的至少两个kvset中的键条目。

[0179] 在操作1015处,将所述冲突键集的每一成员的最近键条目添加到新kvset。在其中所述节点不具有子节点且其中所述kvset子集包含最旧kvset的实例中,将所述冲突键集的每一成员的所述最近键条目写入到所述新kvset且将未在所述冲突键集中的所述kvset子集的成员中的每一键的条目写入到所述新kvset包含省略包含逻辑删除的任何键条目。在其中所述节点不具有子节点且其中所述kvset子集包含最旧kvset的实例中,将所述冲突键集的每一成员的所述最近键条目写入到所述新kvset且将未在所述冲突键集中的所述kvset子集的成员中的每一键的条目写入到所述新kvset包含省略到期的任何键条目。

[0180] 在操作1020处,将未在所述冲突键集中的所述kvset子集的成员中的每一键的条目添加到所述新kvset。在实例中,操作1020及1015可同时操作以将条目添加到所述新kvset。

[0181] 在操作1025处,通过写入所述新kvset且移除(例如,删除、加删除标记等)所述kvset子集而用所述新kvset替换所述kvset子集。

[0182] 图11是根据实施例的图解说明键值压缩的框图。键值压缩在其值处理方面不同于键压缩。键值压缩从合并集读取键值对及逻辑删除,移除过时键值对或逻辑删除,将所得键值对及逻辑删除写入到同一节点中的一或多个新kvset,且从所述节点删除包括所述合并集的所述kvset。所述新kvset在内容方面且在节点中的kvset从最新到最旧的逻辑定序内的放置方面原子地替换且逻辑上等效于合并集。

[0183] 如所图解说明,kvset KVS3、KVS2及KVS1包括合并集。经加阴影键条目及值将保持在合并中且放置在新KVS4中,将新KVS4写入到节点以替换KVS3、KVS2及KVS1。再次,如上文关于键压缩所图解说明,解决键A及B的键冲突以支持最近条目。键值压缩与键压缩的不同之处在于未经引用值的移除。因此,在此处,KVS4经图解说明以仅消耗保存其当前键及值所需要的空间。

[0184] 在实务上,举例来说,当键及值单独地存储于键块及值块中时,KVS4包含新键块(与键压缩的结果相同)及新值块(与键压缩的结果不同)两者。然而,再次,当正执行键值压缩时键值压缩不阻止将额外kvset写入到节点,因为所添加kvset将逻辑上比KVS4(键值压缩的结果)新。因此,在节点的最旧位置中(例如,在右边)图解说明KVS4。

[0185] 图12图解说明根据实施例的用于键值压缩的方法1200的实例。方法1200的操作利用例如在本申请案全篇(包含在下文关于图26)所描述的电子硬件(例如,电路)来实施。

[0186] 在操作1205处,选择来自节点的kvset序列的kvset子集(例如,合并集)。在实例中,所述kvset子集为连续kvset且包含最旧kvset。

[0187] 在操作1210处,定位冲突键集。所述冲突键集的成员包含所述节点的所述kvset序列中的至少两个kvset中的键条目。

[0188] 在操作1215处,将所述冲突键集的每一成员的最近键条目及对应值添加到新kvset。在其中所述节点不具有子节点且其中合并集含有最旧kvset的实例中,将所述冲突键集的每一成员的所述最近键条目写入到所述新kvset且将未在所述冲突键集中的所述kvset子集的成员中的每一键的条目写入到所述新kvset包含省略包含逻辑删除的任何键

条目。在其中所述节点不具有子节点且其中合并集含有最旧kvset的实例中,将所述冲突键集的每一成员的所述最近键条目写入到所述新kvset且将未在所述冲突键集中的所述kvset子集的成员中的每一键的条目写入到所述新kvset包含省略到期的任何键条目。

[0189] 在操作1220处,将未在所述冲突键集中的所述kvset子集的成员中的每一键的条目及值添加到所述新kvset。

[0190] 在操作1225处,通过写入所述新kvset(例如,写入到存储区)且移除所述kvset子集而用所述新kvset替换所述kvset子集。

[0191] 下文关于图15到18所论述的溢出及提升压缩为一种形式的键值压缩,其中合成kvset分别放置在子节点或母节点中。由于每一者遍历树且KVS树实行母节点与子节点之间的确定性映射,因此在论述这些其它压缩操作之前在此处呈现此确定性映射的简要论述。

[0192] 图13图解说明根据实施例的溢出值及其与树的关系的实例。所述确定性映射确保:给定键,可在不考虑KVS树的内容的情况下知晓键值对将映射到哪一子节点。溢出函数接受键且产生与KVS树的确定性映射对应的溢出值。在实例中,所述溢出函数接受键及当前树层级两者且产生所述树层级处的键的母节点或子节点所特有的溢出值。

[0193] 通过阐释方式,简单确定性映射(图13中未图解说明)可包含(举例来说)按字母顺序的映射,其中对于由字母表字符构成的键,每一树层级针对字母表的每一字母包含一个子节点,且映射又使用所述键的字符;例如第一字符确定L1子节点,第二字符确定L2子节点,依此类推。虽然简单且满足KVS树的确定性映射,但本技术在某种程度上受到刚度、树中的不良平衡及缺乏对树扇的控制的影响。

[0194] 更佳技术为对键执行散列且为每一树层级映射指定散列的若干部分。此确保键在其遍历树时均匀地散布(采用充足散列技术)且通过针对任一给定树层级选择散列部分的大小而控制扇出。此外,由于散列技术一般允许配置合成散列的大小,因此可确保充足数目个位(举例来说),从而避免关于上文所论述的简单技术的问题,其中短字(例如“所述”)仅具有足以用于三层级树的字符。

[0195] 图13图解说明具有分别与树的L1、L2及L3对应的部分1305、1310及1315的键散列的结果。关于给定树散列,树的遍历沿着虚线及节点继续前进。具体来说,在根节点1320处开始,部分1305将遍历引导到节点1325。接下来,部分1310将遍历引导到节点1330。当部分1315指向在树的最深层级处的节点1335时遍历完成,此基于所图解说明键散列的大小及分摊而为可能的。

[0196] 在实例中,对于给定键K,键K(或键K的子键)的散列称为键K的溢出值。应注意,两个不同键可具有相同溢出值。当采用子键来产生溢出值时,出现此情况以达成如下文所论述的前缀扫描或逻辑删除通常为合意的。

[0197] 在实例中,对于给定KVS树,给定键K的溢出值为常数,且溢出值的二进制表示包括B个位。在此实例中,溢出值中的B个位经编号为0到(B-1)。而且在此实例中,KVS树经配置使得树层级L处的节点全部具有相同数目个子节点,且此子节点数目为大于或等于2的2的整数幂。在此配置中,可如下文所图解说明而使用用于键分配的键K的溢出值的位。

[0198] 对于在KVS树中的层级L处的节点,使 $2^E(L)$ 为经配置以用于所述节点的子节点数目,其中 $2^E(L) \geq 2$ 。接着,对于KVS树中的给定节点及给定键K,键K的溢出值如下规定用于溢出压缩的节点的子节点:

[0199] A) 层级0:溢出值位0到 $(E(0) - 1)$ 规定键K的子节点数目;

[0200] B) 层级1:溢出值位 $E(0)$ 到 $(E(0) + E(1) - 1)$ 规定键K的子节点数目;且

[0201] C) 层级L ($L > 1$):溢出值位 $\text{sum}(E(0), \dots, E(L-1))$ 到 $(\text{sum}(E(0), \dots, E(L)) - 1)$ 规定键K的子节点数目。

[0202] 下文的表图解说明在给定具有七个 (7) 层级、键K及键K的16位溢出值的KVS树的情况下以上基于根数的键分配技术的特定实例:

[0203]

层级	0	1	2	3	4	5
子节点计数	2	8	4	16	32	2
溢出值位	0	1-3	4-5	6-9	10-14	15
键K溢出值	0	110	01	1110	10001	1
所选择的子节点	0	6	1	14	17	1

[0204] 其中层级为KVS树中的层级编号;子节点计数为经配置以用于规定层级处的所有节点的子节点数目;溢出值位为溢出压缩针对规定层级处的键分配所使用的溢出值位数目;键K溢出值为给定键K的给定16位溢出值的二进制表示,具体来说0110011110100011—为了清晰,将溢出值分段成溢出压缩针对规定层级处的键分配所使用的位;且所选择的子节点为溢出压缩针对具有给定溢出值的任何(非过时)键值对或逻辑删除而选择的子节点编号—此包含具有给定键K的所有(非过时)键值对或逻辑删除,以及不同于键K的可具有相同溢出值的其它键。

[0205] 在实例中,对于给定KVS树,溢出值计算及溢出值大小(以位为单位)对于所有键可为相同的。如上文所述,使用充足散列准许控制溢出值中的位数同时(举例来说)确保足以容纳所要数目个树层级及每一层级处的节点的要数目个子节点的溢出值大小。在实例中,对于给定KVS树,键K的溢出值可视需要而计算或存储于存储媒体上(例如,经高速缓存)。

[0206] 图14图解说明根据实施例的用于溢出值函数的方法1400的实例。方法1400的操作利用例如在本申请案全篇(包含在下文关于图26)所描述的电子硬件(例如,电路)来实施。

[0207] 在操作1405处,提取键的一部分。在实例中,所述键的所述部分为整个键。

[0208] 在操作1410处,从所述键的所述部分导出溢出值。在实例中,从所述键的所述部分导出所述溢出值包含执行所述键的所述部分的散列。

[0209] 在操作1415处,基于母节点的树层级而传回溢出值的一部分。在实例中,基于所述母节点的所述树层级而传回所述溢出值的所述部分包含将预设分摊应用于所述溢出值且传回与所述预设分摊及所述母节点的所述树层级对应的所述溢出值的所述部分。在此处,所述预设分摊定义应用于树的相应层级的溢出值的部分。

[0210] 在实例中,所述预设分摊定义所述树层级中的至少一些树层级的子节点的最大数目。在实例中,所述预设分摊定义树的最大深度。在实例中,所述预设分摊定义位计数序列,每一位计数规定位数,所述序列从低树层级到高树层级而定序,使得最低树层级的溢出值部分等于与在溢出值的起点开始的第一位计数相等的位数且第n个树层级的溢出值部分等于所述位计数序列中的第n个计数,其中以第一位计数开始且以n-1位计数结束的位计数的和的溢出值中具有偏移。

[0211] 图15是根据实施例的图解说明溢出压缩的框图。如上文所述,溢出压缩为键值压缩与树遍历(到子节点)的组合以得到合成kvset。因此,溢出压缩(或仅仅溢出)从合并集读取键值对及逻辑删除,移除所有过时键值对或逻辑删除(无用单元),将所得键值对及逻辑删除写入到含有合并集的节点的子节点中的一些或所有子节点中的新kvset,且删除包括所述合并集的所述kvset。这些新kvset原子地替换且逻辑上等效于合并集。

[0212] 溢出压缩使用用于将合并集中的键值对及逻辑删除分配到含有所述合并集的节点的子节点的确定性技术。具体来说,溢出压缩可使用任何此类键分配方法,使得对于给定节点及给定键K,溢出压缩始终将具有键K的任何(非过时)键值对或逻辑删除写入到所述节点的相同子节点。在优选实施例中,溢出压缩使用基于根数的键分配方法,例如下文详细呈现的实例中的键分配方法。

[0213] 为促进对溢出的理解,母节点包含包括合并集的两个kvset。所述两个kvset中的键值对1505、1510及1515分别具有00X、01X及11X的溢出值,其分别对应于母节点的四个子节点中的三个子节点。因此,将键值对1505放置到新kvset X中,将键值对1510放置到新kvset Y中,且将键值对1515放置到新kvset Z中,其中将每一新kvset写入到对应于溢出值的子节点。还注意,将新kvset写入到相应子节点中的最新(例如,最左)位置。

[0214] 在实例中,用于溢出压缩的合并集必须包含含有合并集的节点中的最旧kvset。在实例中,如果含有合并集的节点在溢出压缩的开始不具有子节点,那么创建经配置数目个子节点。

[0215] 正如上文所论述的其它压缩,可在正执行溢出压缩时将新kvset添加到含有用于溢出压缩的合并集的节点,因为按照定义,这些所添加kvset将不在用于溢出压缩的合并集中,且因为这些所添加kvset将逻辑上比由溢出压缩产生的kvset新。

[0216] 图16图解说明根据实施例的用于溢出压缩的方法1600的实例。方法1600的操作利用例如在本申请案全篇(包含在下文关于图26)所描述的电子硬件(例如,电路)来实施。

[0217] 在操作1605处,选择kvset序列的子集。在实例中,所述子集包含连续kvset,所述连续kvset还包含最旧kvset。

[0218] 在操作1610处,计算所述kvset子集的每一kvset中的每一键的子节点映射。在此处,所述子节点映射为基于特定键及母节点的树层级而从所述母节点到子节点的确定性映射。

[0219] 在操作1615处,基于其中每一kvset集恰好映射到一个子节点的子节点映射而将键及对应值收集到kvset中。在此收集期间可发生键冲突。如上文关于图10及12所论述,解决此冲突以支持较新键条目。

[0220] 在操作1620处,将所述kvset写入到相应子节点中的相应kvset序列中的最新位置。

[0221] 在操作1625处,从根节点移除所述kvset子集。

[0222] 方法1600可经扩展以包含在溢出操作的操作之后响应于子节点的指标超过阈值而对所述子节点执行第二溢出操作。

[0223] 图17是根据实施例的图解说明提升压缩的框图。提升压缩与溢出压缩的不同之处在于:将新kvset写入到母节点。因此,提升压缩或仅仅提升从合并集读取键值对及逻辑删除,移除所有过时键值对或逻辑删除,将所得键值对及逻辑删除写入到含有合并集的节点

的母节点中的新kvset,且删除包括合并集的kvset。这些新kvset原子地替换且逻辑上等效于合并集。

[0224] 由于KVS树中的kvset从树的根到叶经组织为从最新到最旧,因此提升压缩包含含有合并集的节点中的最新kvset且将由所述提升压缩产生的kvset放置在节点的母节点中的kvset序列中的最旧位置中。与上文所论述的其它压缩不同,为了确保来自经压缩的节点的最新kvset在合并集中,在正执行提升压缩时不可将新kvset添加到含有合并集的节点。因此,提升压缩为阻断压缩。

[0225] 如所图解说明,KVS 1705及1710的键值对经合并到新KVS M 1715中且存储于母节点的kvset序列中的最旧位置中。当(举例来说)目标为减少KVS树中的层级数目且因此增加在KVS树中对键进行搜索的效率时可将提升压缩应用于合并集。

[0226] 图18图解说明根据实施例的用于提升压缩的方法1800的实例。方法1800的操作利用例如在本申请案全篇(包含在下文关于图26)所描述的电子硬件(例如,电路)来实施。在实例中,

[0227] 在操作1805处,对子节点执行键与值压缩以产生新kvset而不将所述新kvset写入到所述子节点。

[0228] 在操作1810处,将所述新kvset在所述节点的kvset序列的最旧位置中写入到所述节点。

[0229] 键值压缩、溢出压缩及提升压缩操作可从合并集物理上移除过时键值对及逻辑删除且因此可减少存储于KVS树中的键值数据量(举例来说,以字节为单位)。在如此操作时,这些压缩操作从(举例来说)合并集中的值块读取非过时值,且将这些值写入到由压缩操作产生的kvset中的值块。

[0230] 相比之下,键压缩操作可从合并集物理上移除键(及逻辑删除)但仅逻辑上移除值。因此,所述值物理上保持在由键压缩产生的kvset中。键压缩可通过以下方式增加对含有合并集的节点中的键进行搜索的效率:减少所述节点中的kvset数目同时避免由(举例来说)键值压缩操作引发的值块的额外读取及写入。此外,键压缩提供对于未来维护操作有用的信息。KVS树由于如上文所描述的键块中的键与值块中的值的分隔而唯一地支持键压缩。

[0231] 在满足触发条件时操作上文所描述的KVS树维护技术(例如,压缩)。控制何时及在何处(例如,哪些节点)发生维护可提供对处理或者所花费时间的优化对比经增加空间或搜索效率。在维护期间或在引入期间搜集的一些指标可增强系统优化稍后维护操作的能力。在此处,这些指标称为无用单元指标或基于计算指标的方式而估计的无用单元指标。此类无用单元指标的实例包含节点中的过时键值对及逻辑删除的数目或其消耗的存储容量的量及由节点中的值块中的未经引用数据消耗的存储容量的量。此类无用单元指标指示可通过对节点的kvset执行(举例来说)键值压缩、溢出压缩或提升压缩而消除多少无用单元。

[0232] 再次,针对给定KVS树,计算或估计其节点的无用单元指标会提供数个优点,所述优点包含使如下各项为实际的:

[0233] A) 优先考虑将无用单元收集操作(特定来说物理上移除过时键值对及逻辑删除的无用单元收集操作,例如键值压缩、溢出压缩及提升压缩)应用于具有最多无用单元的所述节点。以此方式优先考虑无用单元收集操作会增加其效率且降低相关联写入放大率;或

[0234] B) 估计KVS树中的有效键值对数目及过时键值对数目以及由每一类别消耗的存储

容量的量。此类估计在报告KVS树的容量利用率方面为有用的。

[0235] 在一些情形中,直接计算KVS树中的给定节点的无用单元指标为有利的,然而在其它情形中估计所述无用单元指标为有利的。因此,在下文描述用于既计算无用单元指标又估计无用单元指标的技术。

[0236] 为促进无用单元指标的收集,可搜集或维持一些kvset统计数据。在实例中,这些统计数据维持在kvset集自身内,例如kvset的主要键块标头中。下文为可维持的kvset统计数据的非详尽列表:

[0237] A) 键值对数目

[0238] B) 键逻辑删除数目

[0239] C) 存储键值对的所有键及逻辑删除所需要的容量

[0240] D) 存储键值对的所有值所需要的容量

[0241] E) 包含最小值、最大值、中间值及平均值的键大小统计数据

[0242] F) 包含最小值、最大值、中间值及平均值的值大小统计数据

[0243] G) 在kvset为键压缩的结果的情况下未经引用值的计数及由未经引用值消耗的容量。

[0244] H) 任一键值对的最小及最大生存时间(TTL)值。KVS树可允许用户规定在存储键值对时的TTL值,且所述键值对在超过其寿命的情况下将在压缩操作期间被移除。

[0245] 经计算无用单元指标涉及用以产生已知结果的已知量的计算。举例来说,如果已知存在在kvset中过时的n个位,那么对kvset进行键值压缩将致使释放所述n个位。经计算无用单元指标的指标源为键压缩。键压缩从合并集逻辑上移除过时键值对及逻辑删除且物理上移除冗余键。然而,未经引用数据可保持在由键压缩产生的kvset的值块中。因此,键压缩致使知晓哪些值在新kvset中未经引用及其大小。知晓所述值的大小准许将在其它压缩下被释放的存储区的精确计数。因此,当对KVS树中的合并集执行键压缩时,可将所得kvset中的每一者的无用单元指标记录在相应kvset中。可从键压缩维持的实例性无用单元指标包含:

[0246] A) kvset中的未经引用值的计数

[0247] B) kvset中的未经引用值的字节

[0248] 在实例中,给定对合并集的第一键压缩,且给定在与第一键压缩相同的节点中的第二键压缩,其中用于第二键压缩的合并集包含由第一键压缩产生的kvset,那么可将第一键压缩记录的无用单元指标添加到从第二键压缩记录的相似无用单元指标。举例来说,如果第一键压缩操作产生具有规定未经引用值的Ucnt计数的相关联键压缩无用单元指标的单个kvset S,那么Ucnt可包含于由第二键压缩操作产生的键压缩无用单元指标中的未经引用值的计数中。

[0249] 在实例中,对于KVS树中的给定节点,如果用于键压缩操作的合并集包含节点中的所有kvset,那么所记录的键压缩无用单元指标可包含:

[0250] A) 节点中的未经引用值的计数

[0251] B) 节点中的未经引用值的字节

[0252] 显然,如果给定节点中的每个kvset为键压缩操作的结果,那么所述节点的键压缩无用单元指标为来自所述节点中的个别kvset中的每一者的相似键压缩无用单元指标的

和。

[0253] 经估计无用单元指标提供估计因对节点执行压缩而产生的增益的值。一般来说，在不执行键压缩的情况下搜集经估计无用单元指标。在下文论述中使用以下术语。使：

[0254] A) T = 给定节点中的kvset数目

[0255] B) $S(j)$ = 给定节点中的kvset, 其中 $S(1)$ 为最旧kvset且 $S(T)$ 为最新kvset

[0256] C) $KVcnt(S(j))$ = $S(j)$ 中的键值对数目

[0257] D) $NKVcnt = \sum(KVcnt(S(j)))$, 其中 j 在范围1到 T 中

[0258] E) $Kcap(S(j))$ = 以字节为单位的存储 $S(j)$ 的所有键所需要的容量

[0259] F) $NKcap = \sum(Kcap(S(j)))$, 其中 j 在范围1到 T 中

[0260] G) $Vcap(S(j))$ = 以字节为单位的存储 $S(j)$ 的所有值所需要的容量

[0261] H) $NVcap = \sum(Vcap(S(j)))$, 其中 j 在范围1到 T 中

[0262] I) $NKVcap = NKcap + NVcap$

[0263] 一种形式的经估计无用单元指标为历史无用单元指标。历史无用单元收集信息可用于估计KVS树中的给定节点的无用单元指标。此历史无用单元收集信息的实例包括但不限于：

[0264] A) 在给定节点中的无用单元收集操作的先前执行中过时键值对的分数的简单、积累或经加权移动平均数；或

[0265] B) 在与给定节点相同的KVS树的层级处的任一节点中的无用单元收集操作的先前执行中过时键值对的分数的简单、积累或经加权移动平均数。

[0266] 在以上实例中，无用单元收集操作包括但不限于键压缩、键值压缩、溢出压缩或提升压缩。

[0267] 给定KVS树中的节点，历史无用单元收集信息及kvset统计数据提供信息以产生节点的经估计无用单元指标。

[0268] 在实例中，可执行节点简单移动平均数(NodeSMA)以创建历史无用单元指标。在此处，使 $NSMA(E)$ = 在给定节点中的无用单元收集操作的最近 E 个执行中过时键值对的分数的平均值，其中 E 为可配置的。在此实例中，给定节点的NodeSMA经估计无用单元指标可包含以下各项：

[0269] A) 节点中的过时键值对的 $NKVcnt * NSMA(E)$ 计数；

[0270] B) 节点中的过时键值数据的 $NKVcap * NSMA(E)$ 个字节；

[0271] C) 节点中的有效键值对的 $NKVcnt - (NKVcnt * NSMA(E))$ 计数；或

[0272] D) 节点中的有效键值数据的 $NKVcap - (NKVcap * NSMA(E))$ 个字节。

[0273] 关于历史无用单元指标的另一变体包含层级简单移动平均数(LevelSMA)无用单元指标。在此实例中，使 $LSMA(E)$ = 在与给定节点相同的KVS树的层级处的任一节点中的无用单元收集操作的最近 E 个执行中过时键值对的分数的平均值，其中 E 为可配置的。在此实例中，给定节点的LevelSMA经估计无用单元指标可包含：

[0274] A) 节点中的过时键值对的 $NKVcnt * LSMA(E)$ 计数；

[0275] B) 节点中的过时键值数据的 $NKVcap * LSMA(E)$ 个字节；

[0276] C) 节点中的有效键值对的 $NKVcnt - (NKVcnt * LSMA(E))$ 计数；或

[0277] D) 节点中的有效键值数据的 $NKVcap - (NKVcap * LSMA(E))$ 个字节。

[0278] 历史无用单元指标的以上实例并非详尽的,而是图解说明经搜集的指标的类型。其它实例性历史无用单元指标可包含节点积累移动平均数(NodeCMA)无用单元指标、节点经加权移动平均数(NodeWMA)无用单元指标、层级积累移动平均数(LevelCMA)无用单元指标或层级经加权移动平均数(LevelWMA)无用单元指标。

[0279] 关于对于KVS树可用的维持键的kvset中的布隆过滤器的经估计无用单元指标的另一变体为布隆过滤器无用单元指标。如上所述,在KVS树的实例中,给定kvset包含布隆过滤器以高效地确定kvset是否可含有给定键,其中在kvset的布隆过滤器中针对kvset中的每一键存在一个条目。这些布隆过滤器可用于估计KVS树中的给定节点的无用单元指标。对于KVS树中的给定节点,技术(例如在帕帕皮特洛、奥德修斯等人的“布隆过滤器、分散式且并行数据库的基数估计及动态长度调适,201”中所论述)可用于约计由包括节点的kvset中的布隆过滤器表示的键集的交集的基数。此经约计值在此处称为节点的经布隆估计基数。

[0280] 给定KVS树中的节点,所述节点的经布隆估计基数及kvset统计数据准许以数种方式产生所述节点的经估计无用单元指标。实例性布隆过滤器无用单元指标包含布隆增量无用单元指标。使NBEC=给定节点中的T个kvset的经布隆估计基数,且Fobs=(NKVcnt-NBEC)/NKVcnt,其为对给定节点中的过时键值对的分数的估计。在此实例中,给定节点的布隆增量无用单元指标可包含:

[0281] A) 节点中的过时键值对的NKVcnt-NBEC计数;

[0282] B) 节点中的过时键值数据的NKVcap*Fobs个字节;

[0283] C) 节点中的有效键值对的NBEC计数;或

[0284] D) 节点中的有效键值数据的NKVcap-(NKVcap*Fobs)个字节。

[0285] 不同于布隆过滤器的概率滤波器(其中可能约计由两个或两个以上此类滤波器表示的键集的交集的基数)可在经估计无用单元指标中用作布隆过滤器的替代者。

[0286] 经计算且经估计无用单元指标可经组合以产生混合无用单元指标,即,由于包含另一形式的经估计无用单元指标而成为另一形式的经估计无用单元指标。举例来说,给定包括T个kvset的节点,如果键压缩无用单元指标可用于这些kvset中的W个kvset且W<T,那么可如下产生节点的混合无用单元指标。对于节点中的W个kvset(其中可获得键压缩无用单元指标),使:

[0287] A) KGM0cnt=对W个kvset中的过时键值对的计数的估计+来自W个kvset中的每一者的未经引用值的计数的和;

[0288] B) KGM0cap=对W个kvset中的过时键值数据的字节的估计+来自W个kvset中的每一者的未经引用值的字节的和;

[0289] C) KGMVcnt=对W个kvset中的有效键值对的计数的估计;及

[0290] D) KGMVcap=对W个kvset中的有效键值数据的字节的估计。

[0291] 其中可在W个kvset为节点中的仅有kvset的假定下使用上文所论述的技术中的一者产生经估计无用单元指标。

[0292] 对于节点中的(T-W)个kvset(其中不可获得键压缩无用单元指标),使:

[0293] A) EGM0cnt=对(T-W)个kvset中的过时(无用单元)键值对的计数的估计;

[0294] B) EGM0cap=对(T-W)个kvset中的过时(无用单元)键值数据的字节的估计;

[0295] C) EGMVcnt=对(T-W)个kvset中的有效键值对的计数的估计;及

[0296] D) $EGMVcap =$ 对 $(T-W)$ 个 $kvset$ 中的有效键值数据的字节的估计。

[0297] 其中可在 $(T-W)$ 个 $kvset$ 为节点中的仅有 $kvset$ 的假定下使用上文所论述的技术中的一者产生这些经估计无用单元指标。给定这些参数, 给定节点的混合无用单元指标可包含:

[0298] A) 节点中的过时键值对的 $KGM0cnt + EGMOcnt$ 计数;

[0299] B) 节点中的过时键值数据的 $KGM0cap + EGMOcap$ 个字节;

[0300] C) 节点中的有效键值对的 $KGMVcnt + EGMVcnt$ 计数; 或

[0301] D) 节点中的有效键值数据的 $KGMVcap + EGMVcap$ 个字节。

[0302] 无用单元指标允许针对具有足以有理由进行无用单元收集操作的开销的量的无用单元的树层级或节点优先考虑无用单元收集操作。以此方式优先考虑无用单元收集操作会增加其效率且降低相关联写入放大率。另外, 估计树中的有效键值对数目及过时键值对数目以及由每一类别消耗的存储容量的量在报告树的容量利用率方面为有用的。

[0303] 图19图解说明根据实施例的用于执行对KVS树的维护的方法1900的实例。方法1900的操作利用例如在本申请案全篇(包含在下文关于图26)所描述的电子硬件(例如, 电路)来实施。

[0304] 在操作1905处, 针对KVS树中的节点创建 $kvset$ 。作为所述 $kvset$ 创建的一部分, 针对所述 $kvset$ 计算 $kvset$ 指标集。在实例中, 所述 $kvset$ 指标集包含所述 $kvset$ 中的键值对数目。在实例中, 所述 $kvset$ 指标集包含所述 $kvset$ 中的逻辑删除数目。在实例中, 所述 $kvset$ 指标集包含用以存储针对所述 $kvset$ 中的键值对及逻辑删除的所有键条目的存储容量。在实例中, 所述 $kvset$ 指标集包含用于所述 $kvset$ 中的键值对的所有值的存储容量。

[0305] 在实例中, 所述 $kvset$ 指标集包含所述 $kvset$ 中的键的键大小统计数据。在实例中, 所述键大小统计数据包含最大值、最小值、中间值或平均值中的至少一者。在实例中, 所述 $kvset$ 指标集包含所述 $kvset$ 中的键的值大小统计数据。在实例中, 所述值大小统计数据包含最大值、最小值、中间值或平均值中的至少一者。

[0306] 在实例中, 所述 $kvset$ 指标集包含所述 $kvset$ 中的键值对的最小或最大生存时间(TTL)值。当引入操作规定键值对将为有效的周期时TTL可为有用的。因此, 在键值对到期之后, 主要目标是经由压缩操作进行回收。

[0307] 在实例中, 响应于压缩操作而创建 $kvset$ 。在此处, 所述压缩操作为键压缩、键值压缩、溢出压缩或提升压缩中的至少一者。在实例中, 所述压缩操作为键压缩。在此实例中, 所述 $kvset$ 指标集可包含所述 $kvset$ 中由于所述键压缩而产生的未经引用值的指标。在实例中, 所述未经引用值指标包含未经引用值的计数或由未经引用值消耗的存储容量中的至少一者。如本文中所使用, 视情况而定, 以由用以保存键条目或值的基础存储装置使用的位、字节、块等单位而测量所消耗的存储容量。

[0308] 在其中通过压缩操作创建 $kvset$ 的实例中, 所述 $kvset$ 指标集可包含对 $kvset$ 中的过时键值对的估计。如本文中所使用, 所述估计为如此是因为压缩仅深入了解经受压缩的合并集中的过时(例如, 作废)键值对, 且因此不知晓是否通过并非压缩的一部分的较新 $kvset$ 中的条目而使看似当前键值对为过时的。在实例中, 对过时键值对的所述估计可通过对来自预压缩 $kvset$ 的未包含于 $kvset$ 中的键条目的数目求和而计算得出。因此, 作为压缩的一部分, 关于合并集的过时对数目将为已知的且可用作对经创建 $kvset$ 中的过时数据的

估计。类似地,对kvset中的有效键值对的估计可通过对来自预压缩kvset的包含于kvset中的键条目的数目求和而计算得出且为所述kvset指标集的一部分。在实例中,所述kvset指标集包含kvset中的过时键值对的经估计存储大小。在实例中,包含kvset中的有效键值对的经估计存储大小,有效键值对的所述经估计存储大小是通过来自预压缩kvset的包含于kvset中的键条目及对应值的存储大小求和而计算得出。这些估计可用于历史指标,因为除非执行键压缩,否则将在压缩中移除经估计过时值。然而,如果节点在压缩中具有普通(例如,历史)性能,那么可假定此性能在未来持续下去。

[0309] 在实例中,所述kvset指标集存储于kvset中(例如,主要键块标头中)。在实例中,所述kvset指标集存储于节点中而非kvset中。在实例中,所述kvset指标的子集存储于kvset中且所述kvset指标的第二个子集存储于节点中。

[0310] 在操作1910处,将kvset添加到节点。一般来说,一旦添加到节点,便还写入kvset(例如,写入到磁盘上存储区)。

[0311] 在操作1915处,基于所述kvset指标集中的指标而针对压缩操作选择节点。因此,kvset指标或下文所论述的节点指标或两者可通过无用单元收集器或类似树维护过程而促成决定。在实例中,针对所述压缩操作选择所述节点包含:针对多个节点收集kvset指标集;基于所述kvset指标集而将所述多个节点排序;及基于来自所述排序的排序次序而选择所述多个节点的子集。在此实例中,可实施操作1920使得对所述节点执行所述压缩操作包含对所述多个节点(包含所述节点)的子集中的每一节点执行所述压缩操作。在实例中,所述多个节点的所述子集的基数由性能值设定。在实例中,所述性能值为如由所恢复空间测量的执行所述压缩的效率。此可通常实施为阈值。在实例中,可使用阈值函数,所述阈值函数接受若干个参数(例如留在基础存储装置上的未使用存储容量的量及对将在压缩操作中回收的容量的估计)以得出关于是否执行给定压缩操作的决定。

[0312] 在操作1920处,对节点执行压缩操作。在实例中,基于所述kvset指标集中的指标而选择压缩操作类型(例如,键压缩、键值压缩、溢出压缩或提升压缩)。

[0313] 方法1900的操作可经扩展以包含响应于将所述kvset添加到所述节点而修改节点指标。在实例中,所述节点指标包含经受对包含所述节点的节点群组执行的先前压缩的kvset中的经估计过时键值对的分数的值。在实例中,所述值为简单平均数。在实例中,所述值为移动平均数。在实例中,所述值为经加权平均数。在实例中,所述值为经受针对所述节点的设定数目次最近先前压缩的kvset中的经估计过时键值对的所述分数的平均值。在实例中,所述值为经受针对所述节点的树层级处的所有节点的设定数目次最近先前压缩的kvset中的经估计过时键值对的所述分数的平均值。

[0314] 在实例中,节点群组仅包含所述节点。在实例中,所述节点群组包含在所述节点的树层级上的所有节点。在实例中,所述节点指标包含由压缩操作产生的所述kvset指标集中的相似指标与因对所述节点执行的压缩操作产生的先前kvset指标的和。

[0315] 在实例中,所述节点指标包含在所述kvset及所述节点的不同kvset中为相同的键的经估计数目。在实例中,键的所述经估计数目是通过以下方式而计算得出:从所述kvset获得第一键布隆过滤器;从所述不同kvset获得第二键布隆过滤器;且将所述第一键布隆过滤器与所述第二键布隆过滤器进行交集运算以产生节点布隆过滤器估计的基数(NBEC)。尽管将此实例写为在两个kvset之间(例如,来自两个kvset的仅两个布隆过滤器的交集),但

可将任一数目个kvset布隆过滤器进行交集运算以得出NBEC,所述NBEC表示对所有kvset(其布隆过滤器为交集的一部分)所共有的键的数目的估计。

[0316] 在实例中,所述节点指标包含从NKVcnt值减去所述NBEC以估计节点中的过时键值的数目。在此处,所述NKVcnt值为其中为产生所述NBEC而将布隆过滤器进行交集运算的节点的每一kvset中的键值的总计数。在实例中,所述节点指标包含将NKVcap值乘以Fobs值。在此处,所述NKVcap值为由其中为产生所述NBEC而将布隆过滤器进行交集运算的所述节点中的每一kvset中的键及值使用的总存储容量,且所述Fobs值为从NKVcnt值减去所述NBEC且除以NKVcnt的结果,其中所述NKVcnt值为其中为产生所述NBEC而将布隆过滤器进行交集运算的所述节点的每一kvset中的键值的总计数。

[0317] 在实例中,所述节点指标存储于所述节点中。在此处,所述节点指标连同来自其它节点的节点指标一起经存储。在实例中,所述节点指标存储于树层级中,所述树层级对于所述KVS树的层级中的所有节点为共同的。

[0318] 可通过在特定情况下修改KVS树或其中的元素(例如,逻辑删除)的普通操作而以若干种方式辅助所述无用单元收集指标及上文所描述的用以改进KVS树性能的其使用。实例可包含逻辑删除加速度、更新逻辑删除、前缀逻辑删除或不可变数据KVS树。

[0319] 逻辑删除表示KVS树中的经删除键值。当将逻辑删除压缩在所述KVS树的叶中且所述压缩包含叶中的最旧kvset时,其实际上经移除,但以其它方式仍然阻止键的可能过时值在搜索中被传回。当键压缩或键值压缩在具有子节点的节点上的合并集中产生逻辑删除时,逻辑删除加速度包含遵循用于KVS树中的溢出压缩的键分配方法而将非过时逻辑删除写入到这些子节点中的一些或所有子节点中的一或多个新kvset。

[0320] 如果用于键压缩或键值压缩操作的合并集包含含有合并集的节点中的最旧kvset,那么经加速度逻辑删除(如果存在)不需要包含于通过压缩操作在所述节点中创建的新kvset中。以其它方式,如果用于键压缩或键值压缩操作的合并集不包含含有合并集的节点中的最旧kvset,那么经加速度逻辑删除(如果存在)还包含于通过压缩操作在所述节点中创建的新kvset中。将经加速度逻辑删除分配到KVS树的较旧区域中通过在不等待将原始逻辑删除推送到子节点的情况下允许移除子节点中的键值对而促进无用单元收集。

[0321] 键压缩或键值压缩操作可应用经规定或经计算准则以确定是否还执行逻辑删除加速度。此逻辑删除加速度准则的实例包含但不限于合并集中的非过时逻辑删除数目及可为已知或估计的通过合并集中的逻辑删除经逻辑上删除的键值数据量(举例来说,以字节为单位)。

[0322] 更新逻辑删除以类似于经加速度逻辑删除的方式来操作,尽管原始引入值并非逻辑删除。本质上,当将新值添加到KVS树时,可对所述键的所有较旧值进行无用单元收集。将与经加速度逻辑删除近似的逻辑删除沿着树向下推送将允许对这些子节点的压缩以移除过时值。

[0323] 在实例中,在KVS树中,引入操作将新kvset添加到根节点,且此新kvset中具有键K的键值对包含如下的标志或其它指示符:其为替换包含于较早引入操作中的具有键K的键值对的更新键值对。预期但不要求此指示符为准确的。如果具有键K的更新键值对与引入操作一起经包含,且如果根节点具有子节点,那么所述引入操作还可遵循用于KVS树中的溢出压缩的键分配方法而将键K的键逻辑删除(更新逻辑删除)写入到根节点的子节点中的新

kvset。

[0324] 在实例中,或者,响应于处理具有键K的更新键值对,对根节点中的合并集的键压缩或键值压缩操作还可遵循用于KVS树中的溢出压缩的键分配方法而将键K的键逻辑删除(再次称为更新逻辑删除)写入到根节点的子节点中的新kvset。在实例中,对于具有键K的给定更新键值对,针对键K写入至少一个对应更新逻辑删除。

[0325] 虽然下文关于图25论述KVS树前缀操作,但所述概念还可用于逻辑删除中。在前缀操作中,键的一部分(前缀)用于匹配。一般来说,键的前缀部分全部用于创建溢出值,尽管较小部分可与较深树确定一起使用,从而在消耗前缀路径之后扇出到所有子节点。前缀逻辑删除使用匹配多个值的前缀的幂以使单个条目表示许多键值对的删除。

[0326] 在实例中,溢出压缩基于键的第一子键的溢出值而使用键分配方法,所述第一子键为键前缀。前缀逻辑删除为包括键前缀的逻辑记录且指示以前缀及其相关联值(如果存在)开始的所有键已在特定时间点从KVS树逻辑上删除。前缀逻辑删除在KVS树中用于与键逻辑删除相同的目的,惟前缀逻辑删除可逻辑上删除一个以上有效键值对然而键逻辑删除可逻辑上删除恰好一个有效键值对除外。在此实例中,由于溢出压缩使用由前缀规定的第一子键值产生前缀逻辑删除的溢出值,因此具有等效第一子键值的每个键值对、键逻辑删除或前缀逻辑删除将采取穿过KVS树的层级的相同路径,因为其将具有等效溢出值。

[0327] 在实例中,逻辑删除加速度可应用于前缀逻辑删除以及键逻辑删除。在应用逻辑删除加速度准则时前缀逻辑删除可以不同于键逻辑删除的方式来处理,因为前缀逻辑删除可在后续无用单元收集操作中引起大量过时键值对或逻辑删除的物理移除。

[0328] 上文所论述的逻辑删除加速度技术致使创建更大数目个kvset且因此可为低效的。由于写入数据的应用程序可知晓先前写入的数据的大小,因此逻辑删除可包含其依据应用程序而替换的数据的大小。此信息可由系统使用以确定是否执行上文所论述的逻辑删除加速度(或产生更新逻辑删除)。

[0329] 一些数据可为不可变的。不可变键值数据的一些实例包含时序数据、日志数据、传感器数据、机器产生的数据及数据库提取、变换与加载(ETL)过程的输出以及其它。在实例中,KVS树可经配置以存储不可变键值数据。在此配置中,预期但不要求通过引入操作添加到KVS树的kvset不含有逻辑删除。

[0330] 在实例中,KVS树可经配置以存储仅受含有KVS树的存储媒体的容量限定的一定量的不可变数据。在KVS树的此配置中,所执行的仅有无用单元收集操作为键压缩。在此处,执行键压缩以通过减少根节点中的kvset数目而增加对KVS树中的键进行搜索的效率。注意,在不具有溢出压缩的情况下,根节点将为KVS树中的仅有节点。在实例中,压缩准则可包含根节点中的kvset数目或键搜索时间统计数据,例如最小、最大、平均数及平均值搜索时间。这些统计数据可在特定事件时经复位,例如在键压缩之后、在引入操作之后、在经配置时间间隔到期时或在执行经配置数目次键搜索之后。在实例中,用于键压缩的合并集可包含根节点中的kvset中的一些或所有kvset。

[0331] 在实例中,KVS树可经配置以存储受保留准则限定的一定量的不可变数据,所述保留准则可通过以先进先出(FIFO)方式从KVS树移除键值对来实行。此保留准则的实例包含:KVS树中的键值对的最大计数;KVS树中的键值数据的最大字节;或KVS树中的键值对的最大年龄。

[0332] 在KVS树的此配置中,所执行的仅有无用单元收集操作为键压缩。在此处,执行键压缩以既增加对KVS树中的键进行搜索的效率(通过减少根节点中的kvset数目)又促进以FIFO方式从KVS树移除键值对从而实行保留准则。在实例中,压缩准则可规定每当根节点中的两个或两个以上连续kvset(包括用于键压缩的合并集)满足称为保留增量的保留准则的经配置分数时执行键压缩。下文为保留要求的一些实例:

[0333] A) 如果保留准则为KVS树中的W个键值对,且保留增量为 $0.10*W$ 个键值对,那么在两个或两个以上连续kvset(合并集)具有经组合 $0/10*W$ 计数的键值对的情况下执行键压缩;

[0334] B) 如果保留准则为KVS树中的键值数据的X个字节,且保留增量为键值数据的 $0.20*X$ 个字节,那么在两个或两个以上连续kvset(合并集)具有经组合 $0.20*X$ 个字节的键值数据的情况下执行键压缩;或

[0335] C) 如果保留准则为KVS树中的键值数据的Y天,且保留增量为键值数据的 $0.15*Y$ 天,那么在两个或两个以上连续kvset(合并集)具有经组合 $0.15*Y$ 天的键值数据的情况下执行键压缩。

[0336] 可存在其中要求用于键压缩的合并集精确地满足经配置保留增量为不实际的情形。因此,在实例中,可使用保留增量的约计。

[0337] 给定KVS树及各自低于经配置保留增量的kvset的引入操作序列,执行如上文所描述的键压缩操作在根节点中产生各自满足或约计保留增量的kvset。此结果的例外可为最新kvset,其经组合可低于保留增量。不管此可能结果,每当KVS树超过保留准则达至少保留增量时,便可删除KVS树中的最旧kvset。举例来说,如果保留准则为KVS树中的W个键值对,且经配置保留增量为 $0.10*W$ 个键值对,那么KVS树的根节点中的kvset将各自具有大致 $0.10*W$ 个键值对,其中经组合的最新kvset的可能例外可具有少于 $0.10*W$ 个键值对。作为结果,每当KVS树超过W个键值对达至少 $0.10*W$ 个键值对时,便可删除KVS树中的最旧kvset。

[0338] 逻辑删除加速度、更新加速度或前缀逻辑删除的无用单元收集促进器可应用于除KVS树以外的其它键值存储区。举例来说,逻辑删除加速度或更新逻辑删除可借助一或多个无用单元收集操作应用于LSM树变体中,所述一或多个无用单元收集操作将键值数据写入到同一树层级(从其读取所述键值数据)且以类似于KVS树中的键压缩或键值压缩的方式操作。更新逻辑删除还可应用于LSM树变体,其中准许将逻辑删除引入从根节点的子节点中。在另一实例中,前缀逻辑删除可用于LSM树变体中,所述LSM树变体每层级具有仅一个节点(此为常见的)或实施键分配方法以用于基于键的一部分(例如子键)而选择子节点。在另一实例中,逻辑删除大小可使用逻辑删除加速度应用于LSM树变体中。此外,用于优化对不可变键值数据的无用单元收集的技术可借助不读取或写入键值数据中的值的无用单元收集操作(类似于KVS树中的键压缩)应用于LSM树变体。

[0339] 实施这些无用单元收集促进器会改进KVS树或若干数据结构中的无用单元收集的效率。举例来说,逻辑删除加速度致使将逻辑删除写入到树的较低层级早于在应用键压缩、键值压缩或类似操作时将以其它方式所发生的,借此使得可能在树的所有层级处更迅速地消除无用单元。连同键压缩或类似操作使用的逻辑删除加速度在写入放大率远小于将由溢出压缩产生的写入放大率的情况下实现这些结果。在其它实例中,前缀逻辑删除允许单个逻辑删除记录逻辑上删除大量相关键值对,更新逻辑删除将逻辑删除加速度的益处带到更

新键值对,逻辑删除大小在评估逻辑删除加速度准则时改进准确度,且用于优化对不可变键值数据的无用单元收集的技术产生键值数据中的值的为一(1)的写入放大率。

[0340] 图20图解说明根据实施例的用于修改KVS树操作的方法2000的实例。方法2000的操作利用例如在本申请案全篇(包含在下文关于图26)所描述的电子硬件(例如,电路)来实施。方法2000涵盖实施上文关于KVS树中的逻辑删除加速度、更新加速度(例如,更新逻辑删除)、前缀逻辑删除及不可变键值数据所论述的若干个特征的操作。

[0341] 在操作2005处,接收对KVS树的请求。在实例中,所述请求包含键前缀及逻辑删除,所述参数集具有在所述请求中将所述逻辑删除定义为前缀逻辑删除的成员,且执行对所述KVS树的所述请求包含将所述前缀逻辑删除写入到所述KVS树的kvset。在实例中,在将键进行比较的KVS树操作时,前缀逻辑删除匹配具有与所述前缀逻辑删除的所述键前缀相同的前缀的任何键。

[0342] 在实例中,所述请求包含键,所述参数集包含规定逻辑删除加速度的成员;且执行对所述KVS树的所述请求包含将逻辑删除写入于通过对所述键执行溢出函数而规定的至少一个子节点中。所述溢出函数为将键(或键的一部分)视为输入且产生溢出值的函数,如上文关于图13所提及。在实例中,将所述逻辑删除写入到通过对所述键执行所述溢出函数而规定的所有现存子节点。在实例中,所述请求包含逻辑删除。在实例中,所述请求包含值。

[0343] 在操作2010处,接收所述KVS树的参数集。

[0344] 在操作2015处,通过根据参数修改所述KVS树的操作而执行对所述KVS树的所述请求。

[0345] 在实例中,所述请求包含键、逻辑删除及在所述KVS树中与所述键对应的值的存储大小。在此处,所述参数集具有规定无用单元收集统计数据存储区的成员,且执行对所述KVS树的所述请求包含将所述键及所述存储大小存储于所述KVS树的数据结构中。在实例中,所述逻辑删除为前缀逻辑删除。

[0346] 在实例中,所述参数集包含规定所述KVS树为不可变的成员,且执行对所述KVS树的所述请求包含将所述请求写入到所述KVS树的根节点。在此处,在所述KVS树为不可变的时所述根节点为所述KVS树中的仅有节点。

[0347] 在实例中,当所述KVS树为不可变的时所述KVS树排他地使用键压缩。在实例中,方法2000可经扩展以响应于所述KVS树为不可变的而存储键搜索统计数据。在实例中,所述键搜索统计数据为最小、最大、平均数或平均值搜索时间中的至少一者。在实例中,所述键搜索统计数据为所述根节点中的kvset数目。

[0348] 在实例中,当所述KVS树为不可变的时,方法2000可经扩展以响应于所述键搜索统计数据满足阈值而执行键压缩。在实例中,所述键压缩可包含响应于以下情形中的至少一者而将所述键搜索统计数据复位:压缩、引入、规定数目次搜索之后或规定时间间隔之后。

[0349] 在其中所述参数集的第二成员规定所述KVS树在先进先出基础上移除元素的实例中,所述参数集第三成员规定所述KVS树的保留约束,所述KVS树基于所述保留约束而对kvset执行键压缩,且所述KVS树在违反所述保留约束时移除最旧kvset。在实例中,所述保留约束为最大键值对数目。在实例中,所述保留约束为键值对的最大年龄。在实例中,所述保留约束为由键值对消耗的最大存储值。

[0350] 在实例中,基于所述保留约束而对kvset执行键压缩包含:将连续kvset分组以产

生群组集,来自所述群组集中的每一成员的经求和指标约计所述保留约束的分数;且对所述群组集的每一成员执行键压缩。

[0351] 图21是根据实施例的图解说明键搜索的框图。所述搜索通过以下方式而进展:在根节点中的最新kvset处开始且逐渐移动到较旧kvset直到找到键或者叶节点中的最旧kvset不具有所述键为止。由于母节点到子节点键映射的确定性本质,因此将存在经搜索的仅一个叶,且所述叶中的最旧kvset将具有最旧键条目。因此,如果遵循经图解说明搜索路径且未找到键,那么所述键不在KVS树中。

[0352] 键的最新键条目一经找到就停止搜索。因此,搜索路径从最新移动到最旧且键的键条目一经定位就停止。此行为通过不要求立即从KVS树移除过时键值对而允许保持kvset的不可变性。替代地,较新值或用以指示删除的逻辑删除经放置在较新kvset中且将首先被找到,从而不考虑仍驻存于KVS树中的较旧键对版本而产生对查询的准确响应。

[0353] 在实例中,可通过将当前节点设定到根节点而执行对键K的搜索。如果在当前节点中找到具有键K的键值对或逻辑删除,那么完成搜索且作为结果而分别传回相关联值或“未找到键”的指示。如果未找到键K,那么将当前节点设定到如由键K及用于溢出压缩的键分配方法确定的节点的子节点。

[0354] 如果不存在此类子节点,那么完成搜索且“未找到键”的指示为结果。否则,执行对当前节点的kvset中的键K的搜索且重复所述过程。在概念上讲,对KVS树中的键K的搜索遵循具有键K的每个键值对或逻辑删除由于溢出压缩而采取的穿过KVS树的相同路径。

[0355] 由于基于键的母节点与子节点之间的确定性映射,因此搜索KVS树中的每层级仅一个节点直到找到具有键K的键值对或逻辑删除或搜索到KVS树中的最后(例如,最大编号的)层级中的节点为止。因此,搜索为高度高效的。

[0356] 图22图解说明根据实施例的用于执行键搜索的方法2200的实例。方法2200的操作利用例如在本申请案全篇(包含在下文关于图26)所描述的电子硬件(例如,电路)来实施。

[0357] 在操作2205处,接收包含键的搜索请求。

[0358] 在操作2210处,选择根节点作为当前节点。

[0359] 在操作2215处,检验当前节点。

[0360] 在操作2220处,检验以对当前节点的最新kvset的查询开始。

[0361] 在决定2225处,如果未找到键,那么方法2200继续进行到决定2240,且否则,如果找到键,那么方法2200继续进行到决定2230。

[0362] 在决定2230处,如果对应于键的键条目包含或引用逻辑删除,那么方法2200继续进行到结果2260且否则继续进行到结果2235。

[0363] 在结果2235处,回答搜索请求而传回与键的最新键条目对应的值。

[0364] 在决定2240处,如果当前节点中存在更多kvset,那么方法2200继续进行到操作2245且否则继续进行到决定2250。

[0365] 在操作2245处,方法2200选择当前节点中的下一最新kvset以查询键且继续进行到决定2225。

[0366] 在决定2250处,如果当前节点不具有与键的溢出函数匹配的任何子节点,那么方法2200继续进行到结果2260且否则以其它方式继续进行到操作2255。

[0367] 在操作2255处,将与键的溢出函数匹配的子节点设定为当前节点且方法2200继续

进行到操作2215。

[0368] 在结果2260处,回答搜索请求而传回搜索的否定指示,例如“未找到键”。

[0369] 扫描操作不同于正寻求的多个键中的搜索。典型扫描操作可包含对键范围的搜索,其中所述搜索规定多个键来限定所述范围。一般来说,扫描规定准则且预期kvs树中满足所述准则的所有键的结果。

[0370] 图23是根据实施例的图解说明键扫描的框图。键扫描或纯粹扫描识别KVS树的每个节点中含有满足扫描准则(例如,属于规定范围内)的键条目的每个kvset。虽然kvset的键存储区准许对特定键的高效搜索,但为确保找到满足扫描准则的每个键,致使搜索每个kvset。然而,由于kvset中的键值存储区的经键排序本质,因此扫描可在不查看每个键的情况下迅速地确定。此仍比由WB树提供的能力好,举例来说,因为键值对未存储于经键排序结构中,而是保持键以解决键散列冲突。因此,必须读取WB树中的每个键以满足扫描。

[0371] 在KVS树中,为促进扫描,以经键排序次序将键存储于kvset中。因此,给定键可位于日志时间中且还可迅速地确定范围内的键(例如,范围中的最高及最低键)。此外,上文关于图1到5所论述的实例性kvset元数据可用于更进一步地加速扫描。举例来说,如果kvset维持含纳于kvset内的最小及最大键值,那么扫描可迅速地确定kvset中的键不满足规定范围。类似地,维持kvset键的布隆过滤器可用于迅速地确定特定键不在给定kvset的键存储区中。

[0372] 在实例(未图解说明)中,除上文以外,扫描可与搜索很像地继续进行,惟访问每个节点除外。因此,扫描从kvset读取满足准则的每个键的最新记录,其中给定键K的最新记录可为键值对或键逻辑删除。如上文所述,在KVS树中的给定节点内,kvset从最新到最旧而定序,且在层级(L+1)处的节点中的kvset比在层级L处的节点中的kvset旧。在找到满足准则的键之后,在结果集中将所述键传回到请求者。

[0373] 当认识到在扫描中发生每个节点中的每个kvset的访问时可改进上文直接描述的类似搜索的扫描。因此,在实例中,可同时读取所述kvset。所有kvset的同时读取可产生非常大的缓冲区(例如,用于经传回结果的存储位置)。然而,此可通过迅速地确定给定kvset是否具有满足扫描准则(例如,在范围内)的键的能力来缓解。因此,可访问每个kvset,但仅读取具有满足所述准则的键的所述kvset。在图23中图解说明此实例。具体来说,读取器同时访问所述kvset中的所有kvset(例如,虚线及虚线kvset)且然而仅读取所述kvset的子集(虚线kvset)。此技术支持迭代器风格语义,其中程序可询问下一或先前键。所述kvset中的键的经排序本质准许迅速识别出下一键以及键上是否存在冲突(例如,同一键的多个条目)、哪一值为将传回到程序的最新者—除非最新值为逻辑删除,在所述情形中迭代器应跳过所述键且为下一键提供最新值。

[0374] 在实例中,扫描可包含接收包含键范围(或其它准则)的扫描请求。

[0375] 扫描通过将由范围规定的键来自树的节点集的每一kvset收集到经找到集中而继续进行。在实例中,所述节点集包含树中的每个节点。

[0376] 扫描通过以下方式而继续进行:通过保持与并非逻辑删除的键的最近条目对应的键值对而将所述经找到集缩减到结果集。

[0377] 扫描通过传回所述结果集而完成。

[0378] 图24是根据实施例的图解说明键扫描的框图。图24提供不同于图23的视角。扫描

的准则为在A与K(包含A及K)之间的键。扫描以根节点的最新kvset(其为KVS树中的最新kvset, kvset 12)来开始。在实例中, kvset 12的键指标允许至少一些键满足所述准则的迅速确定。具体来说, 在此实例中, 其为键A及B。扫描从KVS树的顶部(根)到底部(叶)而从每一节点中的最新kvset到最旧kvset继续进行。注意, 键A、B、C、E及K跨越节点出现在多个kvset中。扫描将仅保留每一kvset的最新者(例如, 选定键)。因此, 结果集将包含在针对键A及B的kvset 12、针对键C的kvset 11、针对键E的kvset 10及针对键K的kvset 6中找到的这些键的值。然而, 如果针对这些键中的任何者的这些kvset中的键条目包含或引用逻辑删除, 那么将从结果集省略所述键。Kvset 5中的键D的唯一性使得其值包含在结果集中(假定键D并不是指逻辑删除)。

[0379] 图25是根据实施例的图解说明前缀扫描的框图。前缀扫描定位KVS树中的所有键值对(如果存在), 其中键全部以规定前缀开始。尽管前缀小于整个键, 且因此可匹配多个键, 但键的前缀部分至少与由溢出函数使用以创建溢出值的键的部分一样大。因此, 如果溢出函数使用键的第一子键, 那么前缀包含第一子键(且可包含额外子键)。此要求允许确定性映射将前缀扫描性能改进为优于纯粹扫描性能, 因为仅访问前缀的路径中的所述节点。

[0380] 在实例中, 溢出值基于键的第一子键。在此实例中, 规定前缀包含键的第一子键的值。在此实例中, 前缀扫描可通过以下方式继续进行: 识别在KVS树的每个节点中含有具有以规定前缀开始的键的键值对或逻辑删除的每个kvset。与纯粹扫描相比较, 前缀扫描不访问KVS树的每个节点。更确切来说, 经检验节点可经拘限于沿着由定义前缀的第一子键值的溢出值确定的路径的所述节点。在实例中, 替代使用第一子键, 可针对溢出值使用最后子键以实现前缀扫描。在此实例中, 规定前缀包含键的最后子键的值。可基于在溢出值计算中使用的特定子键而实施额外各种扫描。

[0381] 再次, 类似于纯粹扫描, 存在检索键或键值对以实施扫描的多种方式。在实例中, 如所图解说明, 同时访问(虚线)沿着由前缀给定的溢出值路径的节点(具有虚线边缘的节点), 针对满足扫描准则的键而测试所述节点内的kvset, 且读取通过测试的kvset(具有虚线边缘的kvset)。

[0382] 前缀扫描为极其高效的, 此既因经检查的节点数目限于KVS树的每层级有一个, 又因kvset键存储区中的键一般存储于允许匹配前缀的键的就绪识别的结构中。另外, 上文关于键扫描所论述的kvset指标还可有助于加速搜索。

[0383] 所述前缀扫描可包含接收具有键前缀的扫描请求。在此处, 将搜索的节点集包含与所述键前缀对应的每一节点。在实例中, 与所述键前缀的节点对应性由从所述键前缀导出的溢出值的一部分确定, 所述溢出值的所述部分由给定节点的树层级确定。

[0384] 所述前缀扫描通过以下方式继续进行: 将由所述前缀规定的键从来自树的所述节点集的每一kvset收集到经找到集中。

[0385] 所述前缀扫描通过以下方式继续进行: 通过保持与并非逻辑删除且未由更近逻辑删除来删除的键的最近条目对应的键值对而将所述经找到集缩减到结果集。

[0386] 所述前缀扫描通过传回所述结果集而完成。

[0387] 如上文所描述, KVS树提供用以将键值数据存储于磁盘上的强大结构。KVS树包含LSM树及WB树的优点中的许多优点而不具有这些结构的缺点。举例来说, 关于存储空间或归因于压缩的写入放大率, 在KVS树中, 可容易地控制节点的大小以限制用于压缩的临时存储

容量的最大量。此外,键压缩可用于在不读取及写入值块的情况下增加节点中的搜索效率,借此减小归因于压缩的读取放大率及写入放大率。在传统LSM树中,压缩所需要的临时存储容量的量以及读取放大率及写入放大率的量可与经压缩的树层级处的键值容量的量成比例—此因如下事实而加剧:LSM树中的树层级的键值容量通常经配置以在树中更深的每一树层级处指数增长。

[0388] 关于键搜索效率,在KVS树中,对键K进行搜索涉及每树层级搜索仅一个节点(其表示KVS树中的总键的仅小分数)。在传统LSM树中,对键K进行搜索需要搜索每一层级中的所有键。

[0389] 关于如上文所述的前缀扫描效率,KVS树的实例准许通过每树层级搜索仅一个节点(其表示KVS树中的总键的仅小分数)而找到以规定前缀开始的所有键。在传统LSM树中,找到以规定前缀开始的所有键需要搜索每一层级中的所有键。

[0390] 关于扫描效率,上文所描述的KVS树的实例准许通过利用kvset中的数据而找到在给定范围中或以规定前缀开始的所有键。在WB树中,所述键为无序的,从而不产生用以实施这些操作中的任一者的高效方式。因此,在WB树中,必须检索且检验树的每个条目以执行这些扫描。

[0391] 关于压缩性能,在KVS树中,键、键值及溢出压缩维护技术(惟提升压缩除外)由于节点中的kvset的时间上经排序本质而非阻断的。因此,可将新kvset添加到节点,通过仅仅将新kvset放置在最新位置中而对所述节点执行键、键值或溢出压缩。在WB树中,压缩为阻断操作。

[0392] 图26图解说明可在其上执行本文中所论述的技术(例如,方法)中的任何一或多者的实例性机器2600的框图。在替代实施例中,机器2600可操作为独立装置或可连接(例如,网络连接)到其它机器。在网络化部署中,机器2600可在服务器-客户端网络环境中作为服务器机器、客户端机器或两者来操作。在实例中,机器2600可在对等(P2P)(或其它分散式)网络环境中用作对等机器。机器2600可为个人计算机(PC)、平板PC、机顶盒(STB)、个人数字助理(PDA)、移动电话、web器具、网络路由器、交换机或桥接器或能够执行规定将由所述机器采取的动作的指令(顺序的或其它)的任何机器。此外,虽然图解说明仅单个机器,但还应将术语“机器”视为包含个别地或联合地执行指令集(或多个指令集)以执行本文中所论述的方法中的任何一或多者的任何机器集合,例如云计算、软件即服务(SaaS)、其它计算机群集配置。

[0393] 如本文中所描述的实例可包含逻辑或若干个组件或机构或者可由所述逻辑或若干个组件或机构操作。电路为在包含硬件的有形实体(例如,简单电路、门、逻辑等)中实施的电路集合。电路成员可为随时间而变通的。电路包含可在操作时单独或以组合形式执行规定操作的成员。在实例中,电路的硬件可以不可变方式经设计以实施特定操作(例如,硬接线)。在实例中,电路的硬件可包含以可变方式连接的物理组件(例如,执行单元、晶体管、简单电路等),包含经物理上修改(例如,以磁性方式、以电方式、质量不变的粒子的可移动放置等)以编码特定操作的指令的计算机可读媒体。在连接物理组件时,硬件组成的基本电性质(举例来说)从绝缘体改变到导体或反之亦然。指令使得嵌入式硬件(例如,执行单元或加载机构)能够经由可变连接形成硬件中的电路的成员以在操作中时实施特定操作的若干部分。因此,计算机可读媒体在装置操作时以通信方式耦合到电路的其它组件。在实例中,

可在一个以上电路的一个以上成员中使用物理组件中的任一者。举例来说,在操作下,执行单元可在一个时间点在第一电路的第一子电路中使用且在不同时间由所述第一电路中的第二子电路重新使用,或由第二电路中的第三子电路使用。

[0394] 机器(例如,计算机系统)2600可包含硬件处理器2602(例如,中央处理单元(CPU)、图形处理单元(GPU)、硬件处理器核心或其任何组合)、主要存储器2604及静态存储器2606,其中的一些或所有可经由互连链路(例如,总线)2608彼此通信。机器2600可进一步包含显示单元2610、字母数字输入装置2612(例如,键盘)及用户接口(UI)导航装置2614(例如,鼠标)。在实例中,显示单元2610、输入装置2612及UI导航装置2614可为触摸屏显示器。机器2600可另外包含存储装置(例如,驱动器单元)2616、信号产生装置2618(例如,扬声器)、网络接口装置2620及一或多个传感器2621,例如全球定位系统(GPS)传感器、罗盘、加速度计或其它传感器。机器2600可包含输出控制器2628,例如串联(例如,通用串行总线(USB)、并行或其它有线或无线(例如,红外(IR)、近红外通信(NFC)等)连接以与一或多个外围装置(例如,打印机、读卡器等)通信或控制所述一或多个外围装置。

[0395] 存储装置2616可包含其上存储有体现本文中所描述的技术或功能中的任何一或多者或由本文中所描述的技术或功能中的任何一或多者利用的一或多个数据结构或指令2624集(例如,软件)的机器可读媒体2622。指令2624还可在其由机器2600执行期间完全地或至少部分地驻存于主要存储器2604内、静态存储器2606内或硬件处理器2602内。在实例中,硬件处理器2602、主要存储器2604、静态存储器2606或存储装置2616中的一者或其任一组合可构成机器可读媒体。

[0396] 虽然机器可读媒体2622经图解说明为单个媒体,但术语“机器可读媒体”可包含经配置以存储一或多个指令2624的单个媒体或多个媒体(例如,集中式或分散式数据库,及/或相关联高速缓冲存储器及服务器)。

[0397] 术语“机器可读媒体”可包含能够存储、编码或载运用于由机器2600执行且致使机器2600执行本发明的技术中的任何一或多者的指令或者能够存储、编码或载运由此类指令使用或与此类指令相关联的数据结构的任何媒体。非限制性机器可读媒体实例可包含固态存储器以及光学及磁性媒体。在实例中,大规模机器可读媒体包括包含具有不变(例如,静止)质量的多个粒子的机器可读媒体。因此,大规模机器可读媒体并非暂时传播信号。大规模机器可读媒体的特定实例可包含:非易失性存储器,例如半导体存储器装置(例如,电可编程只读存储器(EPROM)、电可擦除可编程只读存储器(EEPROM)及快闪存储器装置;磁盘,例如内部硬盘及可抽换磁盘;磁光盘;以及CD-ROM及DVD-ROM磁盘。

[0398] 可经由通信网络2626使用传输媒体经由网络接口装置2620进一步传输或接收指令2624,网络接口装置2620利用若干个传送协议(例如,帧中继、因特网协议(IP)、传输控制协议(TCP)、用户数据报协议(UDP)、超文本传送协议(HTTP)等)中的任一者。实例性通信网络可包含局域网(LAN)、广域网(WAN)、数据包数据网络(例如,因特网)、移动电话网络(例如,蜂窝网络)、普通旧式电话(POTS)网络及无线数据网络(例如,称为Wi-Fi®的美国电气与电子工程师协会(IEEE)802.11系列标准、称为WiMax®的IEEE 802.16系列标准)、IEEE 802.15.4系列标准、对等(P2P)网络以及其它网络。在实例中,网络接口装置2620可包含一或多个物理插座(例如,以太网、同轴或耳机插座)或者一或多个天线以连接到通信网络2626。在实例中,网络接口装置2620可包含多个天线以使用单输入多输出(SIMO)、多输入多

输出 (MIMO) 或多输入单输出 (MISO) 技术中的至少一者无线地通信。术语“传输媒体”应被视为包含能够存储、编码或载运用于由机器2600执行的指令的任何无形媒体,且包含数字或模拟通信信号或其它无形媒体以促进此软件的通信。

[0399] 额外说明及实例

[0400] 实例1为一种系统,其包括经配置以进行以下操作的处理电路:接收对KVS树的请求,所述KVS树为包含节点的数据结构,所述节点包含时间上经定序的kvset序列,所述kvset以经排序次序存储键;接收所述KVS树的参数集;且通过根据参数修改所述KVS树的操作而执行对所述KVS树的所述请求。

[0401] 在实例2中,根据实例1所述的标的物,其中所述请求包含键前缀及逻辑删除,其中所述参数集具有在所述请求中将所述逻辑删除定义为前缀逻辑删除的成员,且其中为执行对所述KVS树的所述请求,所述处理电路经配置以将所述前缀逻辑删除写入到所述KVS树的kvset。

[0402] 在实例3中,根据实例2所述的标的物,其中在将键进行比较的KVS树操作时,前缀逻辑删除匹配具有与所述前缀逻辑删除的所述键前缀相同的前缀的任何键。

[0403] 在实例4中,根据实例1到3中任一或多个实例所述的标的物,其中所述请求包含键,其中所述参数集包含规定逻辑删除加速度的成员;且其中为执行对所述KVS树的所述请求,所述处理电路经配置以将逻辑删除写入于通过对所述键执行溢出函数而规定的至少一个子节点中。

[0404] 在实例5中,根据实例4所述的标的物,其中将所述逻辑删除写入到通过对所述键执行所述溢出函数而规定的所有现存子节点。

[0405] 在实例6中,根据实例4到5中任一或多个实例所述的标的物,其中所述请求包含逻辑删除。

[0406] 在实例7中,根据实例4到6中任一或多个实例所述的标的物,其中所述请求包含值。

[0407] 在实例8中,根据实例1到7中任一或多个实例所述的标的物,其中所述请求包含键、逻辑删除及在所述KVS树中与所述键对应的值的存储大小,其中所述参数集具有规定无用单元收集统计数据存储区的成员,且其中为执行对所述KVS树的所述请求,所述处理电路经配置以将所述键及所述存储大小存储于所述KVS树的数据结构中。

[0408] 在实例9中,根据实例8所述的标的物,其中所述逻辑删除为前缀逻辑删除。

[0409] 在实例10中,根据实例1到9中任一或多个实例所述的标的物,其中所述参数集包含规定所述KVS树为不可变的成员,其中为执行对所述KVS树的所述请求,所述处理电路经配置以将所述请求写入到所述KVS树的根节点。

[0410] 在实例11中,根据实例10所述的标的物,其中当所述KVS树为不可变的时所述KVS树排他地使用键压缩。

[0411] 在实例12中,根据实例11所述的标的物,其中所述处理电路进一步经配置以:响应于所述KVS树为不可变的而存储键搜索统计数据;且响应于所述键搜索统计数据满足阈值而执行键压缩。

[0412] 在实例13中,根据实例12所述的标的物,其中所述键搜索统计数据为最小、最大、平均数或平均值搜索时间中的至少一者。

[0413] 在实例14中,根据实例12到13中任一或多个实例所述的标的物,其中所述键搜索统计数据为所述根节点中的kvset数目。

[0414] 在实例15中,根据实例12到14中任一或多个实例所述的标的物,其中所述处理电路进一步经配置以响应于以下情形中的至少一者而将所述键搜索统计数据复位:压缩、引入、规定数目次搜索之后或规定时间间隔之后。

[0415] 在实例16中,根据实例10到15中任一或多个实例所述的标的物,其中所述参数集的第二成员规定所述KVS树在先进先出基础上移除元素,其中所述参数集的第三成员规定所述KVS树的保留约束,其中所述KVS树基于所述保留约束而对kvset执行键压缩,且其中所述KVS树在违反所述保留约束时移除最旧kvset。

[0416] 在实例17中,根据实例16所述的标的物,其中为基于所述保留约束而对kvset执行键压缩,所述处理电路经配置以:将连续kvset分组以产生群组集,来自所述群组集中的每一成员的经求和指标约计所述保留约束的分数;且对所述群组集的每一成员执行键压缩。

[0417] 在实例18中,根据实例16到17中任一或多个实例所述的标的物,其中所述保留约束为最大键值对数目。

[0418] 在实例19中,根据实例16到18中任一或多个实例所述的标的物,其中所述保留约束为键值对的最大年龄。

[0419] 在实例20中,根据实例16到19中任一或多个实例所述的标的物,其中所述保留约束为由键值对消耗的最大存储值。

[0420] 实例21为至少一个机器可读媒体,其包含在由机器执行时致使所述机器执行包括以下各项的操作的指令:接收对KVS树的请求,所述KVS树为包含节点的数据结构,所述节点包含时间上经定序的kvset序列,所述kvset以经排序次序存储键;接收所述KVS树的参数集;及通过根据参数修改所述KVS树的操作而执行对所述KVS树的所述请求。

[0421] 在实例22中,根据实例21所述的标的物,其中所述请求包含键前缀及逻辑删除,其中所述参数集具有在所述请求中将所述逻辑删除定义为前缀逻辑删除的成员,且其中执行对所述KVS树的所述请求包含将所述前缀逻辑删除写入到所述KVS树的kvset。

[0422] 在实例23中,根据实例22所述的标的物,其中在将键进行比较的KVS树操作时,前缀逻辑删除匹配具有与所述前缀逻辑删除的所述键前缀相同的前缀的任何键。

[0423] 在实例24中,根据实例21到23中任一或多个实例所述的标的物,其中所述请求包含键,其中所述参数集包含规定逻辑删除加速度的成员;且其中所述执行对所述KVS树的所述请求包含将逻辑删除写入于通过对所述键执行溢出函数而规定的至少一个子节点中。

[0424] 在实例25中,根据实例24所述的标的物,其中将所述逻辑删除写入到通过对所述键执行所述溢出函数而规定的所有现存子节点。

[0425] 在实例26中,根据实例24到25中任一或多个实例所述的标的物,其中所述请求包含逻辑删除。

[0426] 在实例27中,根据实例24到26中任一或多个实例所述的标的物,其中所述请求包含值。

[0427] 在实例28中,根据实例21到27中任一或多个实例所述的标的物,其中所述请求包含键、逻辑删除及在所述KVS树中与所述键对应的值的存储大小,其中所述参数集具有规定无用单元收集统计数据存储区的成员,且其中执行对所述KVS树的所述请求包含将所述

键及所述存储大小存储于所述KVS树的数据结构中。

[0428] 在实例29中,根据实例28所述的标的物,其中所述逻辑删除为前缀逻辑删除。

[0429] 在实例30中,根据实例21到29中任一或多个实例所述的标的物,其中所述参数集包含规定所述KVS树为不可变的成员,其中执行对所述KVS树的所述请求包含将所述请求写入到所述KVS树的根节点。

[0430] 在实例31中,根据实例30所述的标的物,其中当所述KVS树为不可变的时所述KVS树排他地使用键压缩。

[0431] 在实例32中,根据实例31所述的标的物,其中所述操作包括:响应于所述KVS树为不可变的而存储键搜索统计数据;及响应于所述键搜索统计数据满足阈值而执行键压缩。

[0432] 在实例33中,根据实例32所述的标的物,其中所述键搜索统计数据为最小、最大、平均数或平均值搜索时间中的至少一者。

[0433] 在实例34中,根据实例32到33中任一或多个实例所述的标的物,其中所述键搜索统计数据为所述根节点中的kvset数目。

[0434] 在实例35中,根据实例32到34中任一或多个实例所述的标的物,其中所述操作包括响应于以下情形中的至少一者而将所述键搜索统计数据复位:压缩、引入、规定数目次搜索之后或规定时间间隔之后。

[0435] 在实例36中,根据实例30到35中任一或多个实例所述的标的物,其中所述参数集的第二成员规定所述KVS树在先进先出基础上移除元素,其中所述参数集的第三成员规定所述KVS树的保留约束,其中所述KVS树基于所述保留约束而对kvset执行键压缩,且其中所述KVS树在违反所述保留约束时移除最旧kvset。

[0436] 在实例37中,根据实例36所述的标的物,其中基于所述保留约束而对kvset执行键压缩包含:将连续kvset分组以产生群组集,来自所述群组集中的每一成员的经求和指标约计所述保留约束的分数;且对所述群组集的每一成员执行键压缩。

[0437] 在实例38中,根据实例36到37中任一或多个实例所述的标的物,其中所述保留约束为最大键值对数目。

[0438] 在实例39中,根据实例36到38中任一或多个实例所述的标的物,其中所述保留约束为键值对的最大年龄。

[0439] 在实例40中,根据实例36到39中任一或多个实例所述的标的物,其中所述保留约束为由键值对消耗的最大存储值。

[0440] 实例41为一种机器实施的方法,其包括:接收对KVS树的请求,所述KVS树为包含节点的数据结构,所述节点包含时间上经定序的kvset序列,所述kvset以经排序次序存储键;接收所述KVS树的参数集;及通过根据参数修改所述KVS树的操作而执行对所述KVS树的所述请求。

[0441] 在实例42中,根据实例41所述的标的物,其中所述请求包含键前缀及逻辑删除,其中所述参数集具有在所述请求中将所述逻辑删除定义为前缀逻辑删除的成员,且其中执行对所述KVS树的所述请求包含将所述前缀逻辑删除写入到所述KVS树的kvset。

[0442] 在实例43中,根据实例42所述的标的物,其中在将键进行比较的KVS树操作时,前缀逻辑删除匹配具有与所述前缀逻辑删除的所述键前缀相同的前缀的任何键。

[0443] 在实例44中,根据实例41到43中任一或多个实例所述的标的物,其中所述请求

包含键,其中所述参数集包含规定逻辑删除加速度的成员;且其中所述执行对所述KVS树的所述请求包含将逻辑删除写入于通过对所述键执行溢出函数而规定的至少一个子节点中。

[0444] 在实例45中,根据实例44所述的标的物,其中将所述逻辑删除写入到通过对所述键执行所述溢出函数而规定的所有现存子节点。

[0445] 在实例46中,根据实例44到45中任一或多个实例所述的标的物,其中所述请求包含逻辑删除。

[0446] 在实例47中,根据实例44到46中任一或多个实例所述的标的物,其中所述请求包含值。

[0447] 在实例48中,根据实例41到47中任一或多个实例所述的标的物,其中所述请求包含键、逻辑删除及在所述KVS树中与所述键对应的值的存储大小,其中所述参数集具有规定无用单元收集统计数据存储区的成员,且其中执行对所述KVS树的所述请求包含将所述键及所述存储大小存储于所述KVS树的数据结构中。

[0448] 在实例49中,根据实例48所述的标的物,其中所述逻辑删除为前缀逻辑删除。

[0449] 在实例50中,根据实例41到49中任一或多个实例所述的标的物,其中所述参数集包含规定所述KVS树为不可变的成员,其中执行对所述KVS树的所述请求包含将所述请求写入到所述KVS树的根节点。

[0450] 在实例51中,根据实例50所述的标的物,其中当所述KVS树为不可变的时所述KVS树排他地使用键压缩。

[0451] 在实例52中,根据实例51所述的标的物任选地包含:响应于所述KVS树为不可变的而存储键搜索统计数据;及响应于所述键搜索统计数据满足阈值而执行键压缩。

[0452] 在实例53中,根据实例52所述的标的物,其中所述键搜索统计数据为最小、最大、平均数或平均值搜索时间中的至少一者。

[0453] 在实例54中,根据实例52到53中任一或多个实例所述的标的物,其中所述键搜索统计数据为所述根节点中的kvset数目。

[0454] 在实例55中,根据实例52到54中任一或多个实例所述的标的物任选地包含响应于以下情形中的至少一者而将所述键搜索统计数据复位:压缩、引入、规定数目次搜索之后或规定时间间隔之后。

[0455] 在实例56中,根据实例50到55中任一或多个实例所述的标的物,其中所述参数集的第二成员规定所述KVS树在先进先出基础上移除元素,其中所述参数集第三成员规定所述KVS树的保留约束,其中所述KVS树基于所述保留约束而对kvset执行键压缩,且其中所述KVS树在违反所述保留约束时移除最旧kvset。

[0456] 在实例57中,根据实例56所述的标的物,其中基于所述保留约束而对kvset执行键压缩包含:将连续kvset分组以产生群组集,来自所述群组集中的每一成员的经求和指标约计所述保留约束的分数;且对所述群组集的每一成员执行键压缩。

[0457] 在实例58中,根据实例56到57中任一或多个实例所述的标的物,其中所述保留约束为最大键值对数目。

[0458] 在实例59中,根据实例56到58中任一或多个实例所述的标的物,其中所述保留约束为键值对的最大年龄。

[0459] 在实例60中,根据实例56到59中任一或多个实例所述的标的物,其中所述保留

约束为由键值对消耗的最大存储值。

[0460] 实例61为一种系统,其包括:用于接收对KVS树的请求的构件,所述KVS树为包含节点的数据结构,所述节点包含时间上经定序的kvset序列,所述kvset以经排序次序存储键;用于接收所述KVS树的参数集的构件;及用于通过根据参数修改所述KVS树的操作而执行对所述KVS树的所述请求的构件。

[0461] 在实例62中,根据实例61所述的标的物,其中所述请求包含键前缀及逻辑删除,其中所述参数集具有在所述请求中将所述逻辑删除定义为前缀逻辑删除的成员,且其中执行对所述KVS树的所述请求包含将所述前缀逻辑删除写入到所述KVS树的kvset。

[0462] 在实例63中,根据实例62所述的标的物,其中在将键进行比较的KVS树操作时,前缀逻辑删除匹配具有与所述前缀逻辑删除的所述键前缀相同的前缀的任何键。

[0463] 在实例64中,根据实例61到63中任何一或多个实例所述的标的物,其中所述请求包含键,其中所述参数集包含规定逻辑删除加速度的成员;且其中所述执行对所述KVS树的所述请求包含将逻辑删除写入于通过对所述键执行溢出函数而规定的至少一个子节点中。

[0464] 在实例65中,根据实例64所述的标的物,其中将所述逻辑删除写入到通过对所述键执行所述溢出函数而规定的所有现存子节点。

[0465] 在实例66中,根据实例64到65中任何一或多个实例所述的标的物,其中所述请求包含逻辑删除。

[0466] 在实例67中,根据实例64到66中任何一或多个实例所述的标的物,其中所述请求包含值。

[0467] 在实例68中,根据实例61到67中任何一或多个实例所述的标的物,其中所述请求包含键、逻辑删除及在所述KVS树中与所述键对应的值的存储大小,其中所述参数集具有规定无用单元收集统计数据存储区的成员,且其中执行对所述KVS树的所述请求包含将所述键及所述存储大小存储于所述KVS树的数据结构中。

[0468] 在实例69中,根据实例68所述的标的物,其中所述逻辑删除为前缀逻辑删除。

[0469] 在实例70中,根据实例61到69中任何一或多个实例所述的标的物,其中所述参数集包含规定所述KVS树为不可变的成员,其中执行对所述KVS树的所述请求包含将所述请求写入到所述KVS树的根节点。

[0470] 在实例71中,根据实例70所述的标的物,其中当所述KVS树为不可变的时所述KVS树排他地使用键压缩。

[0471] 在实例72中,根据实例71所述的标的物任选地包含:用于响应于所述KVS树为不可变的而存储键搜索统计数据的构件;及用于响应于所述键搜索统计数据满足阈值而执行键压缩的构件。

[0472] 在实例73中,根据实例72所述的标的物,其中所述键搜索统计数据为最小、最大、平均数或平均值搜索时间中的至少一者。

[0473] 在实例74中,根据实例72到73中任何一或多个实例所述的标的物,其中所述键搜索统计数据为所述根节点中的kvset数目。

[0474] 在实例75中,根据实例72到74中任何一或多个实例所述的标的物任选地包含用于响应于以下情形中的至少一者而将所述键搜索统计数据复位的构件:压缩、引入、规定数目次搜索之后或规定时间间隔之后。

[0475] 在实例76中,根据实例70到75中任一或多个实例所述的标的物,其中所述参数集的第二成员规定所述KVS树在先进先出基础上移除元素,其中所述参数集的第三成员规定所述KVS树的保留约束,其中所述KVS树基于所述保留约束而对kvset执行键压缩,且其中所述KVS树在违反所述保留约束时移除最旧kvset。

[0476] 在实例77中,根据实例76所述的标的物,其中基于所述保留约束而对kvset执行键压缩包含:将连续kvset分组以产生群组集,来自所述群组集中的每一成员的经求和指标约计所述保留约束的分数;且对所述群组集的每一成员执行键压缩。

[0477] 在实例78中,根据实例76到77中任一或多个实例所述的标的物,其中所述保留约束为最大键值对数目。

[0478] 在实例79中,根据实例76到78中任一或多个实例所述的标的物,其中所述保留约束为键值对的最大年龄。

[0479] 在实例80中,根据实例76到79中任一或多个实例所述的标的物,其中所述保留约束为由键值对消耗的最大存储值。

[0480] 以上详细说明包含对形成所述详细说明的一部分的附图的参考。图式以图解说明的方式展示可实践的特定实施例。这些实施例在本文中还称为“实例”。除了所展示或所描述的所述元件之外,这些实例还可包含若干元件。然而,本发明人还预期其中仅提供所展示或所描述的所述元件的实例。此外,本发明人还预期使用关于特定实例(或者其一或多个方面)或关于本文中所展示或所描述的其它实例(或者其一或多个方面)而展示或描述的所述元件的任何组合或排列的实例(或者其一或多个方面)。

[0481] 此文件中所提及的所有公开案、专利及专利文件将其全文以引用方式并入本文中,就像个别地以引用方式并入一样。倘若本文件与以引用方式如此并入的所述文件之间的使用不一致,那么所并入的参考文献中的使用应被视为对本文件的所述使用的补充;对于不可调和的不一致性,以本文件中的使用为准。

[0482] 在本文件中,如在专利文件中常见,使用术语“一(a或an)”来包含一个或一个以上,独立于“至少一个(at least one)”或“一或多个(one或more)”的任何其它例子或使用。在本文件中,使用术语“或(or)”来是指非排他性,或使得“A或B”包含“A但非B”、“B但非A”及“A及B”,除非另有指示。在所附权利要求书中,将术语“包含(including)”及“其中(in which)”用作相应术语“包括(comprising)”及“其中(wherein)”的普通英语等效形式。此外,在所附权利要求书中,术语“包含(including)”及“包括(comprising)”为开放式的,也就是说,在权利要求中除列于此术语之后的所述元件以外还包含若干元件的系统、装置、物件或过程仍被视为归属于所述权利要求的范围内。此外,在所附权利要求书中,术语“第一(first)”、“第二(second)”及“第三(third)”等仅用作标签,且不打算对其客体强加数字要求。

[0483] 上文说明打算为说明性而非限制性的。举例来说,上文所描述的实例(或者其一或多个方面)可以彼此组合方式使用。例如,所属领域的技术人员可基于审阅上文说明而使用其它实施例。摘要为用以允许读者迅速地确定技术揭示内容的本质且是基于以下理解而提交:其将不用于解释或限制权利要求书的范围或含义。而且,在以上实施方式中,各种特征可分组在一起以简化本发明。此应被解释为预计未主张的所揭示特征对于任一权利要求为必要的。更确切来说,发明标的物可在于少于特定所揭示实施例的所有特征。因此,特此将

所附权利要求书并入到实施方式中,其中每一权利要求独立地作为单独实施例。实施例的范围应参考所附权利要求书连同此权利要求书所授权的等效物的全部范围来确定。

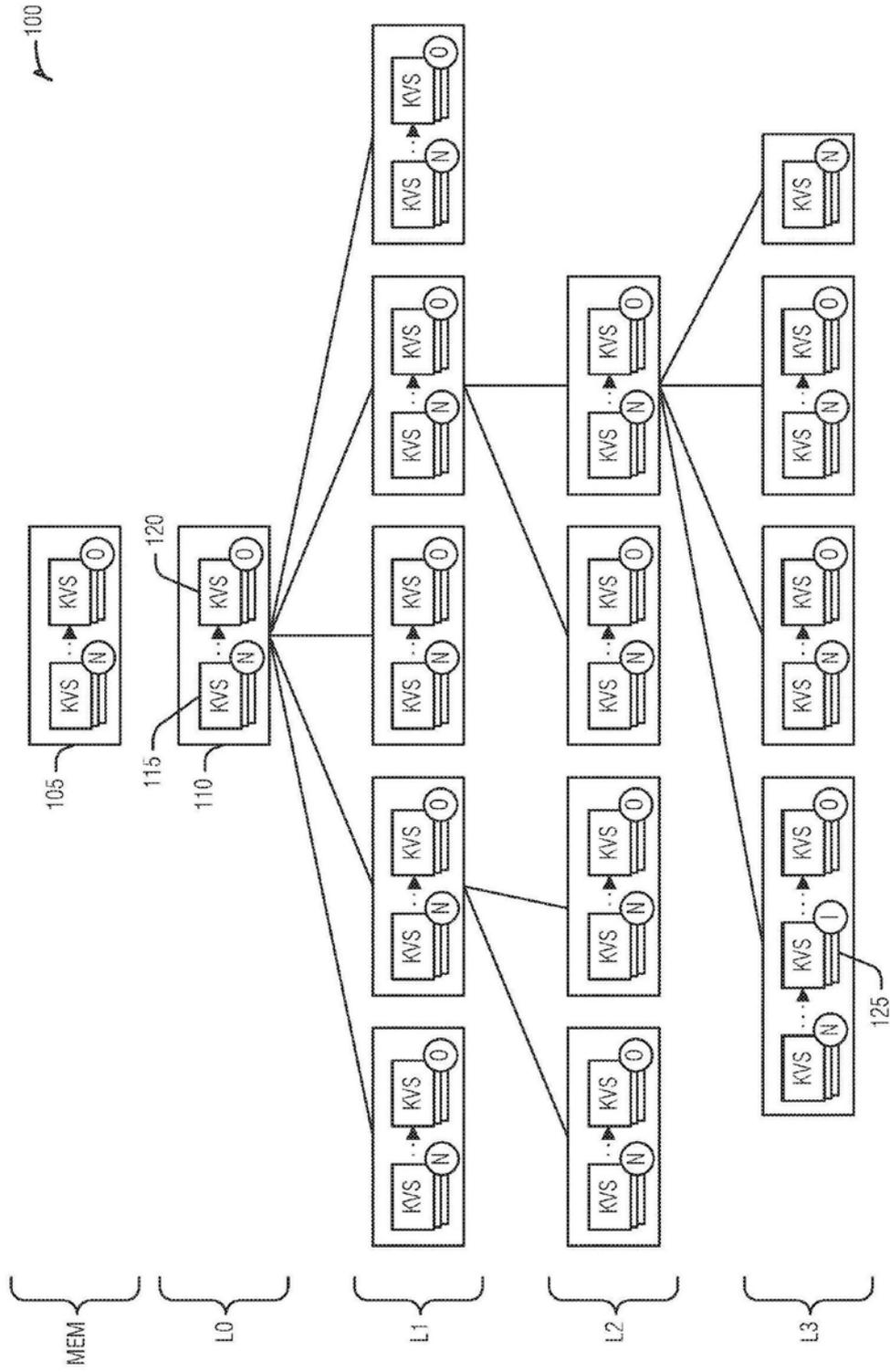


图1

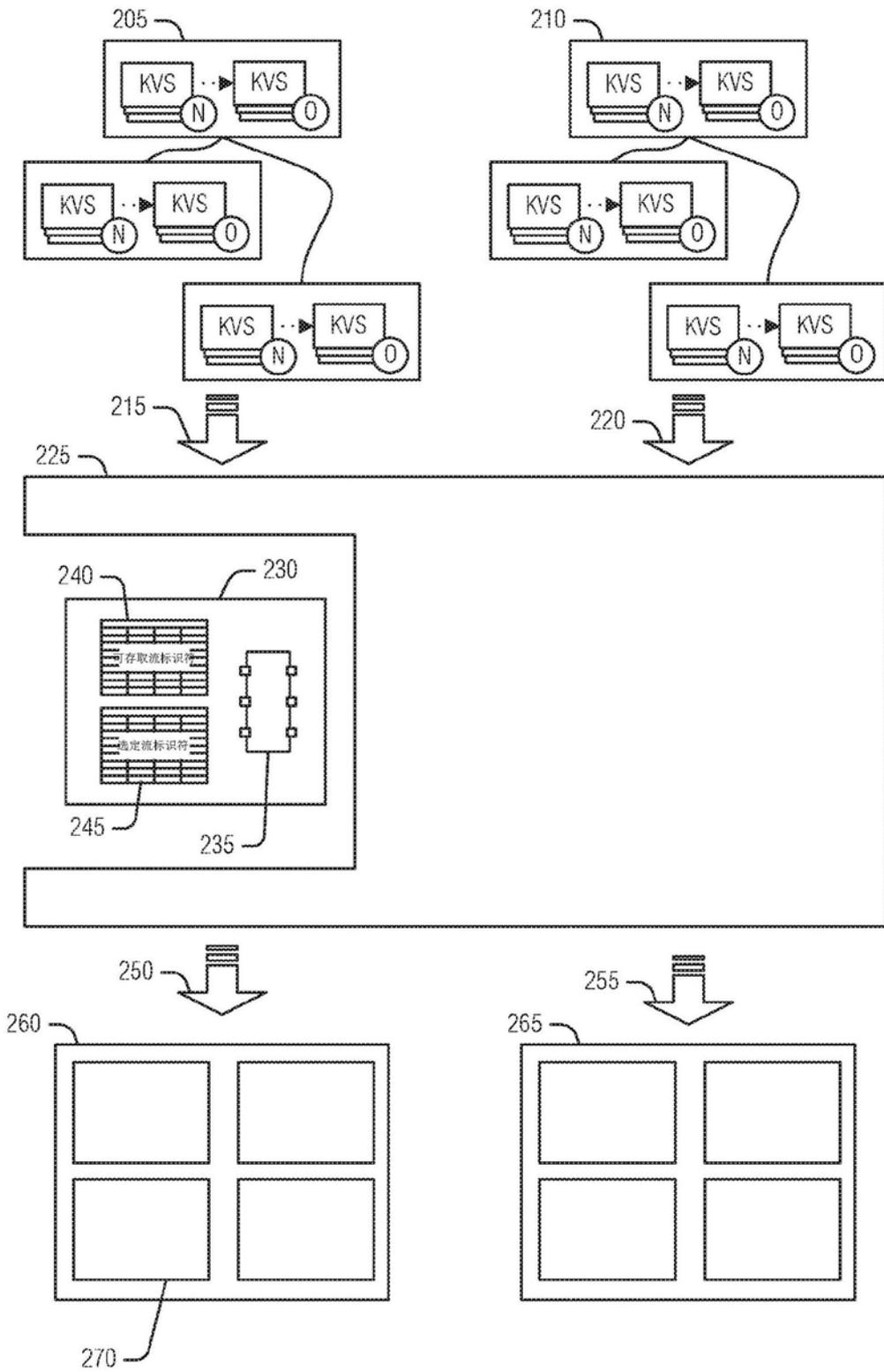


图2

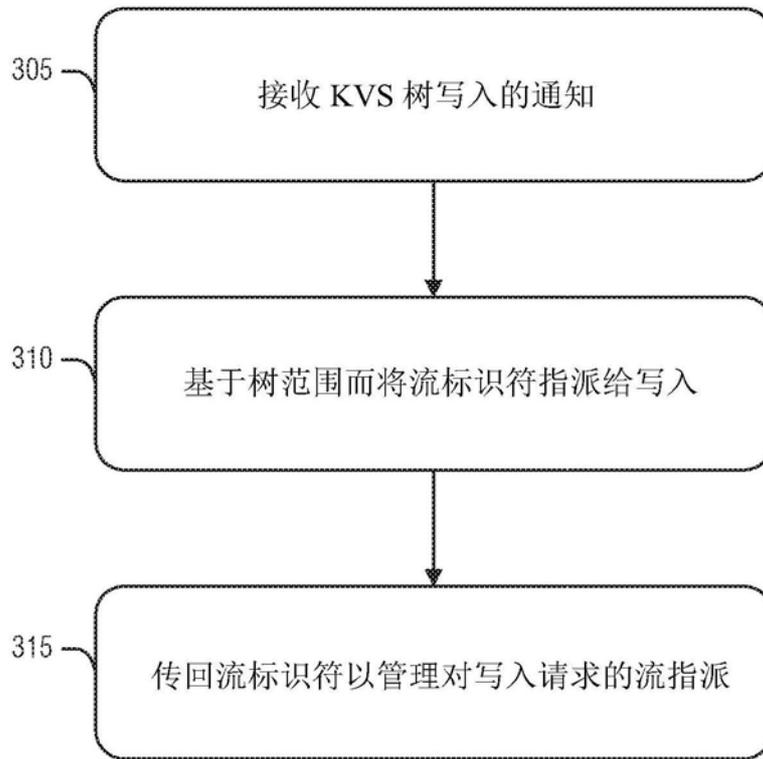


图3

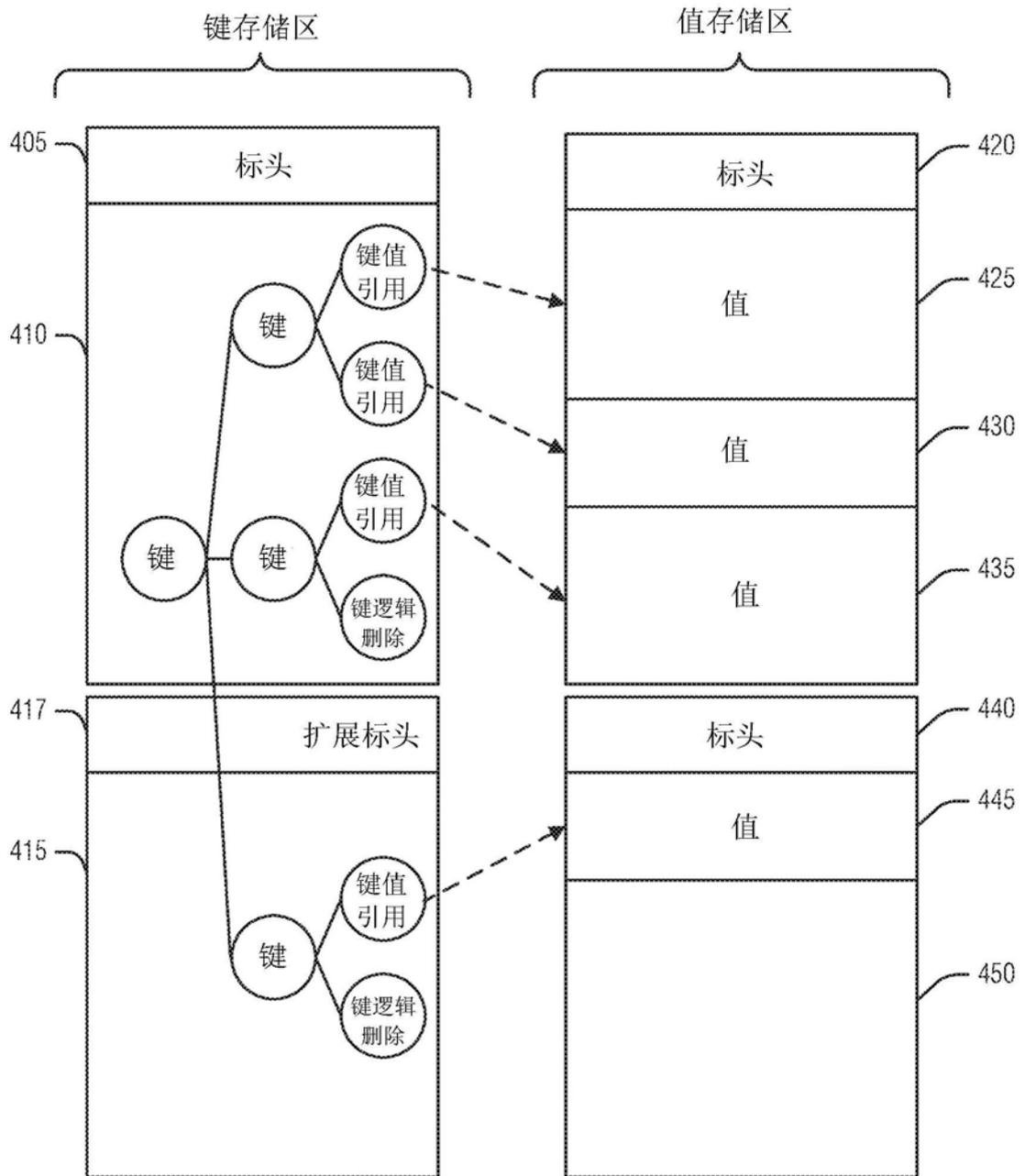


图4

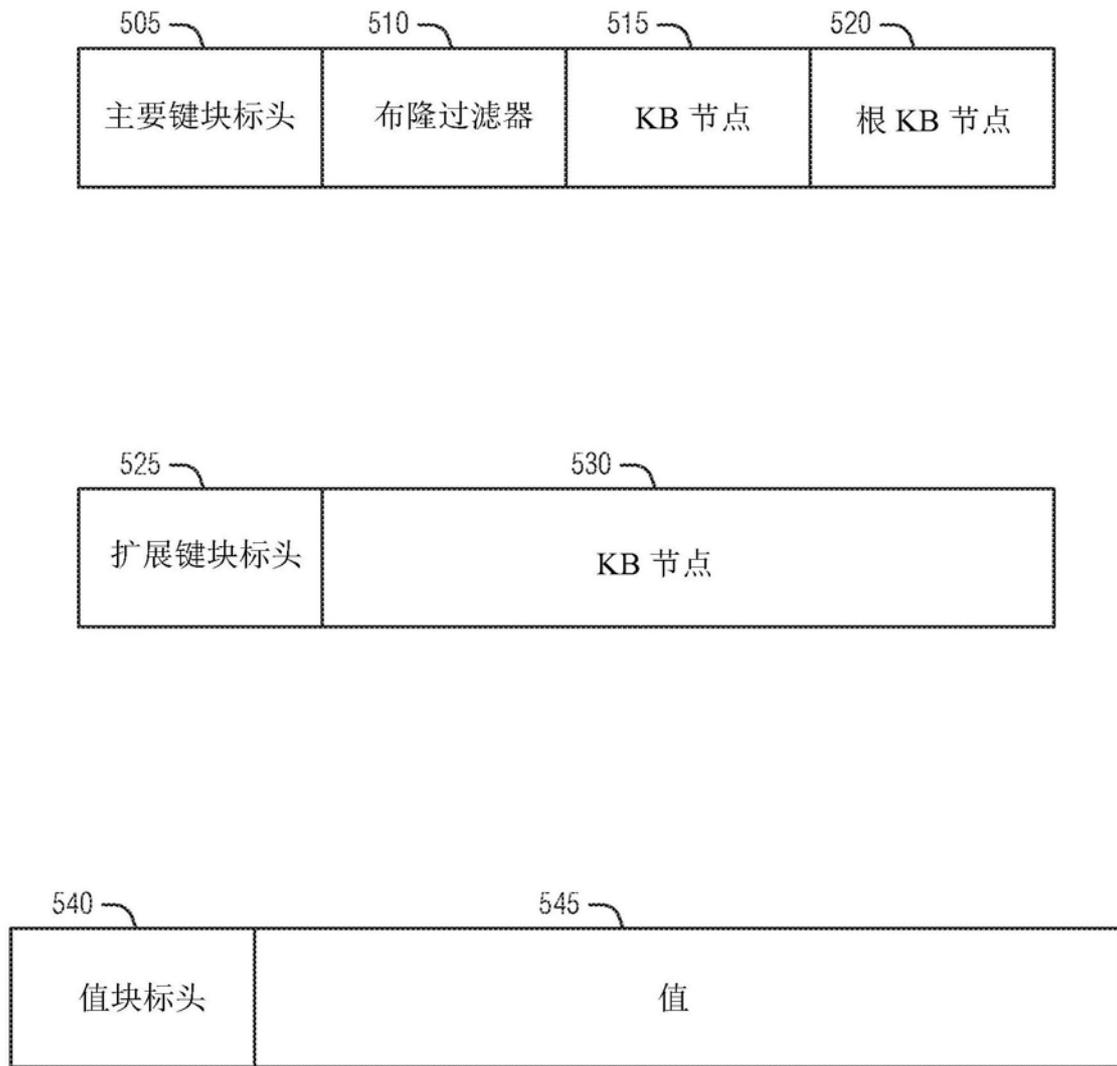


图5

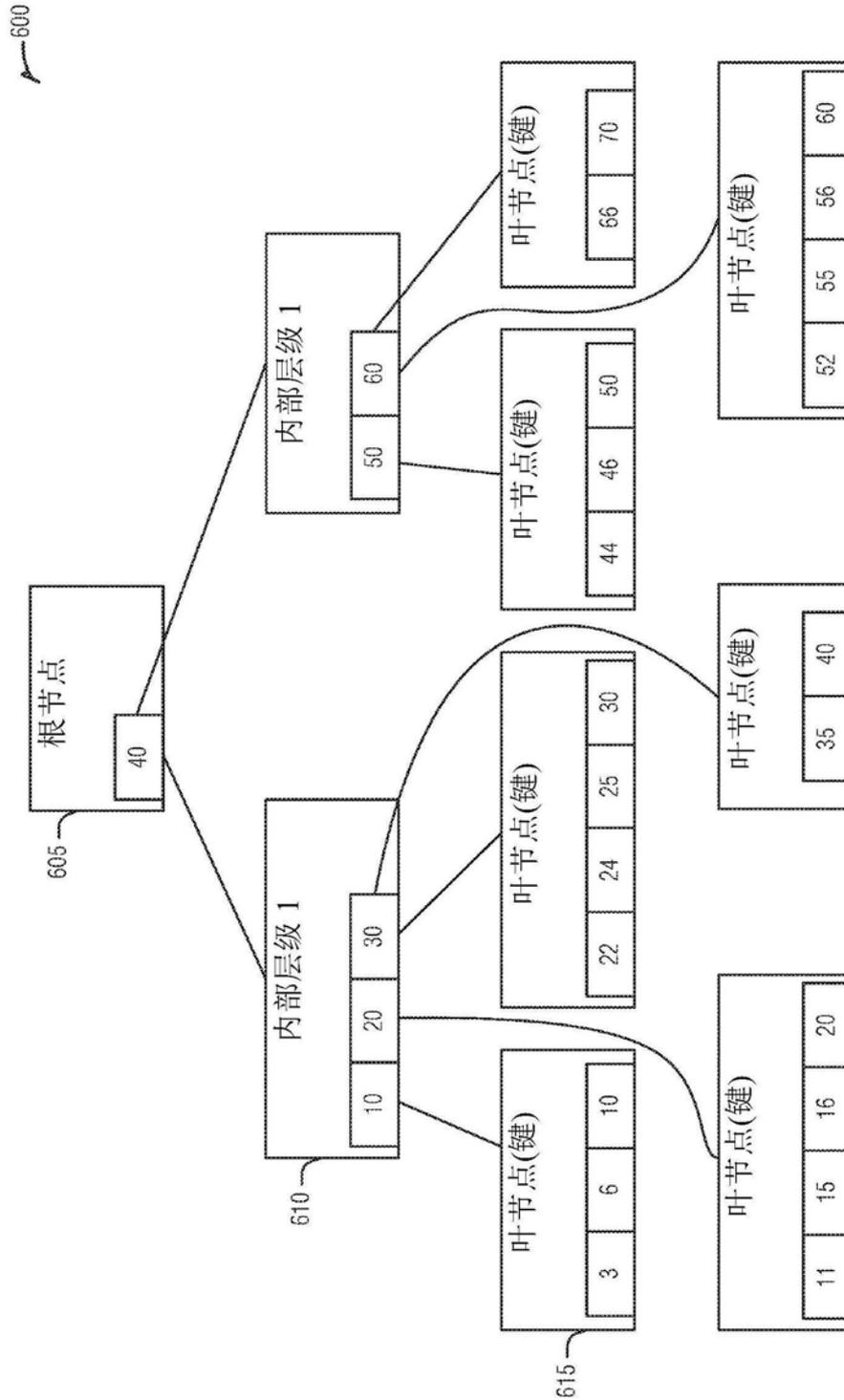


图6

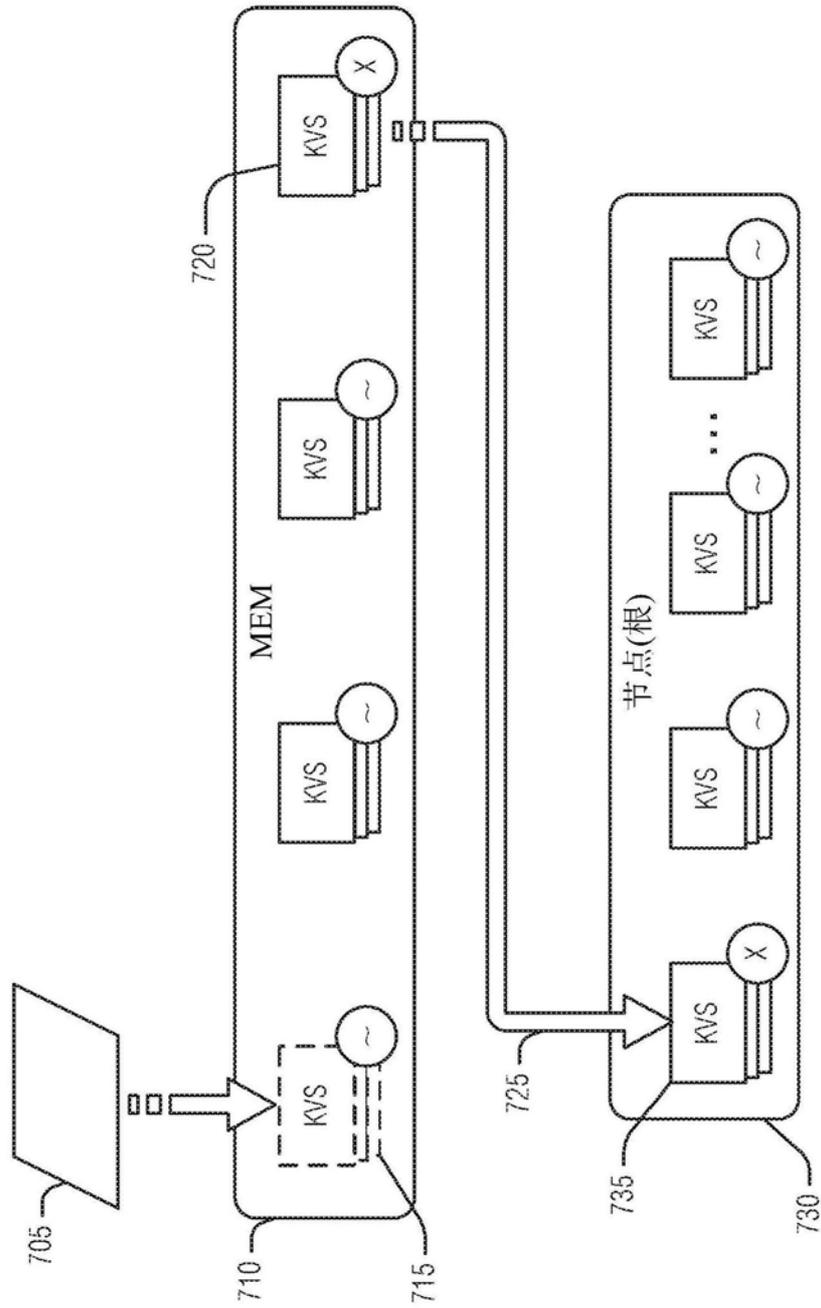


图7

800

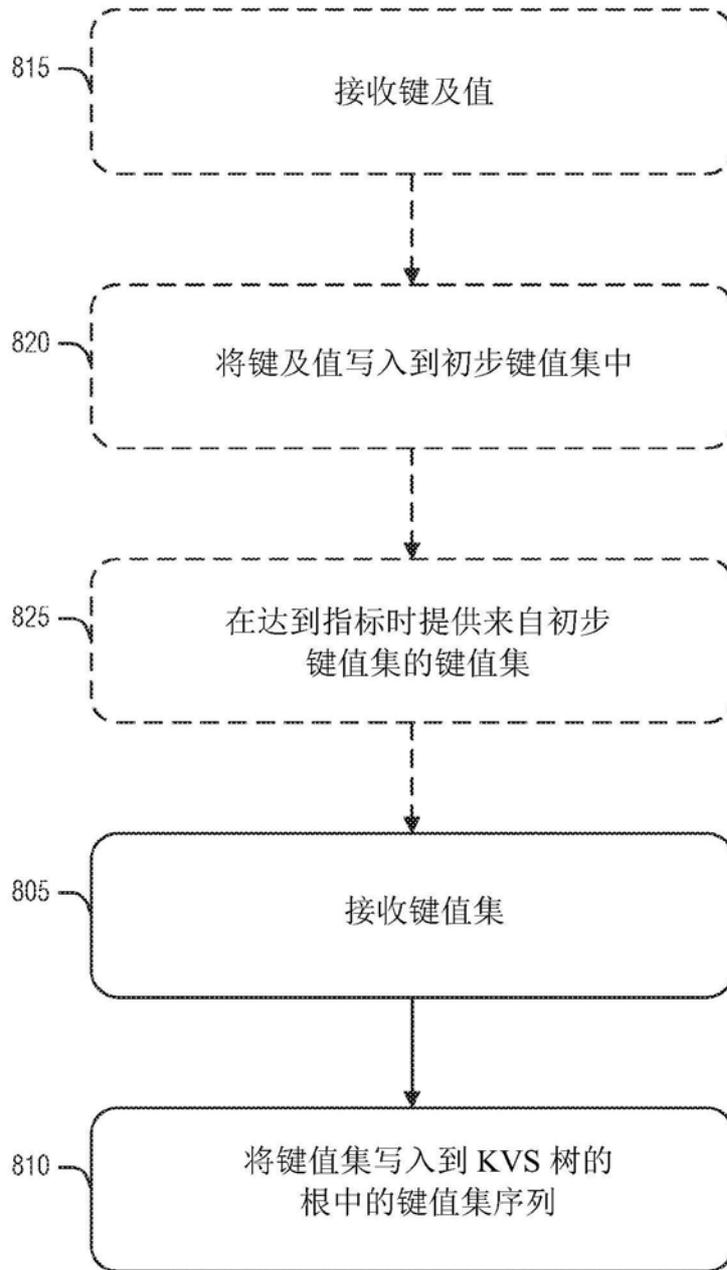


图8

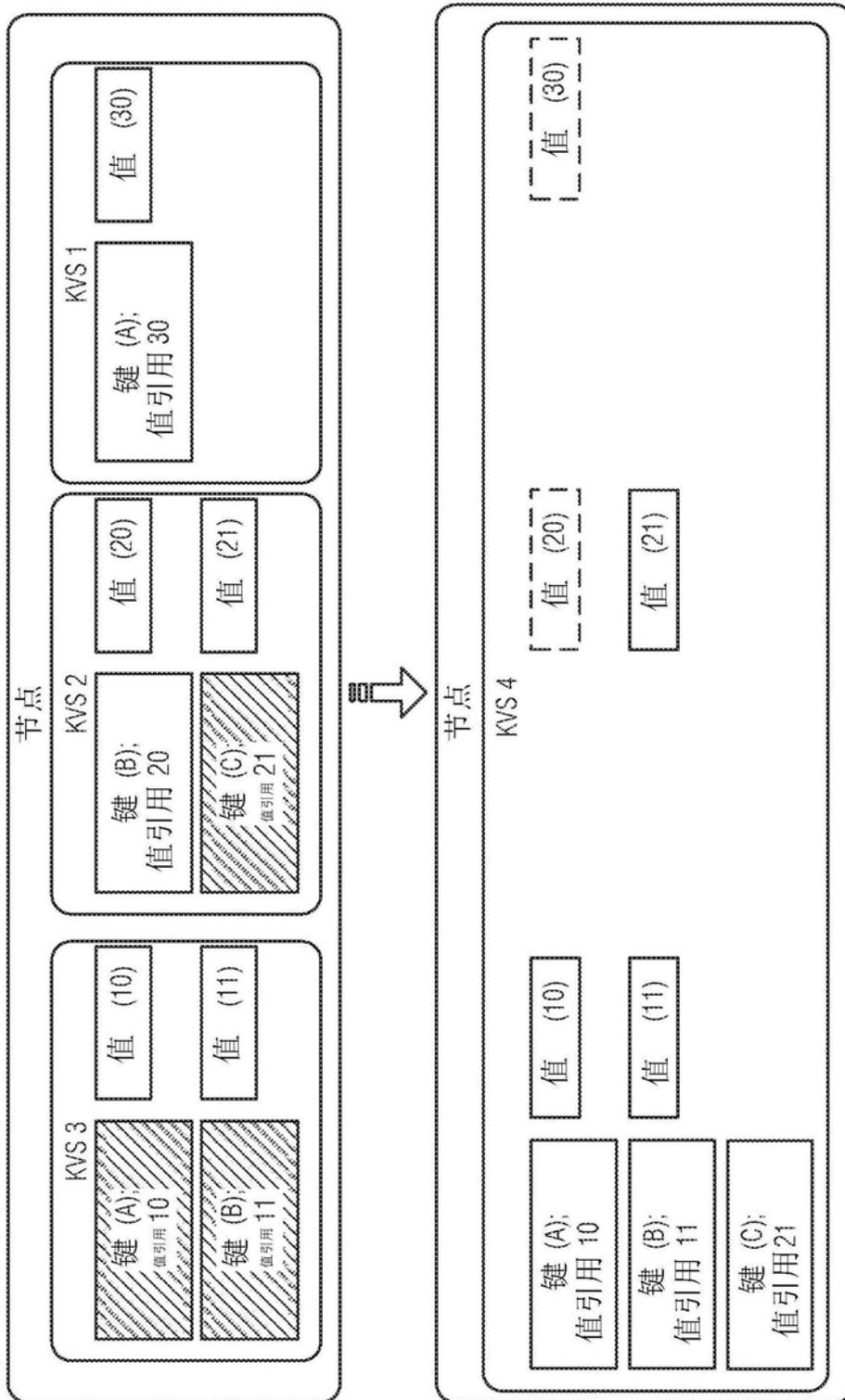


图9

1000

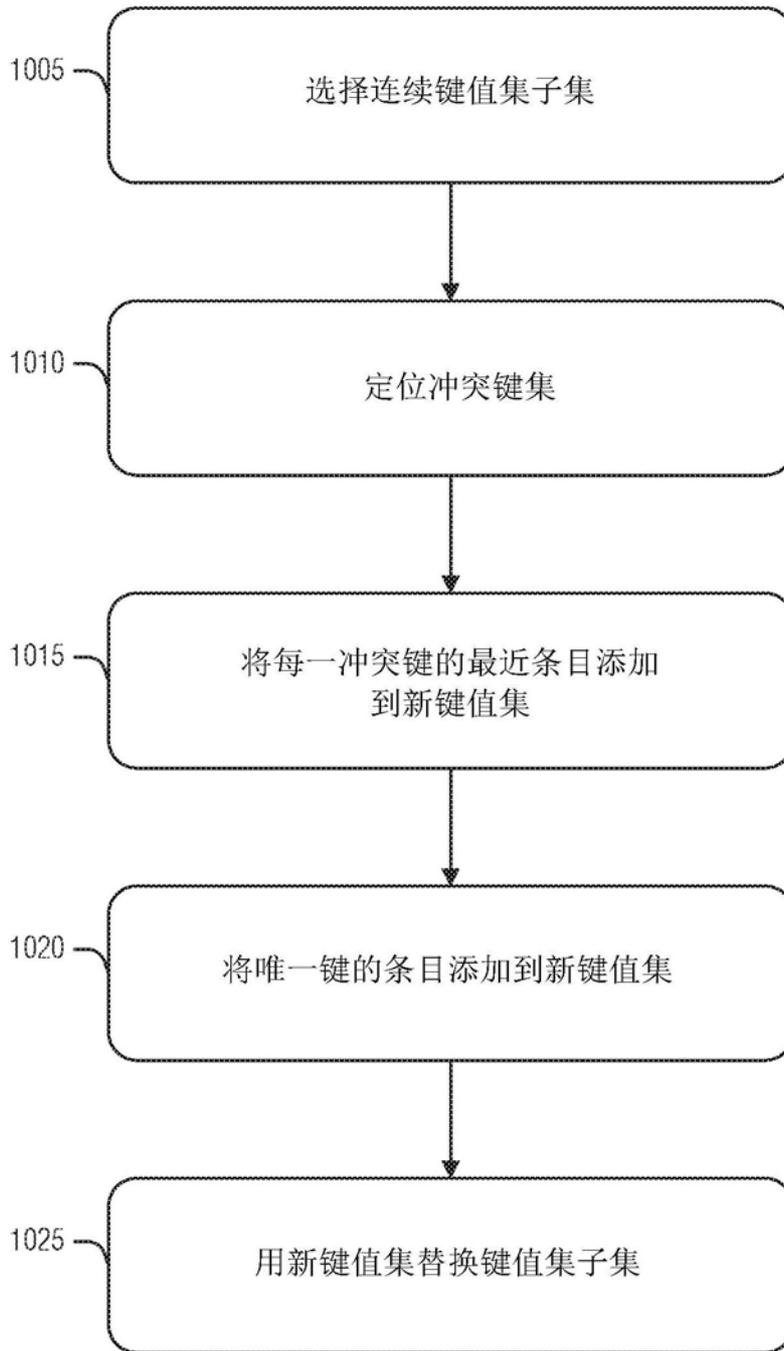


图10

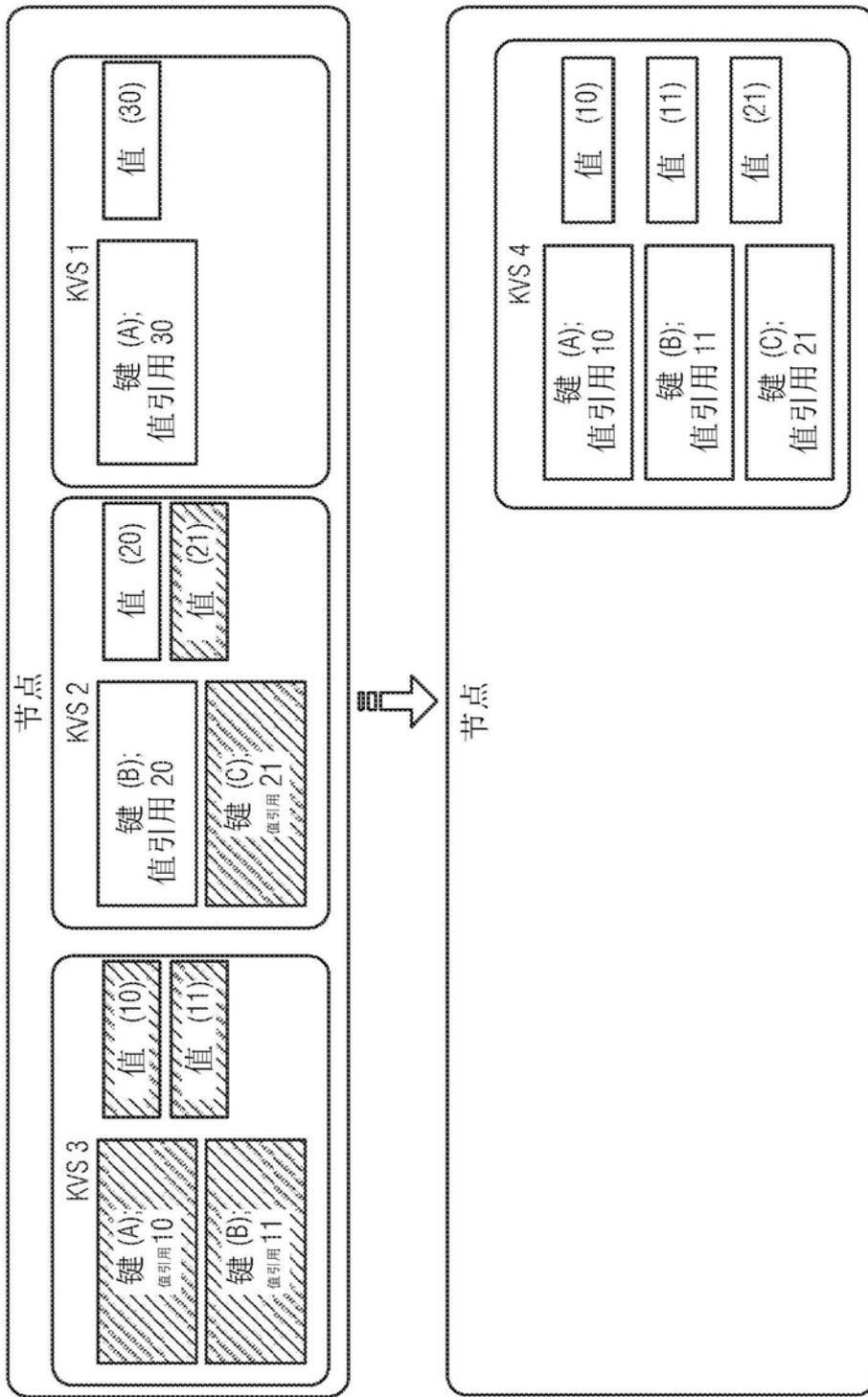


图11

1200

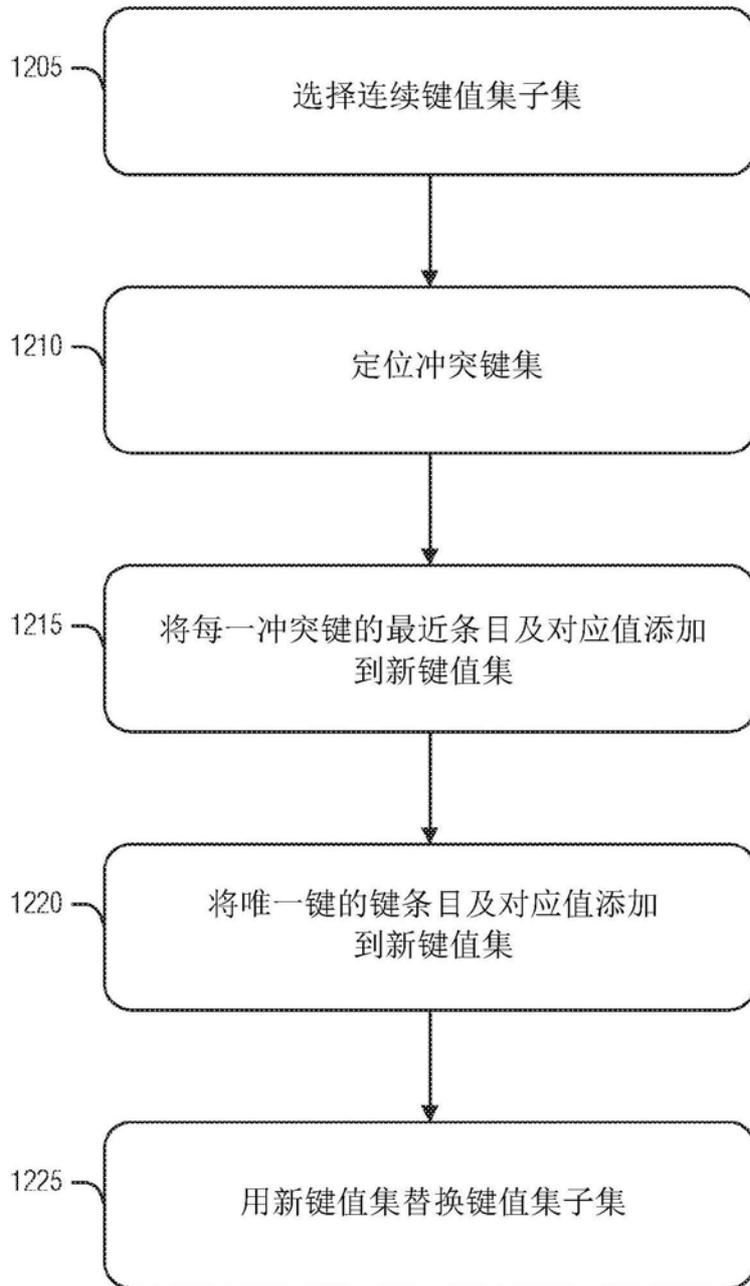


图12

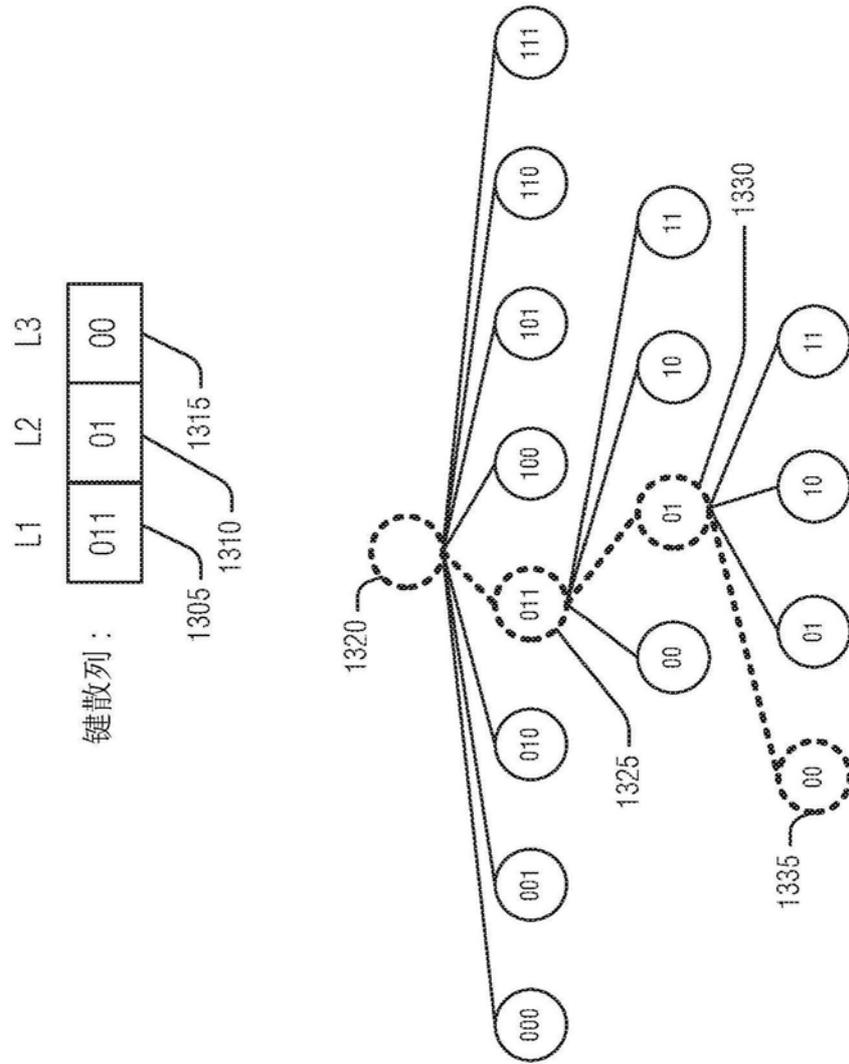


图13

1400

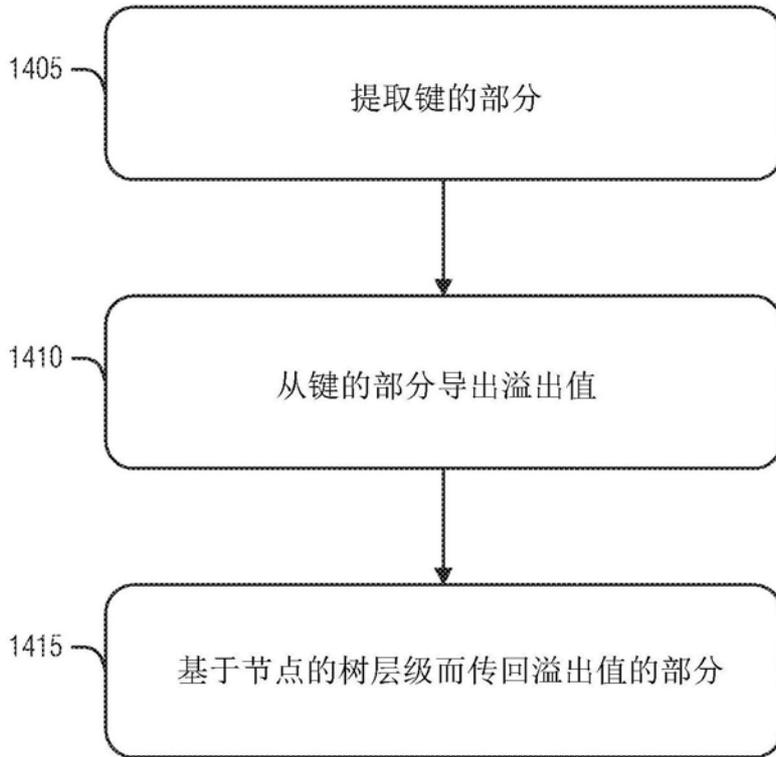


图14

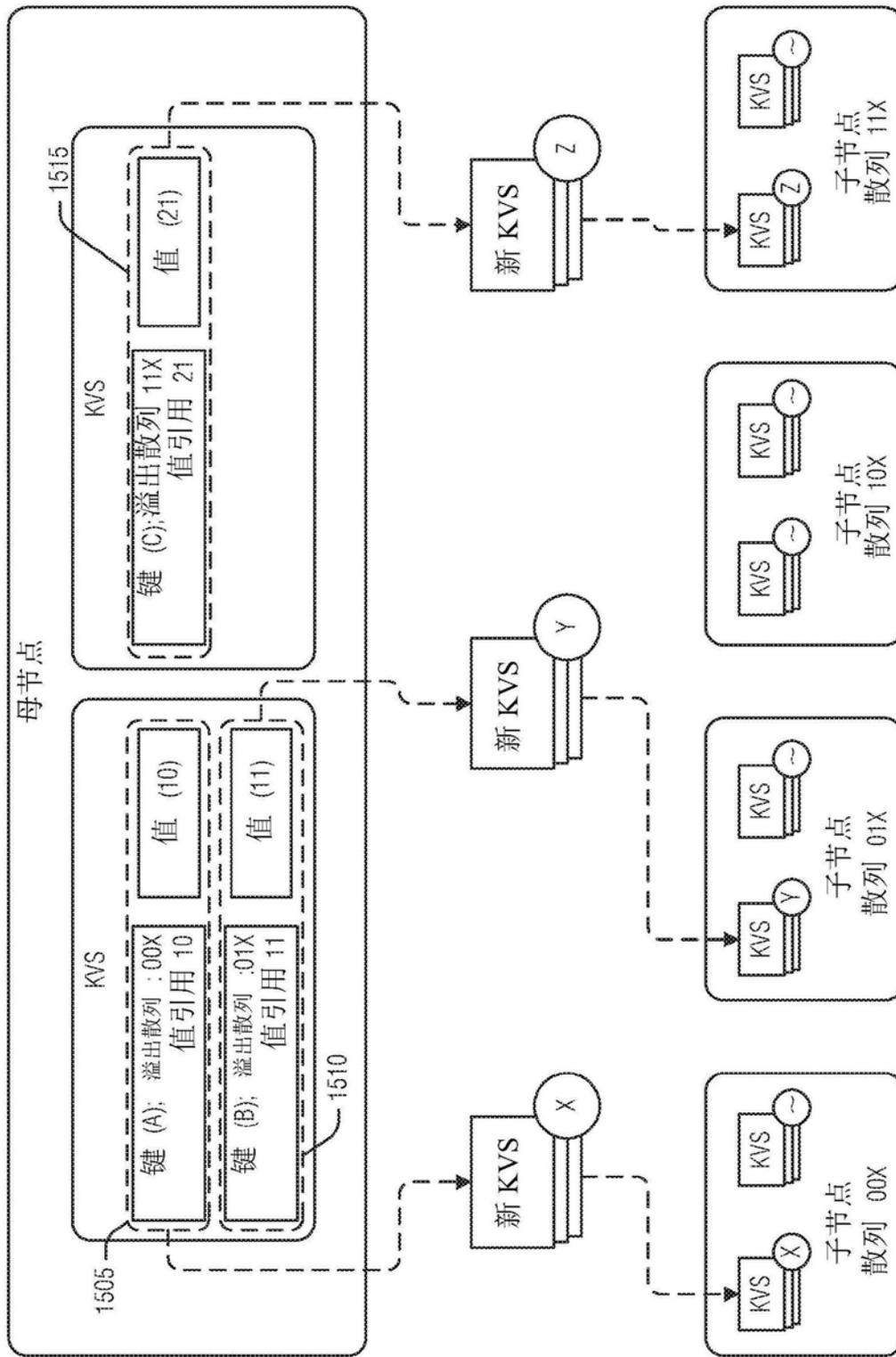


图15

1600

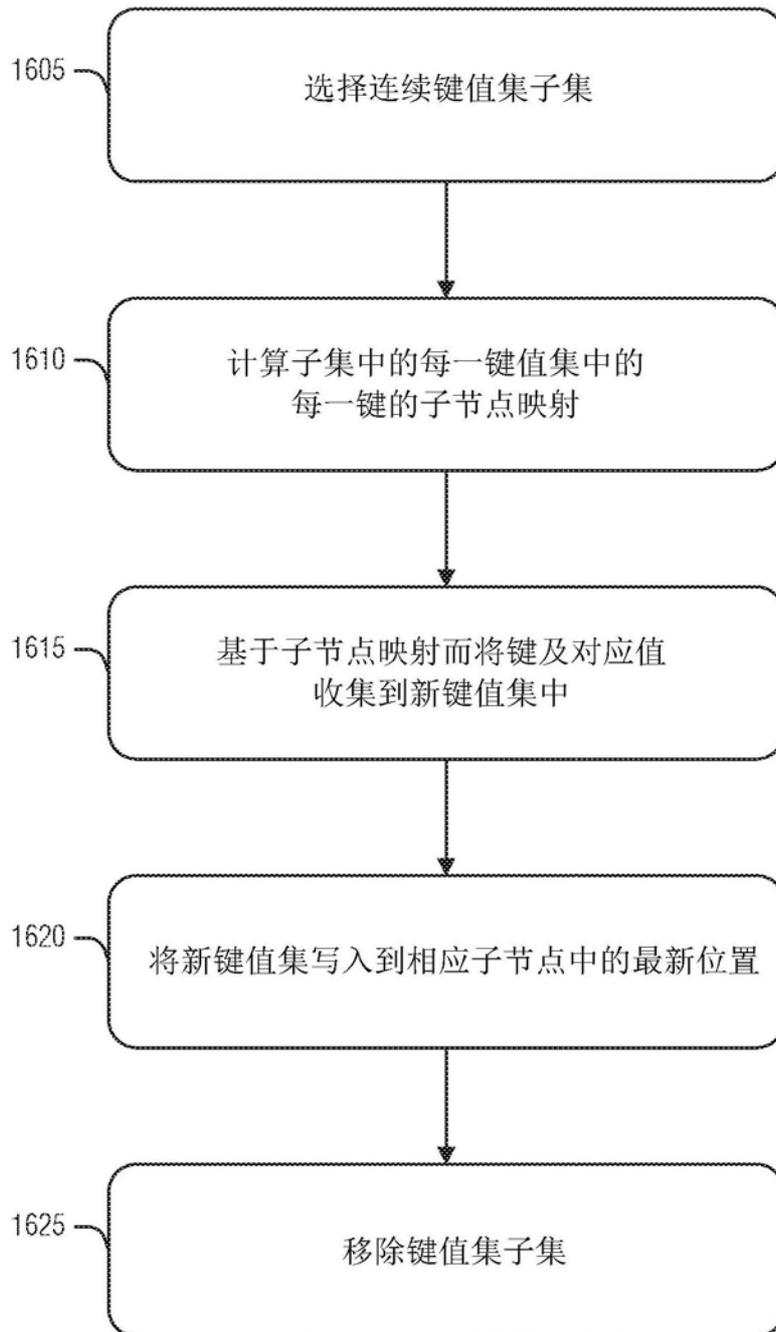


图16

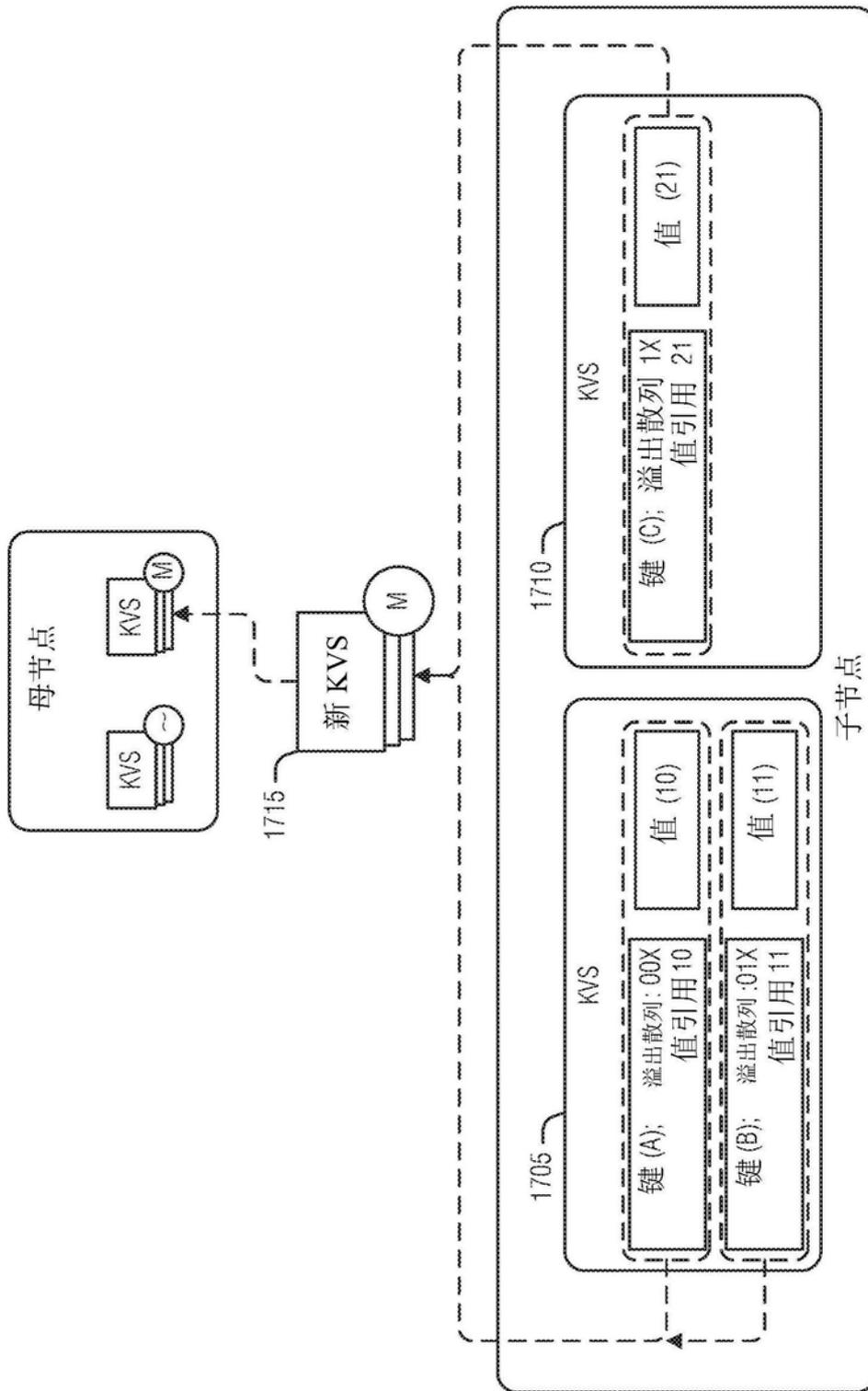


图17

1800

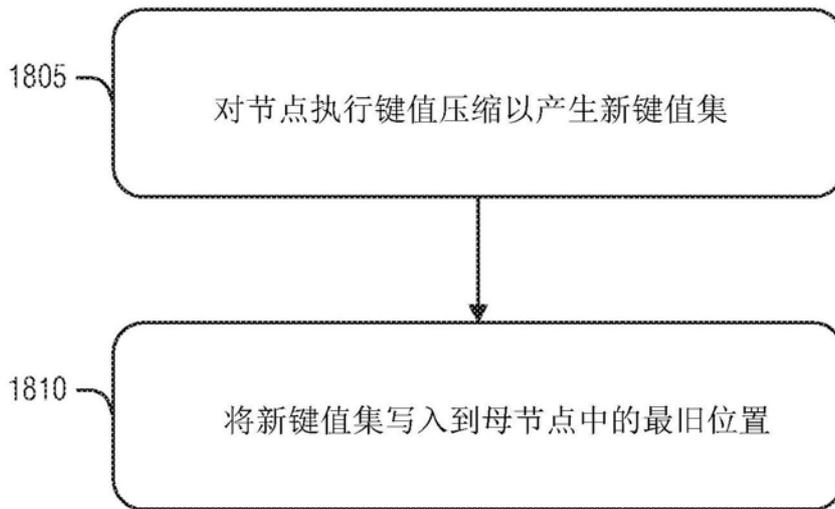


图18

1900

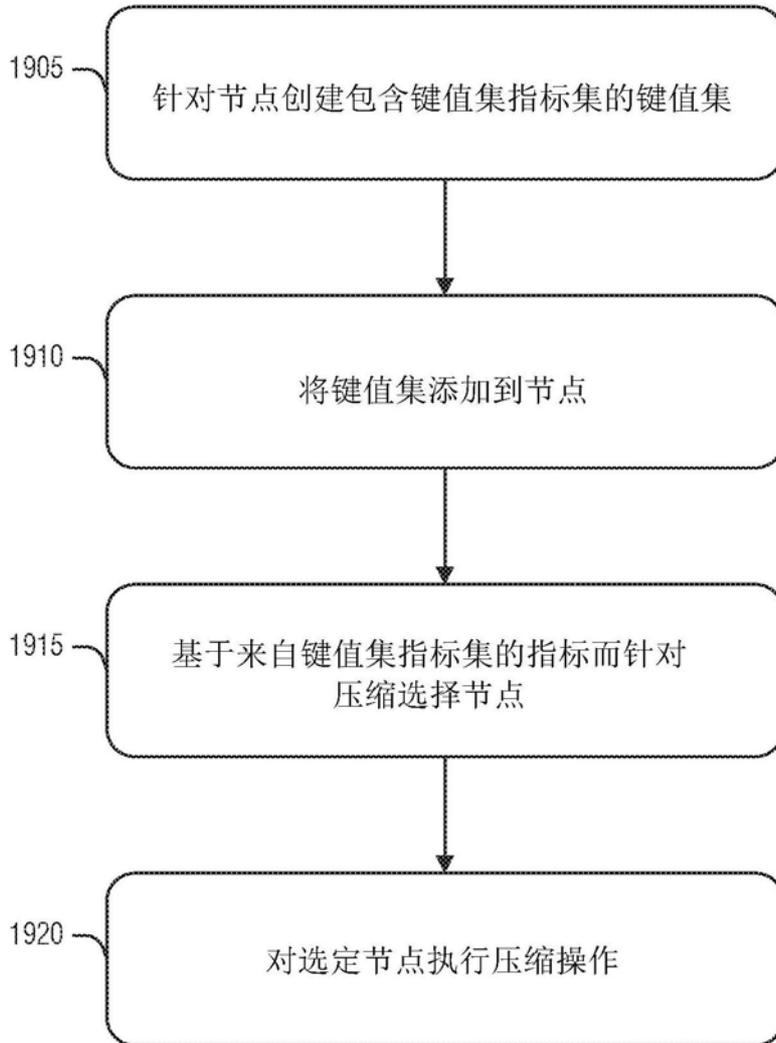


图19

2000

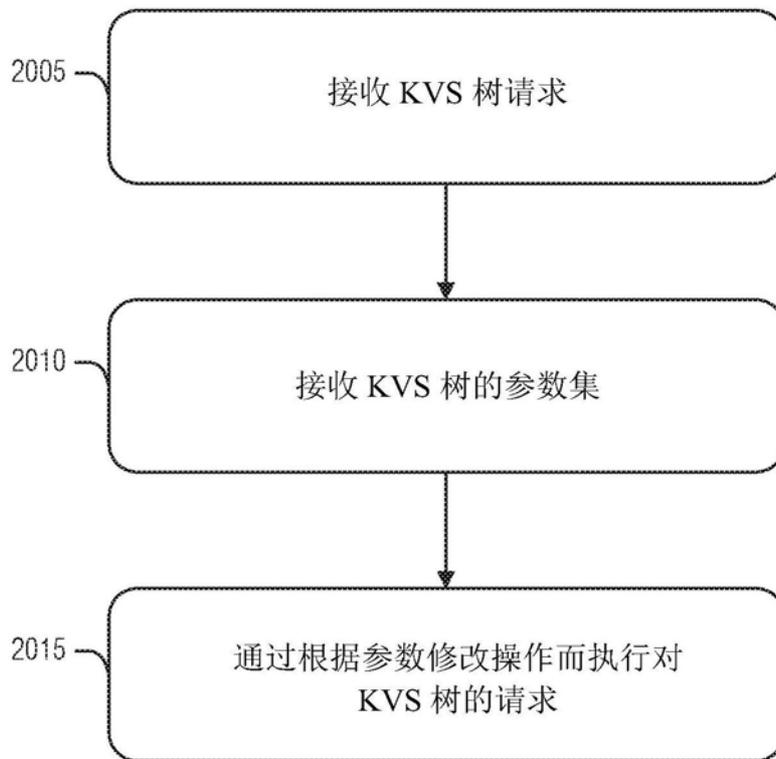


图20

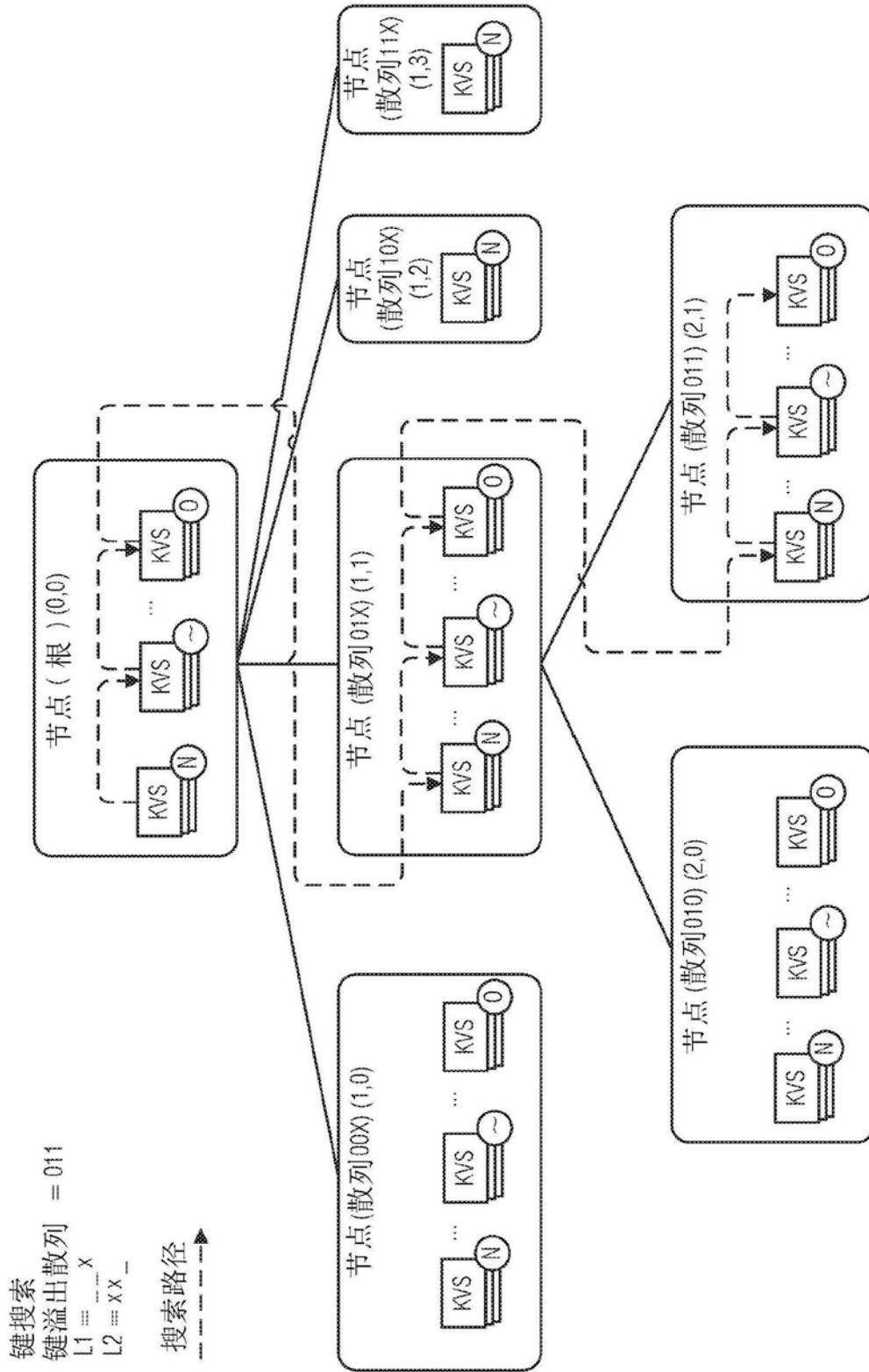


图21

2200

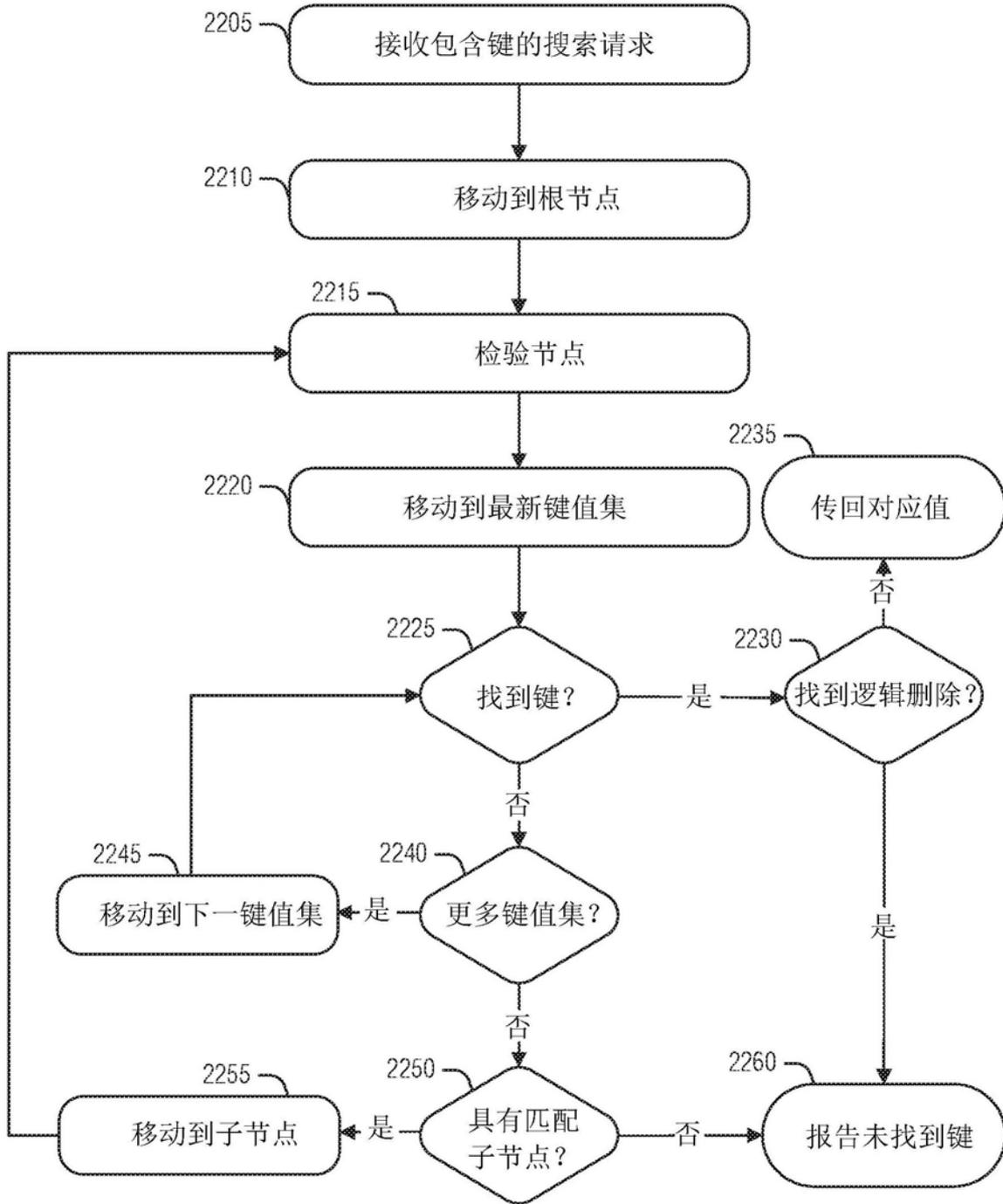


图22

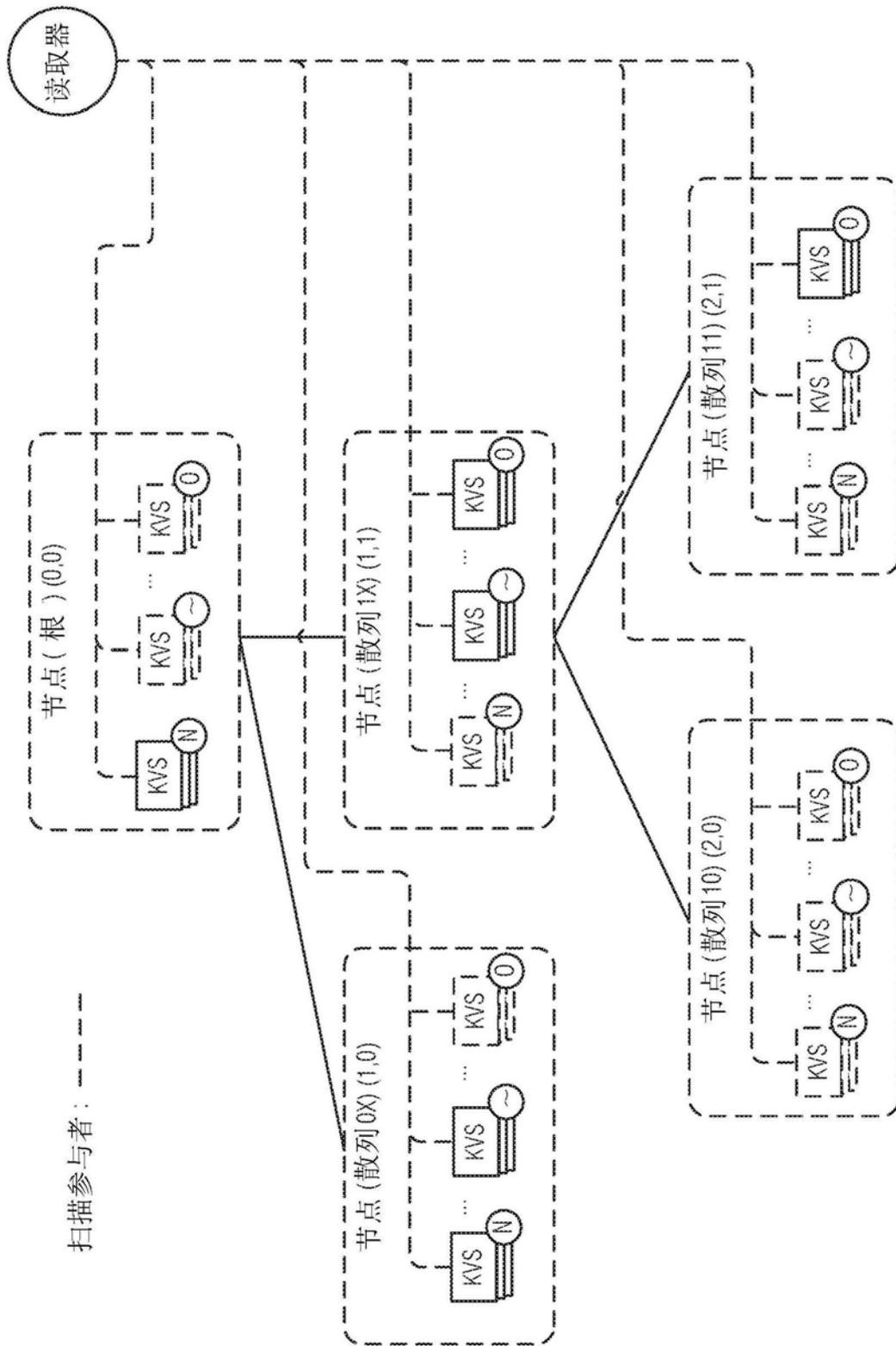


图23

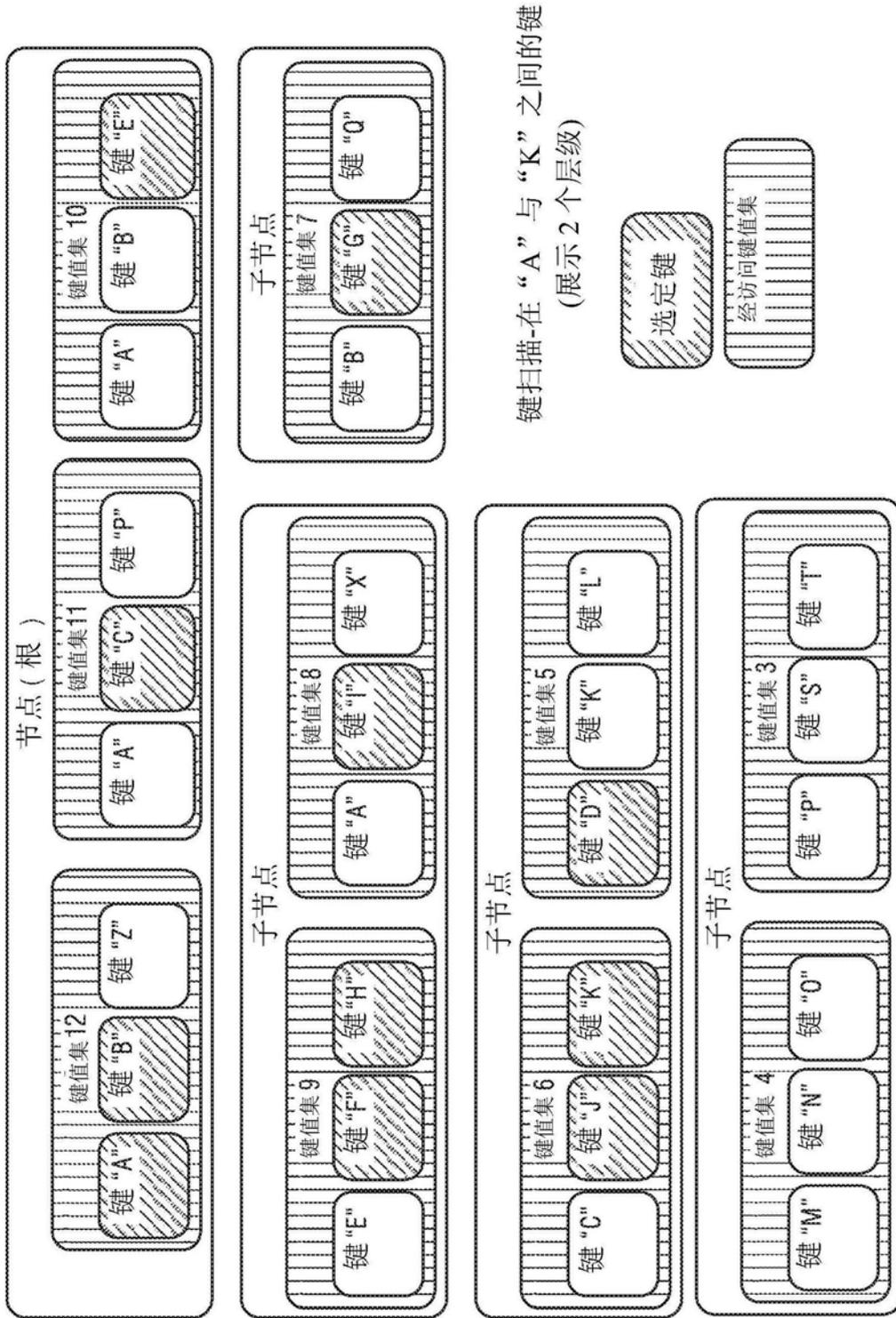


图24

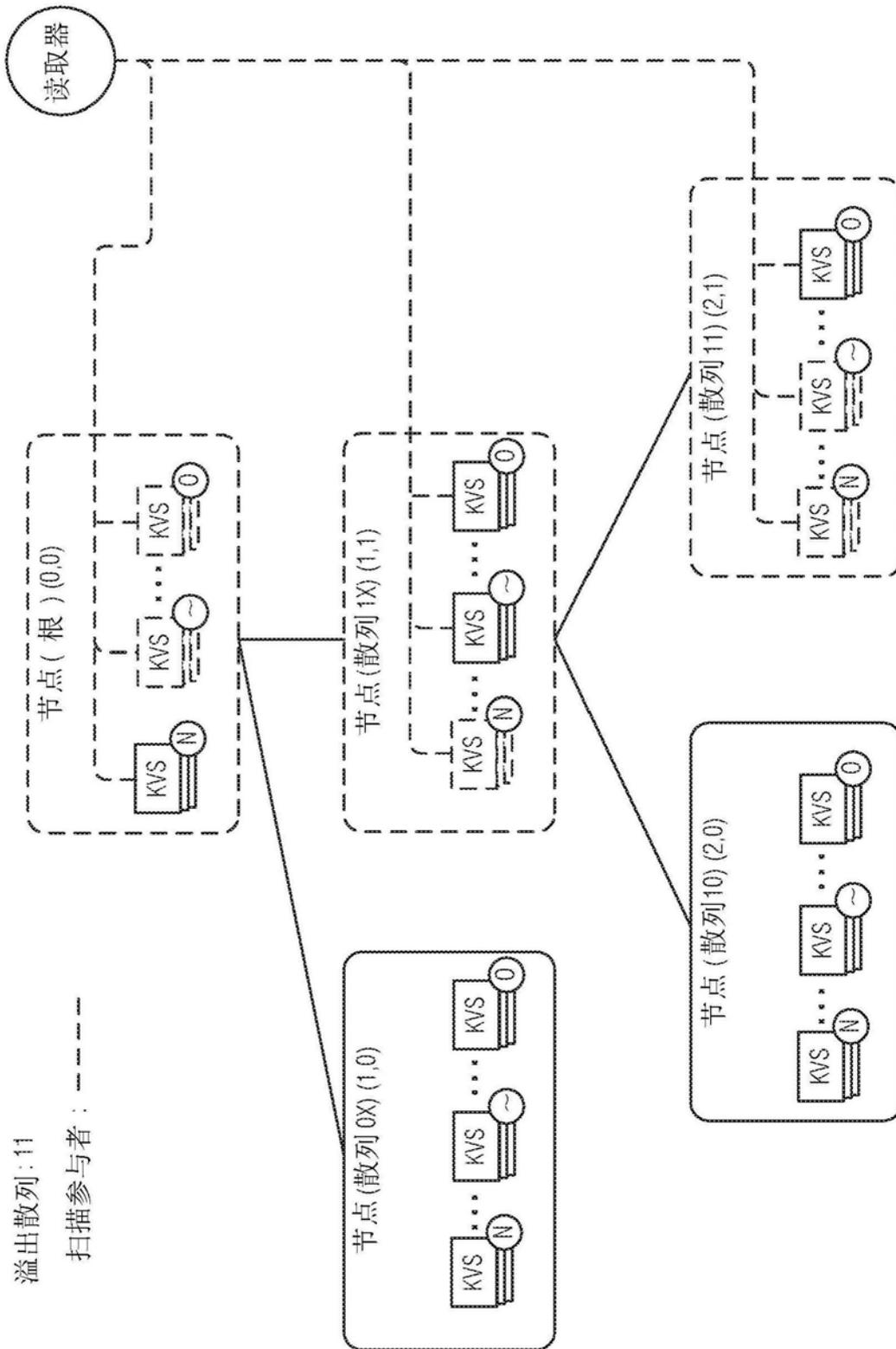


图25

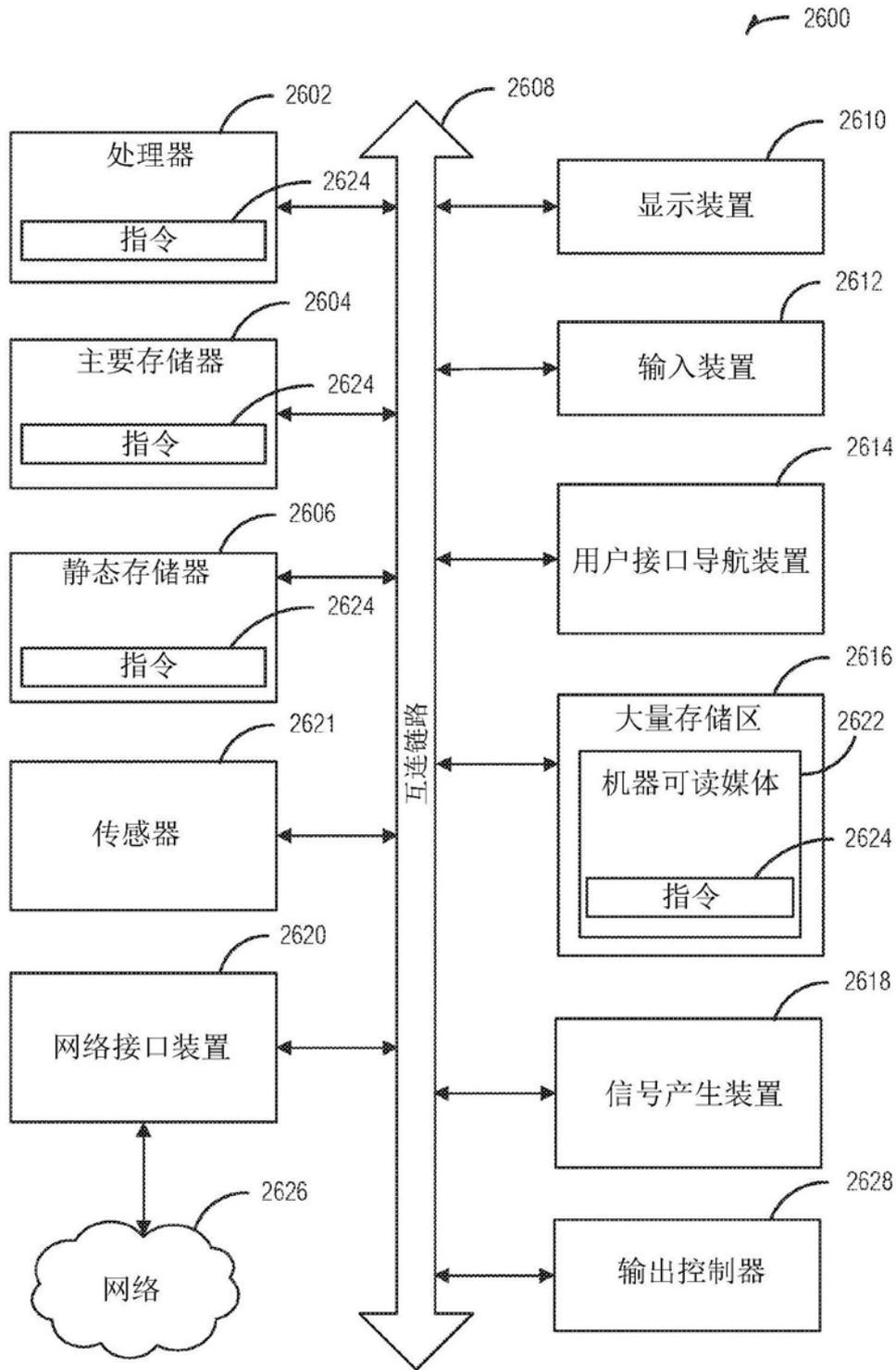


图26