(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 108140049 A
(43)申请公布日 2018.06.08

(21)申请号 201680059766.4
(22)申请日 2016.08.16
(30)优先权数据
14/918,069 2015.10.20 US
14/918,130 2015.10.20 US
14/918,168 2015.10.20 US
(85)PCT国际申请进入国家阶段日
2018.04.12
(86)PCT国际申请的申请数据
PCT/IB2016/054899 2016.08.16
(87)PCT国际申请的公布数据
WO2017/068438 EN 2017.04.27
(71)申请人 国际商业机器公司
地址 美国纽约阿芒克
(72)发明人 L·阿罗诺维奇 K·K·黄 G·T·基施
(74)专利代理机构 北京市金杜律师事务所 11256
代理人 酆迅 李峥宇
(51)Int.Cl.
G06F 17/30(2006.01)

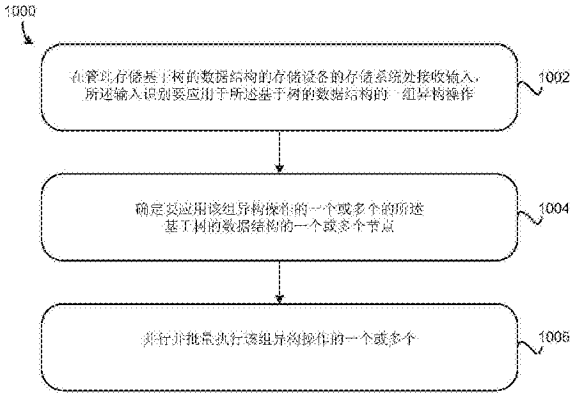
权利要求书4页 说明书25页 附图11页

(54)发明名称

基于树的数据结构的并行批量处理

(57)摘要

用于在基于树的数据结构上并行处理操作的计算机实现的方法包括:在存储系统处接收输入,所述存储系统管理存储基于树的数据结构的存储设备,所述输入标识要应用于所述基于树的数据结构一组异构操作;确定所述一组异构操作的一个或多个将被应用的所述基于树的数据结构的一个或多个节点;以及并行并批量执行一组异构操作的一个或多个。还公开了用于完成这些的系统和方法。



CN 108140049 A

1.一种用于在基于树的数据结构上并行处理操作的计算机实现的方法,包括:

在管理存储所述基于树的数据结构的存储设备的存储系统处接收输入,所述输入识别要应用于所述基于树的数据结构的一组异构操作;

确定要应用所述一组异构操作的一个或多个的所述基于树的数据结构的一个或多个节点;以及

并行并批量执行所述一组异构操作的一个或多个。

2.根据权利要求1所述的方法,其中所述一组异构操作包括以下中的一个或多个:插入操作、删除操作和更新操作。

3.根据权利要求2所述的方法,所述输入包括:

对基于树的数据结构的引用,以及

对的列表,其中每一对由数据条目和相关联的至少一个操作组成。

4.根据权利要求1所述的方法,其中所述确定不涉及访问所述基于树的数据结构的叶节点;

其中叶节点存储一个或多个数据条目;和

其中在输入中指定至少一些数据条目。

5.根据权利要求1所述的方法,包括搜索阶段,所述搜索阶段包括:

以降序的方式评估所述基于树的数据结构的每一层中的节点;和

其中所述评估包括:针对每个层,并行确定所述被评估层中的任何节点是否是以下中的一个或多个:

用于在树中进一步路由搜索的节点;和

应该应用所述一组异构操作的一个或多个的节点。

6.根据权利要求5所述的方法,其中所述搜索阶段生成输出,所述输出包括:节点和操作集,所述节点和操作集包括存储标识符的列表,

其中每个存储标识符标识应当应用所述一组异类操作的一个或多个的节点之一;和

其中至少一些所述存储标识符与由所述搜索阶段生成的至少一个操作相关联。

7.根据权利要求6所述的方法,包括更新阶段,所述更新阶段包括:

将述一组异构操作的一个或多个应用于具有在所述节点和操作集中列出的存储标识符的每个节点,

其中,对于具有在所述节点和操作集中列出的存储标识符的每个节点,应用于该节点的该组异构操作中的一个或多个异构操作基于与该节点的存储标识符相关联的至少一个操作;和

其中所述一组异构操作中的一个或多个被并行、独立地并且批量地应用于具有在所述节点和操作集中列出的存储标识符的每个节点。

8.根据权利要求7所述的方法,其中将所述一组异构操作的一个或多个应用于具有在所述节点和操作集中列出的存储标识符的每个节点包括以下中的一个或多个:

插入一个或多个新节点;

删除一个或多个现有节点;

合并一个或多个现有节点;

分割一个或多个现有节点;和

更新一个或多个现有节点的一个或多个数据条目。

9.如权利要求8所述的方法，其中，在处理的节点被分割时创建的新节点在下一个更新阶段中生成要被添加到基于树的数据结构的下一个上层的条目，

其中从处理的节点到兄弟节点的条目的完全传送在下一个更新阶段中生成树的下一个上层中的条目的删除操作，并且

其中，其条目内容被修改并因此其代表性条目已经改变的经处理的节点在下一更新阶段中生成要删除的条目并插入到树的下一个上层。

10.如权利要求7所述的方法，其中所述更新阶段是迭代过程；和

其中由每个更新阶段产生的操作是在当前更新阶段迭代中由创建节点、删除节点和修改节点而产生的。

11.如权利要求7所述的方法，所述更新阶段包括至少部分地基于将所述一组异构操作的一个或多个应用于具有在所述节点和操作集中列出的存储标识符的每个节点来生成新的节点和操作集；

其中所述新的节点和操作集包括存储标识符的列表，

其中所述新的节点和操作集中的每个存储标识符标识应该应用新的一组异构操作的一个或多个的多个节点的节点之一。

12.如权利要求11所述的方法，所述更新阶段包括：

将所述一组异构操作的一个或多个应用于具有在所述新的节点和操作集中列出的存储标识符的每个节点，

其中所述一组异构操作的所述一个或多个并行并且批量应用于具有在所述新的节点和操作合中列出的存储标识符的每个节点。

13.如权利要求7所述的方法，所述更新阶段包括：

确定响应于将所述一组异构操作的一个或多个应用于具有在所述节点和操作集中列出的存储标识符的每个节点而生成的新的节点和操作集是否包括与基于树的数据结构的下一上层中的节点对应的任何存储标识符；和

响应于确定新的节点和操作集不包括与基于树的数据结构的下一上层中的节点对应的任何存储标识符，终止更新阶段。

14.如权利要求7所述的方法，所述基于树的数据结构包括多个节点层，并且所述方法包括以迭代方式执行所述多个节点层中的一个或多个节点层的更新阶段；

其中所述更新阶段的每次迭代产生要在所述更新阶段的下一次迭代中更新的新的节点和操作集合；和

其中在所述更新阶段的下一次迭代中要更新的节点层位于在所述更新阶段的当前迭代中正在执行所述更新阶段的所述节点层之上。

15.如权利要求7所述的方法，其中在所述搜索阶段中使用共享许可来访问节点，

使用独占许可在所述更新阶段期间访问要应用所述一组异构操作的一个或多个的节点。

16.如权利要求7所述的方法，其中将所述一组异构操作的一个或多个并行批量地应用于具有所述节点和操作集中列出的存储标识符的每个节点包括使用不同处理线程向每个节点应用操作，

其中每个进程线程被并行处理。

17.一种用于在基于树的数据结构上并行处理异构操作的计算机实现的方法，所述基于树的数据结构包括：

由根节点组成的根节点层；

从根节点层下降并且包括至少两个内部节点的第一节点层；

从第一节点层下降并包括至少四个内部节点的第二节点层；和

从第二节点层下降并包括多个叶节点叶节点层；和

该方法包括：

在存储所述基于树的数据结构的存储系统处接收输入，所述输入识别要应用于所述基于树的数据结构的一个或多个叶节点的一组异构操作；和

基于所述输入并行并且批量地将所述一组异构操作一个或多个执行到一个或多个所述叶节点。

18.根据权利要求17所述的方法，包括搜索阶段，所述搜索阶段包括：

以降序从根节点层到叶节点层评估基于树的数据结构的每一层；和

并行地确定应该应用所述一组异构操作的叶节点。

19.根据权利要求18所述的方法，其中所述搜索阶段产生输出，所述输出包括：节点和操作集，所述节点和操作集包括存储标识符的列表，

其中每个存储标识符标识应该应用所述一组异构操作的一个或多个的所述叶节点中之一；和

其中每个存储标识符与在所接收的输入中标识的至少一个数据条目/操作对相关联。

20.根据权利要求19所述的方法，包括更新阶段，所述更新阶段包括：

将所述一组异构操作的一个或多个应用于具有在所述节点和操作集中列出的存储标识符的每个叶节点，

其中，对于具有在所述节点和操作集中列出的存储标识符的每个叶节点，应用于叶节点的所述一组异构操作的一个或多个是基于与叶节点的存储标识符相关的至少一个数据条目/操作对；和

其中所述一组异构操作并行并且批量地应用于具有在所述节点和操作集中列出的存储标识符的每个所述叶节点。

21.根据权利要求20所述的方法，所述更新阶段包括：

至少部分地基于将所述一组异构操作的一个或多个应用于具有在节点和操作集中列出的存储标识符的叶节点来生成新的节点和操作集，其中新的节点和操作集包括存储标识符的列表，并且其中每个存储标识符标识应该应用所述一组异构操作的一个或多个的所述第二节点层中的所述内部节点之一；

将所述一组异构操作的一个或多个应用于所述第二节点层中的具有在所述新的节点和操作集中列出的存储标识符的内部节点；和

其中所述一组异构操作的一个或多个并行并且批量地应用于所述第二节点层中的具有在所述新的节点和操作集中列出的存储标识符的每个内部节点。

22.根据权利要求21所述的方法，所述更新阶段包括：

至少部分地基于将所述一组异构操作的一个或多个应用于具有在所述新的节点和操

作集中列出的存储标识符的内部节点来生成第二新的节点和操作集，其中所述第二新的节点和操作集包括存储标识符的列表，并且其中每个存储标识符标识应当应用所述一组异构操作的一个或多个的所述第一节点层中的内部节点之一；

将所述一组异构操作的一个或多个应用于所述第一节点层中的具有在所述第二新的节点和操作集中列出的存储标识符的内部节点；和

其中所述一组异构操作的一个或多个并行并且批量应用于所述第一节点层中的具有在所述第二新的节点和操作集中列出的存储标识符的每个内部节点。

23.一种用于在基于树的数据结构上的处理操作的计算机实现的方法，包括：

在管理存储所述基于树的数据结构的存储设备的存储系统处接收输入，所述输入识别要应用于所述基于树的数据结构的一组异构操作；

确定要应用所述一组异构操作的一个或多个的所述基于树的数据结构的一个或多个节点；

确定所述一组异构操作的一个或多个组，所述确定至少部分地基于将要应用所述异构操作的所述一个或多个节点。

24.一种用于在基于树的数据结构上并行处理操作的计算机实现的方法，所述方法包括：

在管理存储所述基于树的数据结构的存储设备的存储系统处接收输入，所述输入识别要应用于所述基于树的数据结构的一组异构操作；

确定要应用所述一组异构操作的一个或多个的所述基于树的数据结构的一个或多个节点；

根据要应用所述一组异构操作的所述一个或多个节点来确定所述一组异构操作中的一个或多个组；和

针对所述一个或多个组中的每一个组，根据预定义顺序应用所述一组异构操作。

25.一种包括存储在计算机可读介质上的计算机程序代码的计算机程序，当所述计算机程序代码加载到计算机系统中并在其上执行时，使所述计算机系统执行根据权利要求1至24中任一项所述的方法的所有步骤。

26.一种用于在基于树的数据结构上并行处理操作的存储系统，所述存储系统包括存储系统管理器以及与所述存储系统管理器集成和/或可由所述存储系统管理器执行的逻辑，所述逻辑被配置为使所述存储系统：

接收识别要应用于所述基于树的数据结构的一组异构操作的输入；

确定要应用所述一组异构操作的一个或多个的所述基于树的数据结构的一个或多个节点；和

并行并批量执行所述一组异构操作的一个或多个。

基于树的数据结构的并行批量处理

背景技术

[0001] 本发明涉及数据结构的处理，更具体地说，本发明涉及在以基于树的数据结构(例如分页搜索树数据结构)组织的数据的高效并行批量处理，以及对数据结构本身的处理。

[0002] 存储在常规存储系统上的数据根据众多已知数据结构中的一个来组织。最通常地，数据根据基于树的数据结构来组织，例如分页搜索树，其构成存储数据和/或路由信息以便于搜索感兴趣的数据的节点的分叉网络。在分页的搜索树中，每个节点通常对应于一个磁盘页面。

[0003] 因此，提供用于提高基于树的数据结构修改的效率的系统和技术将是有益的，以便通过增加输入/输出(I/O)来提高广泛范围的数据存储系统的功能和吞吐量效率，并降低存储、组织、搜索和更新数据条目和相应数据结构的计算成本。

发明内容

[0004] 在一个实施例中，一种用于在基于树的数据结构上并行处理操作的计算机实现的方法包括：在存储系统处接收输入，所述存储系统管理存储所述基于树的数据结构的存储设备，所述输入标识将要应用于基于树的数据结构的一组异构操作；确定所述一组异构操作的一个或多个将被应用到的所述基于树的数据结构的一个或多个节点；以及并行并批量执行所述一组异构操作的一个或多个。

[0005] 在另一个实施例中，一种用于在基于树的数据结构上并行处理操作的计算机程序产品包括具有程序指令的计算机可读存储介质。计算机可读存储介质本身不是暂时信号，并且程序指令可由存储系统管理器执行以使存储系统管理器执行方法。该方法包括由存储系统管理器接收识别要应用于基于树的数据结构的一组异构操作的输入；由所述存储系统管理器确定所述一组异构操作的一个或多个将被应用到的所述基于树的数据结构的一个或多个节点；以及由所述存储系统管理器并行并批量执行所述一组异构操作的一个或多个。

[0006] 在又一个实施例中，一种用于基于树的数据结构上并行处理操作的存储系统包括存储系统管理器，以及与存储系统管理器集成和/或可由其执行的逻辑。所述逻辑被配置为使所述存储系统：接收识别要应用于基于树的数据结构的一组异构操作的输入；确定所述一组异构操作中的一个或多个将被应用到的所述基于树的数据结构的一个或多个节点；并行并批量执行一组或多组异构操作。

[0007] 从以下结合附图的详细描述中，本发明的其它方面和实施例将变得显而易见，所述详细描述结合附图以示例的方式说明本发明的原理。

附图说明

[0008] 现在将参照附图仅以举例的方式描述本发明的实施例，其中：

[0009] 图1示出了根据一个实施例的网络体系结构。

[0010] 图2示出了根据一个实施例可以与图1的服务器和/或客户端相关联的代表性硬件

环境。

[0011] 图3示出了根据一个实施例的分层数据存储系统。

[0012] 图4是根据一个实施例的平衡三层分页搜索树的简化示意图。

[0013] 图5是根据一个实施例的不平衡三层分页搜索树的简化示意图。

[0014] 图6是根据一个实施例的不平衡多层分页搜索树的简化示意图。

[0015] 图7是根据优选实施例的在搜索阶段和更新阶段期间的基于n层树的数据结构以及关于基于树的数据结构的处理的进度的简化示意图。

[0016] 图8描绘了根据当前公开的发明构思的优选实施例的表示搜索阶段期间的处理的流程图。

[0017] 图9描绘了根据当前公开的发明构思的优选实施例的表示在搜索阶段期间的处理的流程图。

[0018] 图10是根据一个实施例的用于在分页搜索树数据结构上对异构操作进行并行批量处理的计算机实现的方法的流程图。

[0019] 图11是根据一个实施例的用于隔离分页搜索树数据结构上的并行操作的计算机实现的方法的流程图。

[0020] 图12是根据一个实施例的用于在分页搜索树数据结构上对操作进行有效排序的方法的流程图。

具体实施方式

[0021] 以下描述是为了说明本发明的一般原理的目的而做出的，并不意味着限制在此要求保护的发明构思。此外，本文描述的特定特征可以与各种可能的组合和排列中的每一个中的其他描述的特征组合使用。

[0022] 除非在此另外具体定义，否则所有术语将被给予它们最广泛的可能解释，包括从说明书暗示的含义以及本领域技术人员理解和/或如在词典、论文等中定义的含义。

[0023] 还必须注意的是，除非另外指明，否则如说明书和所附权利要求中所使用的，单数形式“一”，“一个”和“该”包括复数指示物。将进一步理解的是，当在本说明书中使用时，术语“包括”指定所陈述的特征、整体、步骤、操作、元件和/或组件的存在，但不排除存在或添加一个或多个其他特征、整体、步骤、操作、元件、组件和/或其组合。

[0024] 以下描述公开了用于使用异构操作类型的批量并行处理来有效地操纵基于树的数据结构的系统、方法和计算机程序产品的若干优选实施例。

[0025] 在一个一般的实施例中，一种用于在基于树的数据结构上并行处理操作的计算机实现的方法包括：在存储系统处接收输入，所述存储系统管理存储所述基于树的数据结构的存储设备，所述输入标识将要应用于基于树的数据结构的一组异构操作；确定该组异构操作中的一个或多个将被应用于所述基于树的数据结构的一个或多个节点；以及并行并批量执行该组异构操作的一个或多个。

[0026] 在另一个一般实施例中，一种用于在基于树的数据结构上并行处理操作的计算机程序产品包括具有程序指令的计算机可读存储介质。计算机可读存储介质本身不是暂时信号，并且程序指令可由存储系统管理器执行以使存储系统管理器执行方法。该方法包括由存储系统管理器接收识别要应用于基于树的数据结构的一组异构操作的输入；由所述存储

系统管理器确定所述一组异构操作的一个或多个将被应用到的所述基于树的数据结构的一个或多个节点;以及由所述存储系统管理器并行并批量执行所述一组异构操作的一个或多个。

[0027] 在又一个一般实施例中,一种用于基于树的数据结构上并行处理操作的存储系统包括存储系统管理器,以及与存储系统管理器集成和/或可由其执行的逻辑。所述逻辑被配置为使所述存储系统:接收识别要应用于基于树的数据结构的一组异构操作的输入;确定所述一组异构操作中的一个或多个将被应用到的所述基于树的数据结构的一个或多个节点;并行并批量执行一组或多组异构操作。

[0028] 定义

[0029] 相邻节点

[0030] 如本文中所理解的,基于树的数据结构内的节点与树中同一层的直接相邻节点相邻,其中同一树层的所有节点具有与树的根节点相同的距离,并且相邻节点可以从或可以不从下一个上层树层中的相同父节点下降。特定层的相邻节点也被称为兄弟节点。

[0031] 并行批量处理

[0032] 如这里所理解的,并行批量处理涉及在基于树的数据结构上同时处理多个操作。在同一时间窗口内进行处理时,操作被同时处理。在同一时间窗口内同时处理多个操作(而不是单独处理)时,操作被批量处理。

[0033] 在各种实施例中,并行批量处理包括属于树的同一层的基于树的数据结构的多个节点的同时独立处理。在特别优选的实施例中,并行批量处理涉及使用独立操作线程并行处理基于树的数据结构的特定层的所有节点。

[0034] 异构操作

[0035] 如本文中所理解的,异构操作包括可以执行以修改或处理存储在基于树的数据结构中的数据和/或修改或处理基于树的数据结构本身的组织的任何两种或更多种不同类型的操作。如本领域普通技术人员在阅读本说明书后将理解的,示例性操作包括更新、插入或移除存储在基于树的数据结构的节点中的数据条目和/或路由条目。

[0036] 因此,一组异构操作包括本文描述的示例性操作中的任何两种或更多种。在优选实施例中,异构操作涉及数据条目而不是树节点。在特别优选的实施例中,异构操作涉及叶节点而不是内部节点。

[0037] 节点删除

[0038] 如本文中所理解的,节点删除涉及从基于树的数据结构中消除节点。

[0039] 在各种实施例中,节点删除可以在节点合并之后执行,例如,移除由于其内容合并到数据结构中的另一个节点而清空的节点。节点删除可以包括例如修改和/或移除属于被删除节点从其下降的一个或多个层的节点中的路由条目。

[0040] 节点插入

[0041] 如本文所理解的,节点插入涉及在基于树的数据结构内创建先前不存在的节点。

[0042] 在各种实施例中,可以响应于确定特定层中的一个或多个节点被过度填充而执行节点插入。通过将数据条目从过度填充的节点移动到新插入的节点。

[0043] 节点合并

[0044] 如本文中所理解的,节点合并涉及组合两个或更多个现有节点以形成单个节点。

[0045] 在各种实施例中，可以响应于确定存在或将存在未满的节点(例如，作为在数据结构的特定层上执行异构操作的结果)来执行合并。节点合并优选地通过将未满的节点与兄弟节点组合来完成。

[0046] 节点权限

[0047] 用户可以获得访问基于树的数据结构的特定节点的权限，相对于在相同数据结构上操作的所有其他用户原子地访问基于树的数据结构的特定节点。

[0048] 该权限可以被共享，在这种情况下，该节点可以只被读取。只读、共享权限可以由多个用户同时获得。

[0049] 可选地，该权限可以是独占性的，在这种情况下该节点也可以被修改。独占权限与访问节点的其他用户互斥。

[0050] 节点重新平衡

[0051] 如本文中所理解的，节点重新平衡包括将数据条目从充满节点传送到其兄弟节点，或从其兄弟节点传送到未满的节点，以将节点维持在所需的存储利用范围内。

[0052] 在各种实施例中，节点重新平衡对于在基于树的数据结构的节点之间维持适当的存储利用水平是特别有用的。

[0053] 因此，本领域技术人员将基于这些描述认识到，在优选实施例中，节点重新平衡可以包括或位于修改基于树的数据结构中的路由信息以适应对节点的组织和/或对数据条目的位置的任何改变的过程之后。

[0054] 节点分割

[0055] 如本文中所理解的，节点分割操作涉及将节点分割成两个或更多个节点，其中至少一个节点在执行分割操作之前不是数据结构的一部分。

[0056] 在各种实施例中，如本领域普通技术人员在阅读本说明书后将理解的那样，作为在基于树的数据结构的特定层上执行一个或多个异构操作的结果，可以响应于确定基于树的数据结构的现有节点正在或将要变得过度填充而发生节点分割。

[0057] 节点更新

[0058] 如本文中所理解的，更新节点包括修改数据输入有效载荷(例如，针对B+树的叶节点)和修改存储在节点中的一个或多个路由条目(例如，针对B+树)。

[0059] 在各种实施例中，可以响应于引起数据结构组织中的改变的另一操作来执行节点更新，这可能需要更新路由条目以适应改变。

[0060] 另外地和/或可选地，可以更新节点以完成对存储在节点中的数据条目的修改。

[0061] 存储标识符

[0062] 如本文中所理解的，存储标识符是标识存储设备中的节点的位置的数据元素。

[0063] 在各种实施例中，基于树的数据结构中的节点优选地与唯一的存储标识符相关联。例如，在基于树的数据结构中从节点N1指向节点N2，节点N2的存储标识符优选地存储在节点N1中。

[0064] 基于树的数据结构

[0065] 如本文中所理解的，基于树的数据结构包括其中根据分支分级结构来存储和组织数据条目的各种数据结构。优选地，如本领域普通技术人员在阅读本说明书后将理解的那样，结构以分叉方式分支，但是具有更高阶分裂的结构被认为在本公开的范围内，例如，四

分叉、八分叉等。

[0066] 分页搜索树数据结构

[0067] 如本文中所理解的，分页搜索树数据结构被设计用于数据的组织和搜索。在分页搜索树数据结构中，数据存储在节点中，其中节点通常对应于磁盘页面，并且节点按照分层树结构被组织和链接。这些数据结构旨在在大型数据集内提供快速高效的搜索。典型地，这样的数据结构存储在磁盘上，而更快的存储器(例如存储器或固态硬盘)上的高速缓存被用来存储数据结构的部分以提高性能。

[0068] 例如，B+树是各种分页搜索树数据结构。在B+树中，数据条目专门存储在叶节点中，而内部节点存储路由信息以将操作指向适当的叶节点。存储在叶节点中的数据条目包括密钥(以便于搜索)以及包括感兴趣的数据的有效载荷。路由条目包括用于将操作引导到适当的叶节点的密钥和内部链路或节点标识符。

[0069] 计算机、网络和存储系统架构

[0070] 在任何可能的技术细节结合层面，本发明可以是系统、方法和/或计算机程序产品。计算机程序产品可以包括计算机可读存储介质，其上载有用于使处理器实现本发明的各个方面的计算机可读程序指令。

[0071] 计算机可读存储介质可以是可以保持和存储由指令执行设备使用的指令的有形设备。计算机可读存储介质例如可以是——但不限于——电存储设备、磁存储设备、光存储设备、电磁存储设备、半导体存储设备或者上述的任意合适的组合。计算机可读存储介质的更具体的例子(非穷举的列表)包括：便携式计算机盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦式可编程只读存储器(EPROM或闪存)、静态随机存取存储器(SRAM)、便携式压缩盘只读存储器(CD-ROM)、数字多功能盘(DVD)、记忆棒、软盘、机械编码设备、例如其上存储有指令的打孔卡或凹槽内凸起结构、以及上述的任意合适的组合。这里所使用的计算机可读存储介质不被解释为瞬时信号本身，诸如无线电波或者其他自由传播的电磁波、通过波导或其他传输媒介传播的电磁波(例如，通过光纤电缆的光脉冲)、或者通过电线传输的电信号。

[0072] 这里所描述的计算机可读程序指令可以从计算机可读存储介质下载到各个计算/处理设备，或者通过网络、例如因特网、局域网、广域网和/或无线网下载到外部计算机或外部存储设备。网络可以包括铜传输电缆、光纤传输、无线传输、路由器、防火墙、交换机、网关计算机和/或边缘服务器。每个计算/处理设备中的网络适配卡或者网络接口从网络接收计算机可读程序指令，并转发该计算机可读程序指令，以供存储在各个计算/处理设备中的计算机可读存储介质中。

[0073] 用于执行本发明操作的计算机程序指令可以是汇编指令、指令集架构(ISA)指令、机器指令、机器相关指令、微代码、固件指令、状态设置数据、集成电路配置数据或者以一种或多种编程语言的任意组合编写的源代码或目标代码，所述编程语言包括面向对象的编程语言—诸如Smalltalk、C++等，以及过程式编程语言—诸如“C”语言或类似的编程语言。计算机可读程序指令可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中，远程计算机可以通过任意种类的网络—包括局域网(LAN)或广域网(WAN)—连接到用户计算机，或者，可以连接到外部计算机

(例如利用因特网服务提供商来通过因特网连接)。在一些实施例中,通过利用计算机可读程序指令的状态信息来个性化定制电子电路,例如可编程逻辑电路、现场可编程门阵列(FPGA)或可编程逻辑阵列(PLA),该电子电路可以执行计算机可读程序指令,从而实现本发明的各个方面。

[0074] 这里参照根据本发明实施例的方法、装置(系统)和计算机程序产品的流程图和/或框图描述了本发明的各个方面。应当理解,流程图和/或框图的每个方框以及流程图和/或框图中各方框的组合,都可以由计算机可读程序指令实现。

[0075] 这些计算机可读程序指令可以提供给通用计算机、专用计算机或其它可编程数据处理装置的处理器,从而生产出一种机器,使得这些指令在通过计算机或其它可编程数据处理装置的处理器执行时,产生了实现流程图和/或框图中的一个或多个方框中规定的功能/动作的装置。也可以把这些计算机可读程序指令存储在计算机可读存储介质中,这些指令使得计算机、可编程数据处理装置和/或其他设备以特定方式工作,从而,存储有指令的计算机可读介质则包括一个制造品,其包括实现流程图和/或框图中的一个或多个方框中规定的功能/动作的各个方面的指令。

[0076] 也可以把计算机可读程序指令加载到计算机、其它可编程数据处理装置、或其它设备上,使得在计算机、其它可编程数据处理装置或其它设备上执行一系列操作步骤,以产生计算机实现的过程,从而使得在计算机、其它可编程数据处理装置、或其它设备上执行的指令实现流程图和/或框图中的一个或多个方框中规定的功能/动作。

[0077] 附图中的流程图和框图显示了根据本发明的多个实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上,流程图或框图中的每个方框可以代表一个模块、程序段或指令的一部分,所述模块、程序段或指令的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。在有些作为替换的实现中,方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如,两个连续的方框实际上可以基本并行地执行,它们有时也可以按相反的顺序执行,这依所涉及的功能而定。也要注意的是,框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合,可以用执行规定的功能或动作的专用的基于硬件的系统来实现,或者可以用专用硬件与计算机指令的组合来实现。

[0078] 图1示出了根据一个实施例的架构100。在图1中,提供了包括第一远程网络104和第二远程网络106的多个远程网络102。网关101可以连接在远程网络102和邻近网络108之间。在本架构100的上下文中,网络104、106可以各自采取任何形式,包括但不限于LAN、诸如因特网、公共交换电话网(PSTN)、内部电话网等的WAN。

[0079] 在使用中,网关101充当从远程网络102到邻近网络108的入口点。这样,网关101可以用作路由器,其能够引导到达网关101和交换机的给定数据分组,为给定分组提供了进出网关101的实际路径。

[0080] 还包括至少一个耦合到邻近网络108的数据服务器114,并且该数据服务器114可以经由网关101从远程网络102访问。应该注意的是,数据服务器114可以包括任何类型的计算设备/群件。多个用户设备116耦合到每个数据服务器114。用户设备116也可以通过网络104、106、108中的一个直接连接。这样的用户设备116可以包括台式计算机、膝上型计算机、手持计算机、打印机或任何其他类型的逻辑。应该注意的是,在一个实施例中,用户设备111也可以直接耦合到任何网络。

[0081] 外围设备120或一系列外围设备120(例如传真机、打印机、联网和/或本地的存储单元或系统等)可以耦合到网络104、106、108中的一个或多个。应当指出,数据库和/或附加组件可以与耦合到网络104、106、108的任何类型的网络元件一起使用或者集成到其中。在本说明书的上下文中,网络元件可以指任何一个网络。

[0082] 根据一些方法,本文描述的方法和系统可以用虚拟系统和/或系统来实现,所述虚拟系统和/或系统模拟一个或多个其他系统,诸如模拟IBM z/OS环境的UNIX系统,虚拟托管MICROSOFT WINDOWS环境的UNIX系统,模拟IBM z/OS环境的MICROSOFT WINDOWS系统等。在一些实施例中,可以通过使用VMWARE软件来增强该虚拟化和/或仿真。

[0083] 在更多方法中,一个或多个网络104、106、108可以表示通常被称为“云”的系统集群。在云计算中,以按需关系向云中的任何系统提供诸如处理能力、外围设备、软件、数据、服务器等的共享资源,从而允许跨多个计算系统访问和分配服务。云计算通常涉及在云中运行的系统之间的互联网连接,但是也可以使用连接系统的其他技术。

[0084] 图2示出了与图1的用户设备116和/或服务器114相关联的代表性硬件环境。如图1所示,根据一个实施例。这样的图示出了具有诸如微处理器的中央处理单元210和经由系统总线212互连的多个其他单元的工作站的典型硬件配置。

[0085] 图2中所示的工作站包括随机存取存储器(RAM)214、只读存储器(ROM)216、用于将诸如磁盘存储单元220的外围设备连接到总线212的I/O适配器218、用于将键盘224、鼠标226、扬声器228、麦克风232和/或诸如触摸屏和数字照相机(未示出)之类的其他用户接口设备连接到总线212的用户接口适配器222、用于将工作站连接到通信网络235(例如数据处理网络)的通信适配器234和用于将总线212连接到显示设备238的显示适配器236。

[0086] 工作站可以在其上具有诸如Microsoft Windows®、MAC OS、UNIX OS等的操作系统(OS)。将会理解,优选实施例也可以在平台上实现并且操作除了那些提到的系统。可以使用XML、C和/或C++语言或其他编程语言以及面向对象的编程方法来编写优选实施例。面向对象编程(OOP)越来越多地用于开发复杂的应用程序。

[0087] 现在参考图3。在图3中,示出了根据一个实施例的存储系统300。请注意,根据各种实施例,图3中所示的一些元件可以被实现为硬件和/或软件。存储系统300可以包括存储系统管理器312,用于与至少一个较高存储层302和至少一个上层存储层306上的多个介质进行通信。一个或多个较高存储层302优选地可以包括一个或多个随机访问和/或直接访问介质304,诸如硬盘驱动器(HDD)中的硬盘、非易失性存储器(NVM)、固态驱动器(SSD)中的固态存储器、闪速存储器、SSD阵列、闪存阵列、和/或本文中提到的或本领域已知的其它访问介质。上层存储层306可以优选地包括一个或多个上层执行存储介质308,包括诸如磁带驱动器和/或光学介质中的磁带的顺序存取介质、超级访问HDD、超级访问SSD等和/或本文所述或本领域已知的其它存储介质。一个或多个额外的存储层316可以包括系统300的设计者期望的存储存储介质的任何组合。而且,任何较高存储层302和/或上层存储层306可以包括存储设备和/或存储介质。

[0088] 存储系统管理器312可以通过诸如图3所示的存储区域网络(SAN)或其他一些合适的网络类型的网络310与较高存储层302和上层存储层306上的存储介质304、308进行通信,如图3所示。存储系统管理器312还可以通过主机接口314与一个或多个主机系统(未示出)通信,主机接口314可以与是或不是存储系统管理器312的一部分。存储系统管理器312和/

或存储系统300的任何其它组件可以用硬件和/或软件来实现，并且可以利用处理器(未示出)，例如中央处理单元(CPU)、现场可编程门阵列(FPGA)、专用集成电路(ASIC)等来执行本领域已知类型的命令。当然，可以使用存储系统的任何布置，这对于本领域技术人员在阅读本说明书后将显而易见。

[0089] 在更多实施例中，存储系统300可以包括任何数量的数据存储层，并且可以在每个存储层内包括相同或不同的存储介质。例如，每个数据存储层可以包括相同类型的存储介质，诸如HDD、SSD、顺序存取介质(磁带驱动器中的磁带，光盘驱动器中的光盘等)、直接存取介质(CD-ROM、DVD-ROM等)或媒体存储类型的任何组合。在一个这样的配置中，较高存储层302可以包括用于将数据存储在更高性能的存储环境中的大多数SSD存储介质，并且包括上层存储层306和附加存储层316的剩余存储层可以包括SSD、HDD、磁带驱动器等的任何组合，用于将数据存储在上层性能的存储环境中。以这种方式，可以将更频繁访问的数据、具有更高优先级的数据、需要更快访问的数据等存储到较高存储层302，而不具有这些属性中的一个的数据可以存储到附加存储层316(包括上层存储层306)。当然，根据本文给出的实施例，本领域的技术人员在阅读本说明书之后可以设计存储介质类型的许多其他组合来实现不同的存储方案。

[0090] 根据一些实施例，存储系统(诸如300)可以包括被配置为接收打开数据集的请求的逻辑，被配置为确定所请求的数据集是否被存储到分层数据存储系统300的上层存储层306的多个关联部分中的逻辑，被配置为将所请求的数据集的每个关联部分移动到分层数据存储系统300的较高存储层302的逻辑，以及配置为在分层数据存储系统300的较高存储层302上从相关部分组装所请求的数据集的逻辑。

[0091] 当然，根据各种实施例，该逻辑可以作为任何设备和/或系统上的方法或作为计算机程序产品来实现。

[0092] 基于树的数据结构的示例

[0093] 一般而言，目前公开的发明实施例涉及基于树的数据结构的处理和更新，其实际应用于分页搜索树，典型地用于促进存储系统中数据的组织。这里提出的讨论涉及其中所有数据条目被存储在数据结构的终端节点(即“叶节点”)中的分页搜索树(例如“B+树”)的示例性情况，而所有内部节点“分支节点”和“根节点”)存储路由信息，所述路由信息被配置为在搜索其中存储的特定数据的过程中促进遍历树结构。

[0094] 然而，应该理解的是，本公开的范围不限于B+树，B-树或任何其他特定种类的基于树的数据结构。相反，当前公开的发明构思可以被应用于本领域普通技术人员在阅读本说明书之后将会理解的任何合适的基于树的数据结构。

[0095] 基于树的数据结构400的一个示例性实施例在图4中被图形化地表示。在这个实施例中，树400是三层平衡的数据结构。树的最上层(根层410a)包括根节点402，树中的所有其他节点从该根节点402下降。紧挨在根节点402之下的树400的第一节点层410b包括从根节点402分叉的两个内部节点404。在树400是B+树的实施例中，这些内部节点404包括路由信息而不包括数据条目。相反，如图4所示，在图4的B+树实施例中，数据条目专门存储在位于紧接在第一节点层之下的第二节点层410c中的叶节点406中。当然，本领域技术人员将认识到，在其他实施例中，树400可以不是B+树而是基于树的数据结构的一些其它合适形式，并且其可以将数据条目存储在内部节点404以及叶节点406。

[0096] 图5中示出了类似的基于树的数据结构500，但是根据所描述的实施例，树500是不平衡的。如图4所示，树500包括具有根节点502、根节点层下面的第一节点层510b和第一节点层下面的第二节点层510c的根层510a。但是如图5所示，树500是不对称的，并且在第一节点层510b中包括一个内部节点504，但是还包括树500的多个节点层510b、510c中的叶节点506。实际上，根据各种实施例，本发明实施例同样适用于对称和不对称树型。

[0097] 继续参考图4和5。如图4和图5所示，并且如本文中所理解的，每个树400、500的第一节点层410b、510b分别包括两个水平相邻的兄弟节点(参见如图4所示的节点404，如图5所示的节点504和506)。由于这些兄弟节点各自直接从相应的根节点402、502下降，所以它们也垂直地与它们各自的根节点相邻。

[0098] 然而，在一些实施例中，兄弟节点不一定需要从相同的祖先节点下降。例如，如图4所示，第二节点层410c包括四个叶节点406，其可以被认为包括三组兄弟节点：两个最左兄弟节点从第一节点层410b中最左侧的内部节点404下降。两个最右边的兄弟节点从第一节点层410b中的最右边的内部节点404下降。中间两个兄弟节点中的每一个从不同的父节点下降，一个来自第一节点层410b中最左侧的内部节点404，另一个来自第一节点层410b中的最右侧的内部节点404。在各种实施例中，特定节点层中的所有这种水平和直接相邻的节点对将被视为兄弟节点。

[0099] 类似地，并且如通过图6所示，目前公开的发明实施例适用于更复杂的树结构(例如，具有诸如基于树的数据结构600的n层的非对称树。在根据如图6所示的结构的一个实施例中，树600具有在根节点层610a中的根节点602以及在根节点层610a紧下方的第一节点层610b中的两个内部节点604。

[0100] 在一些实施例中可以被指定为第二节点层610d的下一个节点层在第一节点层下方并且可以包括内部节点604和叶节点606。在各种实施例中，n层树600可以具有位于第一节点层610b与第二节点层610d之间和/或位于第三节点层610e和终端节点610n之间的具有内部节点604和/或叶节点606的任何数量的介入附加层(未示出)，如图6中。经由将第一节点层610b与第二节点层610d之间以及第三节点层610e与终端节点层610n之间的分支分开的断开的锯齿状线来形成。

[0101] 在优选的方法中，当前公开的算法在具有最佳最小和最大扇出特性的存储系统和/或体系结构中实现，即，由存储系统和/或体系结构实现的基于树的数据结构是平衡的，并优化树的每层节点的数量以优化树中的层数。另外，在优选的方法中，算法被应用于具有这种扇出特性的B+树。

[0102] 有利的是，如上所述采用具有扇出特性的数据结构通常允许对数据项的异构操作的批量并行处理以及对树结构的相应修改要在三个或更少的更新阶段迭代中完成，如下面进一步详细说明所描述。根据多种方法，该特别高效的过程通过降低用于操纵数据和基于树的数据结构本身的计算成本和性能时间来进一步改进实现当前公开的技术的存储系统的功能。

[0103] 基于树的数据结构的并行批量处理

[0104] 如下所述，用于处理数据和基于树的数据结构(例如分页搜索树)的传统技术尚未解决存储在/经由树中的数据的高效并行处理的问题。本文提出的独特方法提出使用异构操作类型的并行批量处理来降低与用于处理/存储在/经由树中的数据的处理和树结构的

处理的典型技术相关的I/O和处理器成本。简而言之，由于目前公开的发明技术能够批量处理异构操作类型、确定树的必要最小结构改变并且在修改数据和/或树结构的过程中向树上传播那些必要的最小结构改变，提供了显着的效率改进。

[0105] 当前公开的发明构思提供了用于对诸如分页搜索树之类的数据结构中的数据条目执行异构操作以及对由执行异构操作产生的数据结构的组织进行任何必要的修改的技术。当前公开的发明构思是独特的，因为异构操作处理是针对树数据结构的每一层并行并且批量执行的，显着地减少了执行该组异构操作所需的处理和I/O操作的数量和成本。

[0106] 通常，本文描述的技术使用两阶段方法来实现上述的并行批量处理。首先，搜索阶段遍历树数据结构，并且定位应该应用异构操作的输入集合中的一个或多个的叶节点。其次，更新阶段以逐层迭代方式遍历树数据结构，从树叶层到根。

[0107] 例如，当前公开的发明构思的一个实现在图7中示意性地示出。如图7所示，其中基于n层树的数据结构包括多个层710a，710b，710c...710n。根层710a具有根节点702，四个内部节点704从该根节点702下降形成另一个节点层710b。类似地，该节点层的节点704从其下降四个节点，形成又一个节点层710c。树以这种方式展开直到到达包括多个叶节点706的第n层710n，每个叶节点706存储一个或多个数据条目。在不脱离本发明构思的范围的情况下，任何数量的层可以介入根层710a和叶节点层710n。

[0108] 实质上，根据本公开执行的处理发生在两个主要阶段中。搜索阶段在根层710a处开始并逐层前进到第n层710n中的叶节点706。

[0109] 相反地，更新阶段在第一次迭代(i＝0)中在叶节点层710n处开始并且向上前进通过一次或多次迭代(i≥1)，可能一直到第n次迭代(i＝n)中的根节点702。当然，在各种实施例中，更新阶段可以例如响应于确定在紧邻的先前更新阶段迭代期间，例如如图7所示的i＝(n-2)，2<i<(n-2)，i>1等的迭代生成的节点和操作集中没有指定节点而在到达根节点层710a之前终止。

[0110] 在优选实施例中，在更新阶段的每次迭代期间，处理数据结构的层，并且在该层中的所有适当的节点上并行并批量地执行适当的操作。更新阶段处理由于应用异构操作而变满或未满的节点。每个更新阶段的输出是树的下一个上层中的一组节点以及将在这些节点上应用的操作，其中这些操作是在树的当前层中的节点上应用的操作的结果。在更新阶段生成的操作将被应用在树的下一个上层的节点上，以支持在当前层的处理中创建、删除并且其条目内容被更新的节点的新形式。然后将当前更新阶段的节点和操作的输出集作为下一个更新阶段的输入，即对树的下一个上层进行处理。当由处理树的一层所产生的节点和操作集为空时，算法的处理完成。

[0111] 相应地，并参照图10，根据一个实施例示出了方法1000的流程图。方法1000可以根据本发明在图1-9所示的任何环境中以各种方式执行。当然，方法1000可以包括比图10中具体描述的操作更多或更少的操作，正如本领域技术人员在阅读本说明书后将会理解的那样。

[0112] 方法1000的每个步骤可以由操作环境的任何合适的组件来执行。例如，在各种实施例中，方法1000可以由分层存储系统的磁盘管理器或其中具有一个或多个处理器的一些其他设备部分地或完全地执行。处理器(例如以硬件和/或软件实现的处理电路、芯片和/或模块)优选地具有至少一个硬件组件可用于任何设备中，以便执行方法1000的一个或多个

步骤。示例性处理器包括但不限于中央处理单元(CPU)、专用集成电路(ASIC)、现场可编程门阵列(FPGA)、其组合或者任何其他的本领域已知的其他合适的计算设备等。

[0113] 方法1000被配置用于在基于树的数据结构上的操作的并行处理，例如图4-6中所示的基于树的结构中的任何一种，以及本领域普通技术人员在阅读本说明书后将会理解的其他类似的基于树的结构。根节点层和叶节点层可以被任意数量的间歇层分开。基于树的结构可以符合B+树，B树或任何其他合适类型的基于树的结构的定义。

[0114] 不管基于树的数据结构的细节如何，如图10所示，方法1000包括操作1002，其中输入由存储基于树的数据结构的存储系统接收。输入标识要应用于基于树的数据结构的一组异构操作。

[0115] 优选地，输入包括对基于树的数据结构的引用以及对的列表，其中每一对列出数据条目和相关联的操作。每个数据条目可以包括密钥和存储数据的有效载荷。

[0116] 在各种方法中，该组异构操作可以包括以下中的任何一个或多个：插入操作、删除操作和更新操作。插入节点可能会导致节点溢出，因此需要对节点进行拆分，而这又可能需要对树叶上方的树层中的条目应用修改。从节点删除可能导致节点变得未满，因此可能需要将该节点与另一个节点合并，这又可能需要对树叶上方的树层中的条目应用修改。为了考虑数据结构组织中的变化，这样的修改可以包括添加、更新或移除路由条目，并且可以传播到树的根节点。更新节点优选地包括更新其中存储的条目。当然，前述示例将被认为是非限制性的，并且异构操作可以包括本领域技术人员在阅读本说明书后将理解的任何适当类型的操作。

[0117] 在一个实施例中，是否以及如何修改数据结构的决定优选地基于节点平衡标准。节点平衡标准可以基于期望的系统存储利用率、性能等来预定义。例如，一个实施例中的节点平衡标准可以包括针对树对应的存储设备的每个页面(节点)的预定义最小值，平均值等存储利用率。

[0118] 在优选的方法中，预定节点平衡标准包括大约50％的最小容量阈值，使得当小于50％的页面容量在使用中时，节点(页面)可被认为是“未满”。节点平衡标准还可以包括大约75％的预定平均容量阈值。

[0119] 当然，也可以确定节点平衡标准而不是预定义，并且可以基于其中实现基于树的数据结构和当前公开的技术的存储系统的特性来实时修改节点平衡标准。

[0120] 存储系统可以被配置为检测将由本领域技术人员阅读本公开内容的技术人员所理解的以下示例性事件或其等价物中的任何一个或多个的发生，并且通过确定适合的新的最佳节点平衡标准来采取相应的行动为变化的情况。

[0121] 另外地和/或可选地，节点平衡标准可以由用户定义或确定。

[0122] 方法1000还包括操作1004，其中确定要应用该组异构操作中的一个或多个的节点，并且优选地是叶节点。该确定优选地至少部分地基于在操作1002中接收到的输入。更优选地，基于定义与其相关联的数据条目和操作的成对列表，操作1004包括确定数据结构的哪些节点需要用一个或多个的异构操作。

[0123] 在特别有利的方法中，在操作1004中执行的确定可以体现为搜索阶段。例如，并且参考例如图6所示的基于树的数据结构，在一个实施例中，搜索阶段可以包括以降序来评估基于树的数据结构的每个层。对于每个层，搜索阶段可以包括并行地全部或部分地基于在

对的列表中指定的数据条目来确定被评估的层中的任何节点是否是应该应用该异构操作的一个或多个的节点或其后代节点。在一些方法中，确定可以排除访问存储数据条目的节点(即，叶节点)，而是基于存储在内部节点中的路由信息。

[0124] 例如，并且根据一个示例性方法，搜索阶段可以包括针对每个输入对并行地降序树，并且获得数据结构中由输入对指定的操作要应用的叶节点的存储标识符。确定应该将异构操作或异构操作集应用于特定叶节点可以包括将存储在节点中的数据条目与由输入对指定并与一个或多个操作相关联的数据条目进行比较。

[0125] 优选地，使用共享许可来访问节点，针对所有输入对来并行执行搜索。搜索阶段可以利用缓存，以进一步最小化存储访问并改善相应存储系统的功能。

[0126] 在附加的和/或替代的方法中，可以在子集中执行搜索阶段，而不是针对每个输入对单独执行搜索阶段。也就是说，在根节点处，输入对可以根据要被访问的下一个较低层中的节点被划分成子集，并且对这些节点中的每一个的访问可以由处理相关子集的操作的不同线程来执行，从而继续下降。以这种方式，当前公开的发明构思允许跨越多个操作处理线程的搜索阶段的并行化，显着减少了执行搜索阶段所需的时间。

[0127] 无论是针对每个输入对还是以组来单独执行，在优选实施例中，存储数据条目的数据结构的叶节点不是在搜索阶段直接访问，而是仅在更新阶段中访问。

[0128] 搜索阶段的输出是“节点和操作集”，其通常标识在搜索阶段确定的节点、应该应用的该组异构操作中的一个或多个，并且优选地包括存储列表在搜索阶段中确定的应该应用该组异构操作中的一个或多个的节点的存储标识符。

[0129] 在各种实施例中，在节点和操作集中，存储标识符列表中的节点的每个存储标识符与输入数据条目和操作(即插入、删除、更新)对的列表相关联，其中该列表将被应用于由关联的存储标识符标识的节点。

[0130] 在更多方法中，搜索阶段可以基本上根据如图8中所描绘的处理流程进行。

[0131] 因此，在各种方法中，方法1000的操作1004可以包括生成包括节点和操作集的输出。节点和操作集包括存储标识符的列表，并且每个存储标识符标识应该应用该组异构操作的一个或多个的一个节点。此外，在一些方法中，每个存储标识符与在操作1002中接收到的输入中标识的至少一个数据条目和操作对相关联。

[0132] 根据一个实施例，方法1000还包括操作1006，其中该组异构操作中的一个或多个并行并批量地执行。树数据结构优选地以迭代、逐层的方式被处理，其中在节点中标识的所有节点和属于特定层的操作集被并行并批量处理。该处理将在下面在本文公开的发明更新阶段的上下文中进一步详细描述。

[0133] 在优选实施例中，将该组异构操作应用于树数据结构的各个节点是经由“更新阶段”的一个或多个迭代来实现的，该更新阶段包括基于树的数据结构的逐层并行批量更新。更新阶段可以通过将该组异构操作中的一个或多个应用于基于树的数据结构的最低层(例如分别如图4和5所示的第二节点层410c和510c，或如图6所示的第n层610n)中的一个或多个节点启动例如如图4-6所示的基于树的数据结构，其中那些叶节点具有在节点和操作集中列出的存储标识符。

[0134] 因此，对于具有在由前一层的处理生成的节点和操作集中列出的存储标识符的第二节点层中的每个节点(即，先前处理的层之上的层)，应用于该节点的该组异构操作可以

基于与节点和操作集中的该节点的存储标识符相关联的数据条目/操作对。

[0135] 重要的是，在一种方法中，该组异构操作被并行并批量地应用于第二节点层中的具有在节点和操作集中列出的存储标识符的每个节点。优选地，并行批量处理包括隔离在当前节点层中正在处理的每个节点，使得能够评估和鉴定适合作为将异构操作应用到当前层中的节点的结果所需的节点平衡操作的相邻节点。

[0136] 在各种实施例中，将异构操作应用于特定节点层中的节点可以涉及在树的一个或多个层中插入新节点；删除树的一个或多个层中的现有节点；合并树的一个或多个层中的现有节点；在树的一个或多个层中分割一个或多个现有节点；和/或更新树的一个或多个层中的现有节点的一个或多个数据条目。

[0137] 如果在叶节点层上方的层上应用异构操作包括插入条目，插入操作优选地执行响应于确定当前层下面的至少一个层中的一个或多个现有节点被分割。此外，插入操作可能需要在基于树的数据结构的当前层中创建一个或多个节点。

[0138] 如果在叶节点层上方的层上应用异构操作包括删除条目，删除操作优选地执行以响应于确定当前层下面的基于树的数据结构的至少一个层中的一个或多个现有节点被合并。删除操作也可能需要删除基于树的数据结构的当前层中的一个或多个节点。

[0139] 在树的当前层的处理期间生成的条目和操作支持在树的当前层的处理中更新、创建和删除的节点的新形式。产生这样条目和操作的三个示例性情况立即在下面提出。

[0140] 首先，在一个实施例中，当被处理的节点被分割并且生成要被添加到树的下一个上层的条目时创建的新节点。

[0141] 其次，在另一个实施例中，从已处理节点到兄弟节点(例如，经由合并)完全传送条目导致处理节点被删除，并且生成引用树的下一个上层中的被删除节点的条目的删除操作。

[0142] 第三，在更多实施例中，条目内容被修改并因此其代表性条目已经改变的经处理的节点产生要删除的条目并插入到树的下一个上层。

[0143] 当然，在各种实施例中，在阅读本说明书的情况下本领域技术人员将理解的其中一个或多个示例性情况可以在处理基于树的单个迭代(或多个迭代)中体验数据结构。

[0144] 一般而言，更新阶段优选地生成标识在下一次迭代期间应当应用的一组操作的输出，其优选地对应于基于树的数据结构的不同层的节点，更优选地是在更新阶段的当前迭代期间被处理层之上的层，并且最优选地在更新阶段的当前迭代期间被处理的层的正上方的层。

[0145] 在一些实施例中，更新阶段至少部分地基于将该组异构操作应用于在更新阶段期间处理的节点层来生成新的节点和操作集作为输出。新节点和操作集包括存储标识符的列表。新节点和操作集中的每个存储标识符标识要更新的下一个节点层中的一个节点。新节点和操作集中标识的下一个节点层中的节点是应该应用该组异构操作的一个或多个的节点。优选地，新节点和操作集中的每个存储标识符与至少一个数据条目和操作对相关联。

[0146] 更新阶段可以包括任何数量的迭代，以从基于树的数据结构的终端节点到根节点的逐渐升高(自下而上)的方式逐层地逐步更新基于树的数据结构。

[0147] 在各种实施例中，迭代更新可以在基于树的数据结构的最低层开始，并且逐渐地更新每个层直到到达包含根节点的层。可选地，更新阶段可以逐步地仅更新基于树的数据

结构的层的子集。

[0148] 在更多方法中，迭代更新过程也可以导致树结构的新层的产生以例如适应节点平衡操作和/或标准。节点平衡操作可能需要生成新的层，例如，响应节点平衡导致根节点分割，需要在前一个根节点之上的层中的新根节点。诸如节点利用阈值之类的节点平衡标准也可能要求创建新的节点，例如，通过要求根节点分割等。

[0149] 相应地，由一个更新阶段迭代生成的节点和操作集可以用作用于下一个更新阶段迭代的输入。优选地，这个新的节点和操作集标识在当前迭代中更新的层之上的层中的一组节点，所标识的节点是应当在下一更新阶段迭代中应用一组异构操作中的一个或多个的节点。

[0150] 在更多的方法中，一个或多个更新阶段迭代还可以包括确定节点或操作输入集是否为空，并且如果是，则完成基于树的数据结构的处理，因为没有进一步的修改必须应用于基于树的数据结构的上层。在一些实施例中，处理的完成可涉及释放对根节点的许可，如果这样的许可先前在根节点上被保护。在进一步的实施例中，这样的许可可以是独占许可。

[0151] 如果节点和操作输入集不为空，则这里讨论的技术可以包括创建新的空节点和操作集；在下一个更新阶段迭代中填充应用了一组异构操作的一个或多个的节点的存储标识符设置的空节点和操作集；以及将存储标识符与要在下一更新阶段迭代中在相应节点上执行的一个或多个异构操作的适当集合相关联。

[0152] 在一个实施例中，更新阶段迭代可以包括将该组异构操作集中的一个或多个应用到具有在新节点和操作集中列出的存储标识符的特定节点层中的每个节点，该新节点和操作集对于后续的($i \geq 1$)更新阶段迭代在先前更新阶段迭代期间生成，或者对于第一次更新阶段迭代($i = 0$)在搜索阶段期间生成。优选地，在具有在新节点和操作集中列出的存储标识符的特定节点层中的节点包括叶节点的情况下，应用于叶节点的异构操作基于与用于该叶节点的存储标识符相关联，并作为输入提供给存储系统的至少一个数据条目和操作对。此外，异构操作并行并批量地应用于具有在新节点和操作集中列出的存储标识符的特定节点层中的每个节点。

[0153] 如图4所示，并且根据优选实施例，该迭代更新阶段过程包括在第一次迭代($i = 0$)期间更新第一节点层410c中的节点406，生成新的节点和操作集标识应该应用异构操作的第二节点层410b中的节点404，以及在更新阶段的第二次迭代($i = 1$)中更新第二节点层410b的节点404。

[0154] 在特别优选的实施例中，第一节点层410c中的节点406是存储数据条目的叶节点，并且是B+树数据结构的一部分。第二节点层410b中的节点404是存储路由信息的内部节点。

[0155] 当然，在本公开的范围内的另外的实施例可以包括具有存储数据条目和/或路由信息的多层节点、具有在树的多层处存储数据条目的节点(例如如通常在图5和6中所示)等的基于树的数据结构。尤其是根据这些实施例，迭代更新阶段可以包括更新基于树的数据结构的任何层上的节点的数据条目，更新基于树的数据结构的任何层上的节点的路由信息，修改基于树的数据结构的任何层上的节点之间的关系，在基于树的数据结构的任何层上的节点之间传递信息等，如本领域普通技术人员在阅读本描述时将理解的。

[0156] 不管特定的树结构如何，根据各种实施例，更新阶段可以包括确定原始节点和/或新的节点和操作集是否列出与树的下一个上层中的节点相对应的任何存储标识符。响应于

确定节点和操作集不包括对应于树的下一个上层中的节点的任何存储标识符，优选地终止更新阶段。

[0157] 在特别优选的方法中，在搜索阶段中使用共享许可来访问节点，并且在更新阶段中使用独占许可来访问节点。

[0158] 对特定层中的节点应用异构操作并行并批量地发生，这涉及使用不同的处理线程将操作应用于每个节点。更优选地，每个处理线程并行运行以减少计算时间并通过加速在基于树的数据结构中定位和更新数据的过程来改进实现当前公开的技术的存储系统，而所有这些都需要较少的计算成本。

[0159] 例如，通过批量处理异构操作，并且在单个迭代过程中适应任何必要的节点平衡或树结构的其他操作，当前公开的技术避免了在树上应用特定类型的操作的需要，然后执行另一次搜索以定位用于不同类型的操作的数据条目以确定不同类型的操作所针对的数据条目中的任何数据条目是否在与跨树施加特定类型的操作之前不同的位置处。

[0160] 根据各种实施例，在更新阶段期间的处理节点可以以允许异构操作的并行批量处理以任何合适的方式跨基于树的数据结构的特定层中的多个节点的执行。在一个特别优选的实施例中，更新阶段期间的处理基本上如图9所示。

[0161] 在优选实施例中，与输入节点和操作集中的节点相关联的操作以特定的顺序执行，这赋予当前公开的发明构思额外的效率，并进一步改进存储系统本身的功能。操作顺序将在下面进一步详细讨论。

[0162] 在初始更新阶段期间，可以处理包括叶节点的层的迭代可以根据用户提供的操作顺序来处理与和由搜索阶段生成的输入节点和操作集中的节点相关联的操作，或者根据这里指定的优选顺序。这里指定的优选顺序有利地使由于在给定节点上应用操作而导致的结构变化最小化，因此有助于算法的效率并进一步改进其中实施算法的存储系统的功能。不管实施的具体顺序如何，更新阶段可以通过获得对根节点的独占访问来启动。

[0163] 优选的顺序包括执行更新操作，之后是删除操作，以及随后的插入操作。

[0164] 更具体地，要执行的第一操作优选地是更新节点中的数据条目的有效载荷。第二个操作是从节点删除条目。此时，在完成所有输入操作的处理之前，不需要执行进一步的节点平衡或合并操作。实际上，优选地，不执行平衡或合并操作，除非正在处理的节点由于应用所有输入操作而变得未满(under-filled)。第三个操作是将条目插入到节点中。在插入的情况下，在应用所有输入操作之前，节点可能变满，并且因此在这些情况下，如果节点变满，则可能需要在插入期间应用节点重新平衡或分割。

[0165] 在各种实施例中，可以通过将节点利用水平(即，存储在节点中的数据量)与节点的最大容量进行比较来确定节点变满。

[0166] 类似地，通过将节点利用率水平与最小节点利用率阈值进行比较，可以确定节点变得未满。最小节点利用率阈值可根据用户偏好或存储系统的特定需求(例如，存储利用率和/或性能要求)。在优选的方法中，最小节点利用率阈值大约是总节点容量的50％。当然，在不脱离本公开的范围的情况下，可以使用25％，30％，35％等小于100％的其他值。

[0167] 为了解决节点变满的情况，可以采用节点重新平衡。优选地，该节点重新平衡包括确定该满节点是否具有任何相邻节点，优选地在节点中未标识的相邻节点以及作为当前迭代的输入而提供的操作集。响应于确定这样的相邻节点存在，节点重新平衡包括确定从该

满节点传送条目是否将导致相邻节点本身变满。响应于确定相邻节点不会变满，节点重新平衡包括确定在输入节点和操作集中待决的进一步插入是否由于条目的传送而不必在相邻节点上应用(所有插入必须应用于在输入节点和操作集中标识的节点而不是相邻节点)。响应于确定不需要在邻近节点上应用进一步的插入，节点重新平衡包括确定相邻节点是否将由于条目的传送而变未满。响应于由于条目的传送而确定相邻节点不会变满，节点重新平衡包括确定到相邻节点的条目的传送是否将导致该满节点具有在指定的节点利用率内的节点利用率水平范围(例如高于最小节点利用率阈值)。响应于确定转移将完成该结果，条目优选地被转移。更优选地，可以传送多个条目，以便为正在处理的节点中的附加未决的插入创建容量。

[0168] 更优选地，仅在传送期间锁定(例如，通过独占访问)条目被传送到的节点。最优选地，仅在所有插入被应用到被处理的层中的节点之后才执行该节点重新平衡，并且没有向接收所传送的条目的相邻节点应用插入。

[0169] 在更多实施例中，节点重新平衡以减轻满节点可包括分割节点，例如，如果由于上面列出的任何判定都被否定而上述传送程序是不可能的。

[0170] 值得注意的是，在一些情况下，将一组异构操作应用于基于树的数据结构的特定层的节点可能导致若干次发生正在处理的节点在该更新阶段迭代的过程中变满，并且因此每次更新阶段迭代可能需要多次执行节点重新平衡。

[0171] 根据若干实施例，在更新阶段迭代期间变得未满的节点可以以类似的方式被重新平衡。例如，在一个实施例中，节点重新平衡以减轻未满的节点的问题涉及确定未满的节点是否具有在节点中未被标识的任何相邻节点以及被提供为当前迭代的输入的操作集。

[0172] 在更多实施例中，并且响应于确定这样的相邻节点存在，节点重新平衡包括确定是否从相邻节点传送条目将导致相邻节点变得未满。

[0173] 在又一些实施例中，并且响应于确定相邻节点不会变得未满，节点重新平衡包括确定到未满的节点的条目的传送是否将导致未满的节点具有节点利用率水平在指定的节点利用率范围内(例如高于最小节点利用率阈值)。响应于确定传送将完成此结果，条目优选地被传送。

[0174] 更优选地，在一些方法中，传送条目的节点仅在传送期间被锁定(例如，通过独占访问)。最优选地，该节点重新平衡涉及仅传送使未满的节点处于期望的节点利用范围内所需的多个条目或一定量的信息。

[0175] 在更多实施例中，节点重新平衡以释放未满的节点可以包括合并未满的节点，例如，如果由于上面提出的一个或多个判定结果是否定的而上述传送程序是不可能的。在这种情况下，合并可能包括一系列决定。例如，在一种方法中，合并节点涉及确定未满的节点是否具有未包括在输入节点和操作集中的相邻节点。响应于确定存在这样的相邻节点，合并未满的节点还包括：确定相邻节点是否由于将所有条目从未满的节点传送到相邻节点而将变满。响应于由于将所有条目从未满的节点传送到相邻节点而确定相邻节点不会变满，则合并未满的节点还包括将所有条目从未满的节点的传送到相邻的节点。在将所有条目从先前未满的节点传送之后，可以删除现在为空的该节点。条目可以从未满的节点传送到多于一个的相邻节点，以清空未满的节点。

[0176] 有利的是，上面刚刚讨论的合并操作中的传送的方向性，即从未满的节点向相邻

的节点传送条目，而不是从相邻的节点到未满的节点传送，用于防止需要删除相邻的节点参与合并操作。这是有益的，因为其他并行操作可能需要同时访问这些相邻节点，并且在一些方法中，该访问必须是可能的，因为这样的节点仍然是从另一个上层树引用的。另外，方向性期望地导致其中并行操作不需要被删除节点的情况，保持与其他并行操作的一致性。

[0177] 在优选实施例中，当当前节点的处理完成时，如果节点上先前获取了独占许可，则被释放。另外，在处理特定树层期间，用于定位相邻节点目的的对另一树层(例如上层树层)的访问由所有并行操作(例如，使用共享权限)并行执行。

[0178] 如上所述，当前层的节点上的操作的并行处理的输出是新的节点和操作集，其包括应该应用操作的树的下一层中的节点的存储标识符的列表。在一些实施例中，新节点和操作集可以由存储标识符的列表和与每个存储标识符相关联的一组一个或多个异构操作组成。

[0179] 优选地，应该应用于树的下一层的新节点和操作集中定义的操作是在当前层中的节点应用操作的结果。在这样的实施例中，新节点和操作集包括对的列表，每对包括条目(例如，数据或路由条目)，以及要应用的相关操作(例如，插入、删除、更新等)。每个这样的对列表可以有利地与应该应用列表中的操作的节点的存储标识符相关联。

[0180] 再次，在处理树的当前层期间生成的条目和操作有利地支持在处理树的当前层的过程中更新、创建和删除的节点的新形式。

[0181] 在各种实施例中，可能生成支持节点的新形式的条目和操作的情况包括在处理的节点被分割时创建新节点。这将生成要添加到树的下一个上层的条目。这些条目被包括在当前树层处理期间生成的节点和操作集中。

[0182] 在更多实施例中，可能生成支持节点的新形式的条目和操作的情况包括条目的传送，特别是从已处理节点到相邻节点的条目的传送(例如，经由合并操作)。特别地，在这种传送导致被处理节点被删除的情况下，由于对被删除节点的引用现在是无效的，因此要求删除引用该树的下一层中的被删除节点的条目可能是有利的。

[0183] 在更多实施例中，可能产生支持节点的新形式的条目和操作的情况包括更新操作，这些更新操作涉及以引起节点的代表性条目改变的方式修改条目。在这种情况下，生成用于从树的下一个上层删除和/或插入的条目是有利的。本领域普通技术人员在阅读本说明书后将会理解，这种删除和插入可以通过确保代表性条目根据修改的树结构正确地标识数据和/或路由信息来帮助保持树结构和路由条目内的一致性。

[0184] 在优选实施例中，对于特定层执行接收到的输入中指定的一组异构操作，针对特定层执行任何节点平衡操作以及在完成这些操作时生成新节点和操作集并且输出用于更新阶段的后续迭代中。

[0185] 当对于基于树的数据结构的下一层不需要操作时，新节点和操作集将是空的。因此，更新阶段的每次迭代可以包括确定节点和操作集是否为空，并且响应于如此确定，终止更新阶段。更新阶段的终止可以包括和/或之后释放对基于树的数据结构的根节点的独占访问。

[0186] n层B+树的并行批量处理

[0187] 在涉及n层B+树的更具体的情况下，可以类似地利用本文描述的技术来显着提高其中将n层B+树实现为数据结构的数据存储系统的性能。计算机实现的方法被设计为便于

并行处理分页搜索树数据结构上的异构操作。

[0188] 该方法优选地包括在其中存储基于树的数据结构的存储系统处接收输入;以及基于所述输入并行并且批量执行多个异构操作到一个或多个所述叶节点。输入标识要应用于分页搜索树数据结构的一组异构操作。

[0189] 与方法1000一样,在各种实施例中,更具体的实现可以涉及搜索阶段。搜索阶段优选地包括:以从根节点层到叶节点层的顺序评估基于树的数据结构的每个层;同时确定应该应用该组异构操作的叶节点。

[0190] 另外,搜索阶段以节点和操作集的形式生成输出,其包括存储标识符的列表。每个存储标识符优选地标识应该应用该组异类操作的一个或多个的一个叶节点;并且每个存储标识符与在所接收的输入中标识的至少一个数据条目和操作对相关联。

[0191] 再次以与方法1000类似的方式,该方法可以包括更新阶段,其需要将该组异构操作中的一个或多个应用于叶节点层(例如图4所示的叶节点层410c)中的每个叶节点,叶节点层具有在节点和操作集中列出的存储标识符。更具体地说,在该第一次迭代(即,i=0)中,对于具有在节点和操作集中列出的存储标识符的每个叶节点,应用于特定叶节点的该组异构操作优选地基于与该叶节点的存储标识符相关联的数据输入和操作对。此外,在优选的方法中,该组异构操作并行并且批量地应用于在节点和操作集中列出存储标识符的每个叶节点。

[0192] 更新阶段可以为完成的每个迭代生成新的节点和操作。新节点和操作集识别比当前迭代中处理的层高的层中的节点以及要在这些节点上执行的操作。优选地,该定义基于应用于当前层的操作的结果。在每次迭代中产生这个输出涉及至少部分地基于将异构操作应用于具有在节点和操作集中列出的存储标识符的节点来生成新的节点和操作集。

[0193] 在一个实施例中,在第一次迭代期间生成的新节点和操作集因此优选地包括存储标识符的列表,并且每个存储标识符标识B+树的下一个上层节点层中的内部节点之一应该应用该组异构操作的一个或多个。相应地,在下一个更新阶段迭代(i=1)中,该组异构操作被应用于具有在新节点和操作集中列出的存储标识符的下一个上层节点层中的内部节点。优选地,该组异构操作并行并且批量地应用于第二节点层中的每个内部节点,其具有在节点和操作集中列出的存储标识符。

[0194] 上述迭代过程可以继续,以逐层的方式向B+树的条目传播任何必要的修改和/或对B+树结构的修改,直到不需要进一步的操作为止,节点和操作集被确定为空,并且因此更新阶段终止。优选地,该组异构操作并行并且批量地应用于具有在先前更新阶段迭代期间生成的节点和操作集中列出的存储标识符的特定层中的每个节点。

[0195] 在树操作期间隔离并行操作

[0196] 目前公开的发明构思通过减少更新数据条目和树结构的计算成本来改进利用基于树的数据结构的常规存储系统的功能。如上所述,这种提高效率的重要方面源于用于在基于树的数据结构的上下文中异构操作的并行批量处理的发明技术。

[0197] 目前公开的发明技术的另一方面通过隔离在基于树的数据结构的特定层中正在处理的节点的处理来促进进一步的计算效率,以促进这些异构操作被同时并批量处理。在各种实施例中,这种隔离技术通常涉及限定那些可以从当前更新阶段迭代中正在处理的节点接收条目和/或给出条目的节点。

[0198] 实际上，这使得更新过程能够有效地处理满和未满的节点，并且通过这样做便于以独立和隔离的方式处理每个节点。尤其是当与各种操作处理线程的并行化相结合时，这种隔离提高了并行处理给定树层的节点所贡献的效率。

[0199] 一般而言，目前公开的创造性隔离过程通过限定要考虑参与节点再平衡和其他结构修改操作的节点来隔离在输入节点中指定用于处理的节点的处理和为特定层设置的操作特别是传送条目。

[0200] 例如，根据一个方面，可以限定的节点是在其上执行的操作没有任何依赖性的节点，使得节点可以参与传送操作、合并操作等，而不干扰包括整个树的修改/更新过程的其他操作。实际上，这些合格的节点包括(1)与被处理的节点相邻的节点，(2)本身不包括在要处理的节点的输入集中，以及(3)满足一个或多个附加的资格标准，具体情况。这些额外的资格标准在下面进一步详细描述。

[0201] 在其中条目可以从处理的节点传送到合格节点的一个实施例中，额外的资格标准包括建议的传送是否会导致相邻节点变满。如果是这样的话，那么该节点可能不符合传输条件，否则该节点可能有资格进行传输。

[0202] 相反地，在一个实施例中，当从其传送条目的节点本身不会由于传送而变成未满时，节点可以有资格参与从节点到相邻节点的条目传送。因此，额外的资格标准通常可以涉及节点的期望的利用范围，并且资格认证过程可以优选地包括对照期望的利用范围评估节点利用级别，确定建议的传送是否会导致违反期望的利用范围，并且响应于确定提议的传送的合格节点将不会导致这样的违反。

[0203] 在更多的实施例中，这种情况下的附加资格标准可以包括传送是否将要求进一步的未决插入操作被应用于提出传送的相邻节点。如果是这样，那么节点可能不符合条件，否则可能有资格。

[0204] 在又一种情况下，额外的资格标准可以包括所提议的合并操作的方向性。

[0205] 更具体地说，资格认定可以包括确定所处理的节点与相邻节点的建议合并是否涉及将所处理的节点的所有条目传送到相邻节点中，反之亦然。响应于确定所提出的合并涉及将所处理的节点的所有条目传送到相邻节点中，相邻节点可以有资格传送条目。否则，相邻的节点可能没有资格。

[0206] 此外，所允许的合并的单向性意味着经处理的节点而不是相邻节点在合并操作之后受到删除。优选地，相邻节点不被合并操作删除。

[0207] 不管特定的场景和资格标准如何，在优选的方法中，通过独占许可来锁定相邻节点，以使传送能够与其他操作并行执行。然而，为了最小化与这种独占访问相关联的延迟，相邻节点优选地被暂时锁定，并且甚至更优选地仅在传送操作的持续时间内被锁定。在完成传送后，节点的独占访问被释放，并且其他并行处理可以以有效的方式继续。类似地，通过修改操作的处理的节点优选地仅通过独占访问仅被暂时锁定，并且更优选地仅在特定修改操作的持续时间内被锁定。

[0208] 如本领域普通技术人员在阅读本说明书后将理解的那样，当然可能的情况是其中由一个或多个异构操作处理的特定节点可能不具有适合于容纳如本文所述的传送。在这样的情况下，当处理结果变为已处理节点变满时，并且响应于确定不存在有资格从处理节点传送条目的相邻节点，则处理节点优选被分割。值得注意的是，可以有几个这样的分割操作

发生，特别是在处理节点上的插入操作的处理期间。

[0209] 值得注意的是，根据各种方法，当前公开的发明实施例通过水平并行处理实现了极好的并发性，而不需要添加间接地址映射。添加的间接地址映射不必要地引入额外的开销和额外的资源消耗(例如，额外的I/O操作、处理时间、存储等)，其避免改进存储系统的功能。

[0210] 现在参考图11，根据一个实施例示出了用于隔离节点以促进其并行批量处理的方法1100的流程图。方法1100可以根据本发明在图1-6所示的任何环境中、在各种实施例中执行。当然，方法1100中可以包括比图11中具体描述的操作更多或更少的操作，如本领域技术人员在阅读本说明书后将理解的那样。

[0211] 方法1100的每个步骤可以由操作环境的任何合适的部件执行。例如，在各种实施例中，方法1100可以由存储系统管理器或其中具有一个或多个处理器的一些其他设备部分地或全部地执行。处理器(例如以硬件和/或软件实现且优选地具有至少一个硬件组件的处理电路、芯片和/或模块)可用于任何设备中以执行一个或多个示例性处理器包括但不限于中央处理单元(CPU)、专用集成电路(ASIC)、现场可编程门阵列(FPGA)等等，其组合或者任何其他的本领域已知的其他合适的计算设备。

[0212] 如图所示，如图11所示，方法1100可以以操作1102开始，其中输入在管理存储基于树的数据结构的存储设备的存储系统处被接收。输入标识要应用于基于树的数据结构的一组异构操作。

[0213] 方法1100还包括操作1104，其中标识或以其他方式确定将应用该组异构操作中的一个或多个的基于树的数据结构的一个或多个节点。在各种实施例中，可以以与本描述一致的任何合适的方式来完成确定。在优选的方法中，如上所述，确定是基于在基于树的数据结构中搜索输入条目以确定它们的容纳节点，或者由搜索阶段或先前更新阶段迭代产生的节点和操作输出。

[0214] 另外，方法1100包括操作1106，其中将被应用于一个或多个节点的该组异类操作一个或多个组被标识或以其他方式确定。在各种实施例中，可以以与本描述一致的任何合适的方式来完成确定。优选地，以各种方式基于哪些节点将要应用一个或多个操作来完成确定，这可以基于包括在节点和操作集中的数据来实现。

[0215] 在优选方法中，如上所述，确定至少部分地基于从搜索阶段或先前更新阶段迭代输出的节点和操作。具体地，如本领域普通技术人员在阅读本说明书后将理解的那样，确定可以包括根据应用异构操作的节点对异构操作进行分组。

[0216] 在更优选的方法中，节点和操作集合中的每个节点占据基于树的数据结构的相同层。

[0217] 在操作1108中，方法1100包括隔离节点和操作集中的每个节点的处理用于独立处理。如本文所理解的，节点处理隔离包括被配置为使得能够使用一个或多个异构操作来处理节点的任何合适的技术或机制，其中处理独立于处理节点和操作集中的其他节点上的异构操作而发生。优选地，节点隔离包括隔离节点和操作集中的每个节点，使得将要应用于集合中的节点的整组异构操作可以被执行，而不会干扰在树的其他节点，特别是当前层上的异构操作的处理。例如，在一个实施例中，孤立节点的独立处理包括处理不同处理线程上的每个节点。

[0218] 因此，方法1100还包括操作1110，其中使用被确定为应用于该组节点的该组异构操作集合中的一个或多个组来处理该组节点和操作中的每个节点。节点集中的每个节点最好并行独立处理。

[0219] 优选地，并行处理所有节点，使得每个处理线程基本上同时执行。当然，某些线程可能比其他线程需要更长的时间才能完成，但是本领域普通技术人员在阅读本说明书后将会理解，并行的独立处理涉及这样的实施例，其中该集合中的所有线程的处理基本上在同时，在一个基本相同的时间窗口内处理所有的线程。

[0220] 如本领域普通技术人员在阅读本说明书后将理解的那样，上述方法1100设想了使用一个或多个异构操作的一组确定的一组节点的独立的、隔离的并行处理以应用于集合中的节点。当然，方法1100可以包括以与上述类似的方式将多个不同组的异构操作处理到多个不同组的节点。

[0221] 例如，在一个示例性实施例中，方法1100可以包括将多个不同组的操作处理到不同组的节点，其中各个组中的所有节点占据基于树的数据结构的单个层。为了简单起见，假定示例性的基于树的数据结构具有包括四个节点N1，N2，N3和N4的层(例如，如图4-6所示)。

[0222] 在这样的示例性场景中，方法1100可以包括确定包括更新操作的一组异构操作，并且插入操作应当被应用于包括N1，N2和N3的一组节点，以及确定不同组的异构操作包括应当被应用于包括N3和N4的一组节点的更新操作和删除操作。本领域技术人员在阅读本说明书后将会理解，可以确定任何数量的这样的组和集合，并且相应的节点被隔离用于在此讨论的并行的独立处理。

[0223] 当然，方法1100可以包括任何数量的附加和/或替代特征，诸如上面讨论的特征以及下面阐述的说明性特征。

[0224] 在一个实施例中，方法1100可以包括确定一个或多个节点是否有资格参与节点重新平衡操作。优选地，该确定基于节点重新平衡标准，并且具体可以包括以下考虑的任何组合、置换或合成。

[0225] 在一种方法中，确定一个或多个节点是否有资格参与节点再平衡操作包括：识别与该组节点中的至少一个节点相邻的一个或多个节点；确定是否从该组节点中排除所述一个或多个相邻节点中的任何节点；以及响应于确定所述一个或多个相邻节点中的所述至少一个被排除在该组节点之外，限定所述一个或多个相邻节点中的至少一个相邻节点。优选地，相邻节点不是存储系统接收的输入中指定的任何操作的目标。

[0226] 在优选实施例中，节点重新平衡操作至少包括在该组节点中的节点与从该组节点中排除的节点之间的条目的传送。当然，方法1100可以类似地包括节点重新平衡操作，诸如更新操作、分割操作、合并操作等，如本领域普通技术人员在阅读本说明书后将理解的那样。

[0227] 在条目的传送包括将条目从该组节点中的节点传输到该组节点中排除的节点的实施例中，确定一个或多个节点是否有资格参与节点再平衡操作可包括一个或更多的以下组分操作。在一种方法中，确定节点是否合格包括确定从该组节点中排除的节点是否由于条目传送而变满；以及响应于确定从该组节点中排除的节点将不会因所述条目的传送而变满而使从该组节点中排除的节点被限定用于所述传送。

[0228] 此外，确定从该组节点中排除的节点是否由于条目的传送将变满可以包括：估计

传送之后从该组节点中排除的节点的利用率水平;以及将从该组节点中排除的节点的估计利用率水平与最大利用率阈值或节点的存储容量进行比较。在估计的利用率水平未超过最大利用率阈值或节点的存储容量的情况下,节点可以是合格的。在估计的利用率水平确实超过最大利用率阈值或节点的存储容量的情况下,节点优选地不合格。

[0229] 以类似的方式,并且对于条目的传送包括将条目从该组节点中排除的节点的条目传送到该组节点中的节点的实施例,确定一个或多个节点是否有资格参与节点再平衡操作可能包括以下内容。在一个实施例中,该过程涉及确定从公共节点集合中排除的节点是否由于条目的传送而变得未满;以及响应于确定从该组节点中排除的节点将不会由于条目的传送而变得未充满,限定从该组节点排除的节点用于所述传送。

[0230] 因此,确定从该组节点中排除的节点是否由于条目传送的结果将被填充不足,可以包括:估计传送之后从该组节点中排除的节点的利用率水平;以及将从该组节点排除的节点的估计利用率水平与最小利用率阈值进行比较。在估计的利用率达到或超过最小利用率阈值的情况下,节点可以是合格的。在估计的利用率水平未达到或超过最小利用率阈值的情况下,节点优选地不合格。

[0231] 在又一些实施例中,节点重新平衡操作可以包括合并操作。因此,该方法可以包括将该组节点中的节点中的所有条目转移到从该组节点中排除的节点。确定一个或多个节点是否有资格参与涉及合并操作的节点重新平衡操作优选地包括:确定从该组节点中排除的节点是否由于条目传送而变满;以及响应于确定从该组节点中排除的节点将不会因所述条目的传送而变满而使从该组节点中排除的节点被限定用于所述传送。为了完成合并操作,在将所有条目传送到从该组中排除的节点之后,删除从其转移条目并且变为空的节点集合中的节点。

[0232] 如上所述,在各种实施例中,节点重新平衡可包括获得对参与节点重新平衡操作的资格的节点的独占访问。优选地,在发起节点重新平衡操作涉及的节点的更新之前,获得独占访问;并且更优选紧接在发起节点重新平衡操作中涉及的节点的更新之前。该过程的示例性实施例可以包括使用节点有资格参与节点再平衡操作来执行节点再平衡操作;以及在完成节点重新平衡操作时,释放对符合参与节点重新平衡操作的节点的独占访问。因此,独占访问优选地存在与节点重新平衡操作的持续时间大致相等的时间量。

[0233] 在甚至更多实施例中,节点再平衡操作可以包括例如根据分割操作从该组节点中的节点传送条目。确定一个或多个节点是否有资格参与节点再平衡操作可因此包括:识别与该组节点中的至少一个节点相邻的一个或多个节点;确定是否从该组节点中排除该一个或多个相邻节点中的任何节点;确定从该组节点中排除的相邻节点是否由于条目的传送将变满;以及响应于确定从该组节点中排除的所述相邻节点将由于条目的传送而变满而将该组节点中的节点分割。

[0234] 当然,以上所述仅是本发明技术中的并行批量处理节点的示例性实施例,不应认为是对本发明的范围的限制。

[0235] 有效的操作顺序

[0236] 如上所述,当前公开的发明构思还通过提供将被应用于节点的异构类型的操作的新颖且有效的排序,在常规存储系统架构上赋予改进的功能:异构操作根据它们影响的节点分组,然后根据他们的类型在每组内按照以最大限度减少由于应用操作导致的结构变化

的特定顺序排序。一个有利的结果是显着地减小了由算法的每个阶段产生的输出操作集的大小，从而减少了完成基于树的数据结构的处理所需的操作的总数并且有助于提高由该算法目前的技术所赋予的计算效率。

[0237] 例如，在一个实施例中，特定的一组节点受更新操作的影响，其中特定组中的每个节点的条目将被修改。节点被分组处理，并且在该组内进行任何由于执行更新操作(例如插入、删除等)而需要的结构改变。

[0238] 在确定完成更新所需的一组操作以及任何期望的节点平衡之后，该组操作优选根据此处所述的优选顺序来执行。再次，优选的顺序包括执行更新操作，随后执行删除操作，以及随后执行插入操作。然而，在其它实施例中，用户定义的顺序可以是强制的(例如，在接收到的输入中)并且被实施。

[0239] 因此，如图所示，如图12所示，方法1200可以以操作1202开始，其中输入在管理存储基于树的数据结构的存储设备的存储系统处被接收。输入标识要应用于基于树的数据结构的一组异构操作。

[0240] 方法1200还包括操作1204，其中标识或以其他方式确定要应用该组异构操作中的一个或多个的基于树的数据结构的一个或多个节点。在各种实施例中，可以以与本描述一致的任何合适的方式来完成确定。在优选的方法中，如上所述，确定是基于在基于树的数据结构中搜索输入条目以确定它们的容纳节点，或者由搜索阶段或先前更新阶段迭代产生的节点和操作输出。

[0241] 另外，方法1200包括操作1206，其中将要应用于一个或多个节点的该组异类操作中一个或多个组被标识或以其他方式确定。在各种实施例中，可以以与本描述一致的任何合适的方式来完成确定。

[0242] 在优选方法中，如上所述，确定至少部分地基于从搜索阶段输出的节点和操作，和/或从更新阶段的先前迭代输出的节点和操作集。具体来说，如本领域普通技术人员在阅读本说明书后将理解的，所述确定可以包括基于其存储标识符来识别节点集合，并且确定所述节点集合全部针对插入操作、删除操作、合并操作、拆分操作、更新操作等中的一个或多个。

[0243] 在更优选的方法中，一个或多个节点的集合中的每个节点占据基于树的数据结构的同一层。

[0244] 方法1200还包括操作1208，其中应用了该组异构操作中的一个或多个组。重要的是，根据预定义的顺序应用在每个组中应用所应用的一组操作。优选地，预定义顺序是如本文所述的优选顺序。

[0245] 当然，如本领域技术人员在阅读本说明书后将理解的，方法1200还可以包括如本文所述的任何数量的附加功能和/或特征。在各种实施例中，方法1200可以包括以下特征和/或功能中的任何一个或多个。

[0246] 有利地，包括根据预定顺序将一组异构操作应用于节点集合的当前公开的发明实施例通过以下方式来改进存储系统的功能：最小化基于树的数据结构的结构修改的数量；以及减小作为将该组异构操作应用于所述基于树的数据结构的特定层而产生的输出节点和操作集的大小。

[0247] 如上所述，预定义的顺序包括执行应用更新操作，随后执行删除操作，随后执行插

入操作。因此，在一个实施例中，方法1200可以包括在执行该组异构操作中识别的任何删除操作之前执行该组异构操作中识别的任意更新操作，并且在执行该组异构操作中识别的任何插入操作之前执行该组已购操作中识别的任意删除操作。

[0248] 类似地，并且在更多实施例中，预定义顺序可以包括：在优选地执行对于某一特定节点的该组异构操作中识别的所有更新、删除和插入操作之后，执行任何节点重新平衡操作(例如条目的传送、分割或合并操作)。执行节点重新平衡优选地是响应于在执行输入中指定的操作组之后确定节点集合中的节点充满或未满而执行的。

[0249] 在进一步的方法中，预定义顺序包括：响应于确定节点集合中的节点在执行该组操作的过程中变满，执行拆分操作。

[0250] 此外，如上所述，在一些实施例中，操作的顺序可以由用户指定，特别是当操作要被应用于存储数据条目的一组节点时，甚至更具体地当操作在如如本文所述的更新阶段的第一次迭代期间要被应用于B+树的一组叶节点时。

[0251] 因此，方法1200可以附加地和/或可替换地包括：响应于确定在输入中指定了用户提供的顺序，覆盖预定义的顺序，用户提供的输入对应于应用于存储数据条目的基于树的数据结构的该组异构操作中的一个或多个。

[0252] 当然，以上仅仅是用于隔离节点进行并行批量处理的本发明技术的示例性实施例，不应被认为是限制本公开的范围。

[0253] 使用前述的发明构思，当前公开的技术实现并提供了大量的有益特征，其单独地和组合地操作以改进常规数据存储系统的功能，例如，通过减少存储系统访问操作、I/O和处理器负载。更具体地说，目前公开的发明构思能够以针对异构操作集合的高效并行处理而优化的方式批量处理异构类型的操作。另外，隔离在当前树层中正在处理的每个节点的处理使得能够高效地并行批量处理操作。此外，通过定义操作的最大有效顺序来最小化由应用操作导致的结构变化，从而有助于算法的效率。

[0254] 因此，当前公开使用前述的发明构思，当前公开的技术实现并提供了大量的有益特征，其单独地和组合地操作以改进常规数据存储系统的功能，例如，通过减少存储系统访问操作，I/O和处理器负载。更具体地说，目前公开的发明构思能够以针对异构操作集合的高效并行处理而优化的方式批量处理异构类型的操作。另外，隔离在当前树层中正在处理的每个节点的处理使得能够高效地并行批量处理操作。此外，通过定义操作的最大有效顺序来最小化由应用操作导致的结构变化，从而有助于算法的效率。的发明构思适合于并且为使用事务来访问和修改数据结构的应用提供全面的解决方案，这是许多使用情况和应用非常常见的架构，所述使用情况和应用累积操作并且然后在应用于数据结构之前可能减少操作。更具体地说，目前公开的发明构思适用于分页搜索树数据结构的一般族，并且提供了具有广泛适用范围的解决方案，以提高在许多应用中广泛使用的组件的效率。然而，应该理解的是，这些公开内容呈现适用于许多使用基于树的数据结构的系统的概念，例如数据库系统、文件系统、存储和重复数据删除系统、因此具有广泛的适用性。

[0255] 将清楚的是，可以以任何方式组合上述系统和/或方法的各种特征，从这里给出的描述创建多个组合。将进一步认识到，可以以代表客户部署的服务的形式来提供本发明的实施例以按需提供服务。

[0256] 虽然下面已经描述了各种实施例，但是应该理解，它们仅仅是作为例子而不是限

制。因此，优选实施例的宽度和范围不应该被任何下面描述的示例性实施例限制，而是应该仅根据下面的权利要求及其等同物来限定。

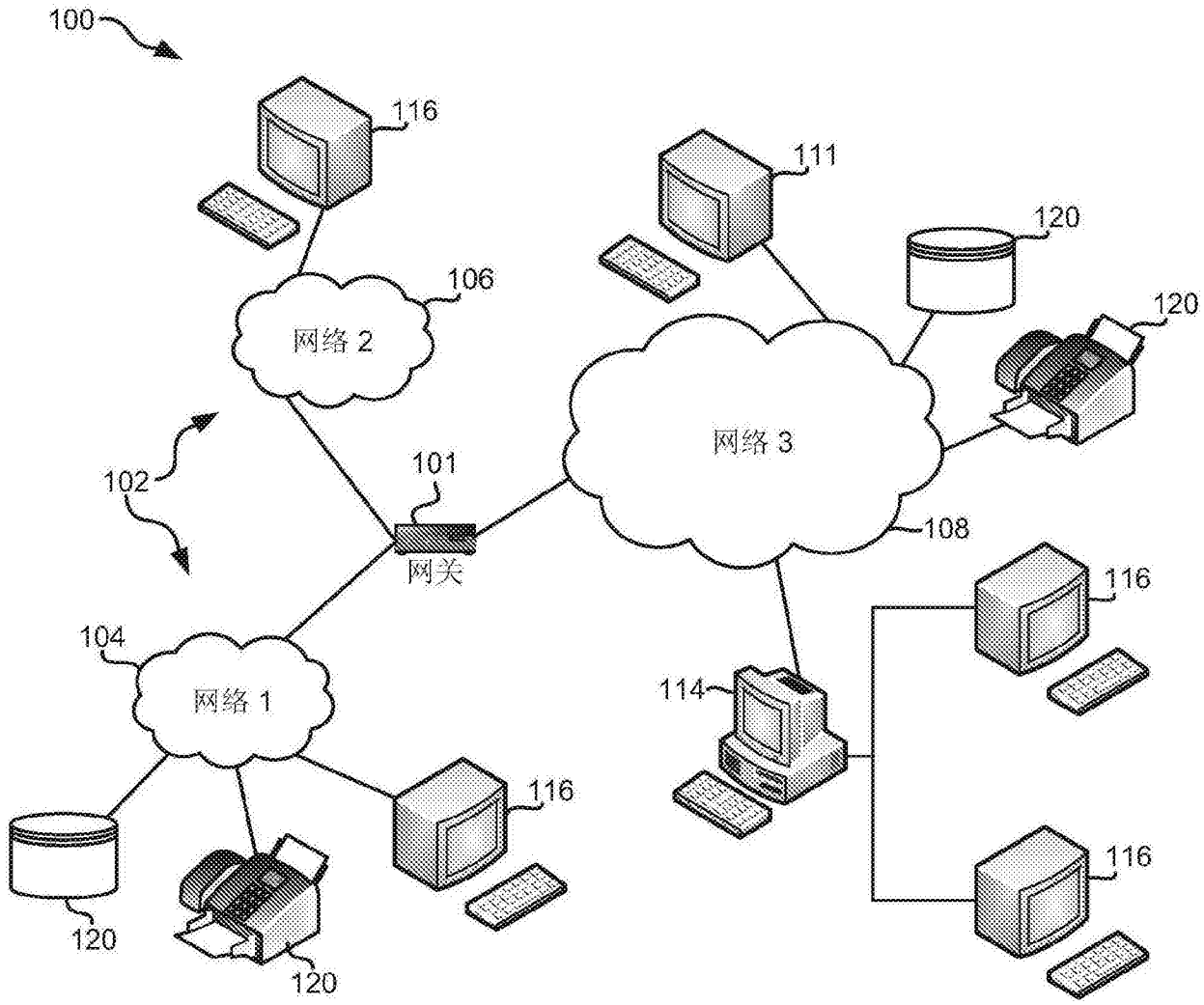

100
116
111
120
106
网络 2
120
102
101
网络 3
网关
108
104
116
网络 1
114
116
116
120
120

图1

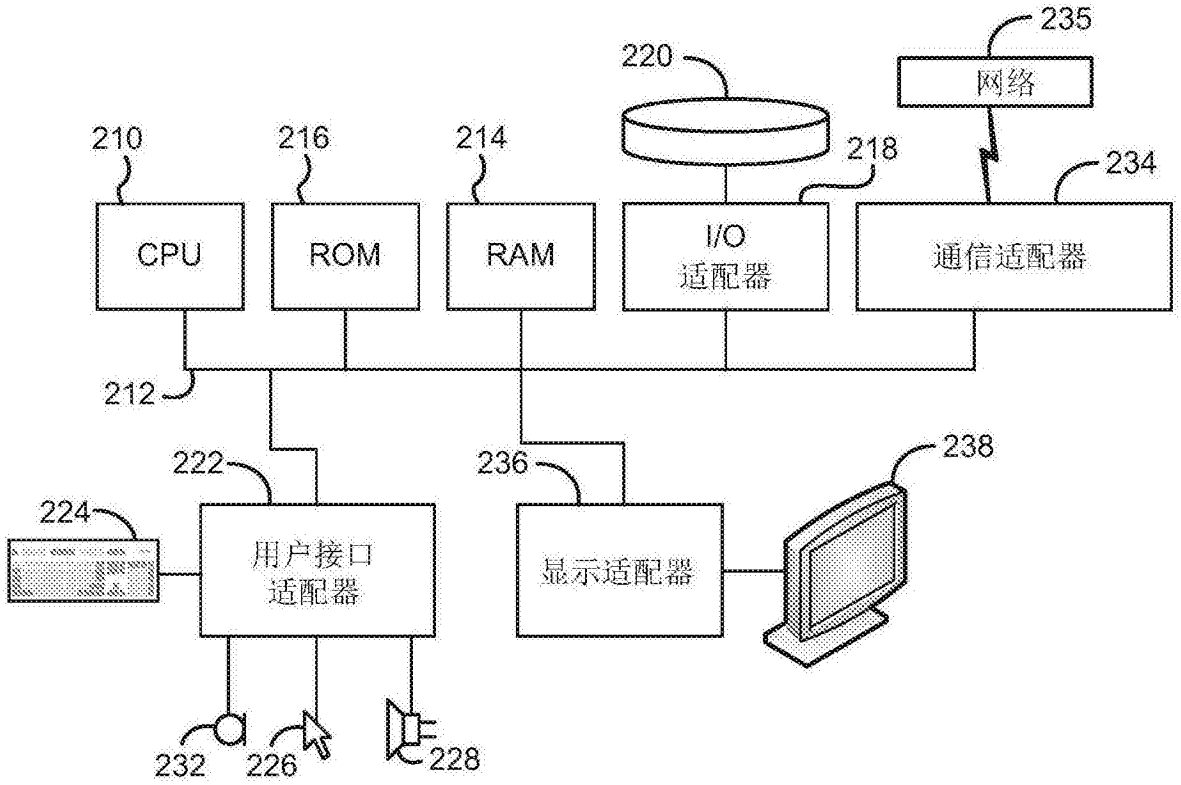

235
网络
220
210
216
214
218
234
CPU
ROM
RAM
I/O
适配器
通信适配器
212
222
236
238
224
用户接口
适配器
显示适配器
232
226
228

图2

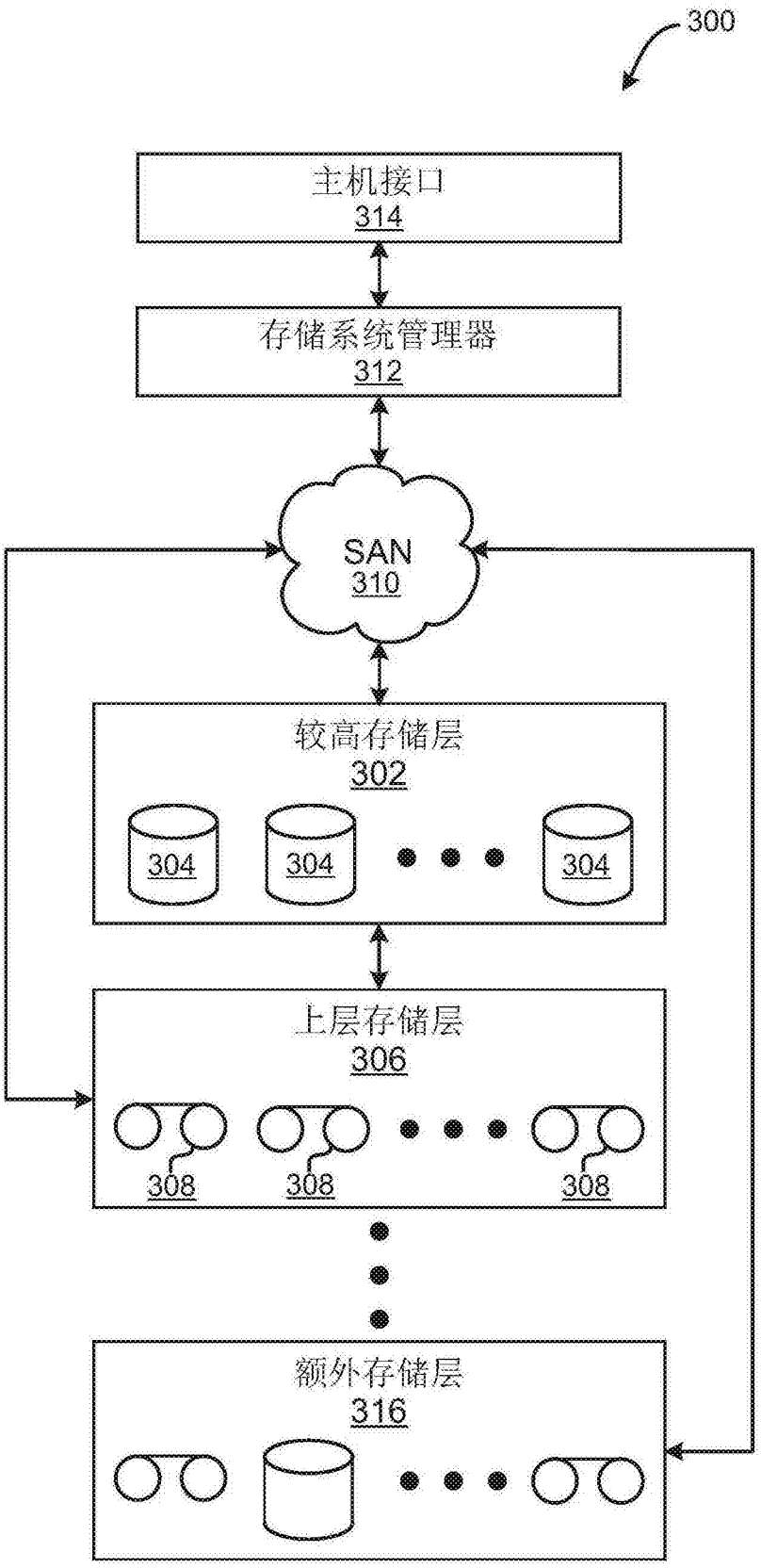

300
主机接口
314
存储系统管理器
312
SAN
310
较高存储层
302
304
304
304
上层存储层
306
308
308
308
额外存储层
316

图3

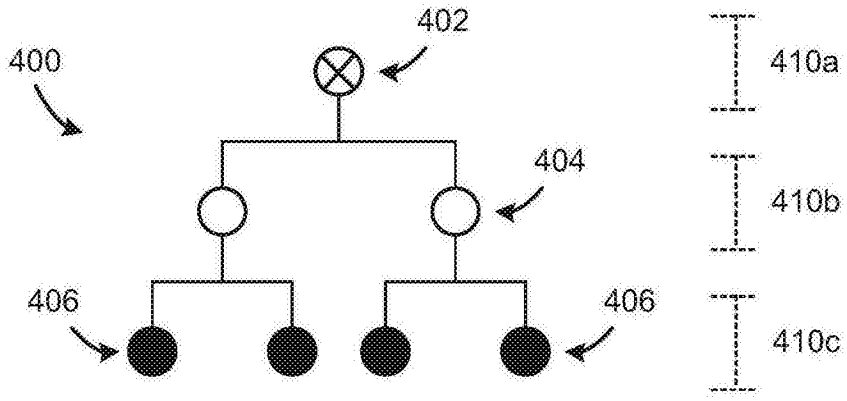

400
402
404
406
406
410a
410b
410c

图4

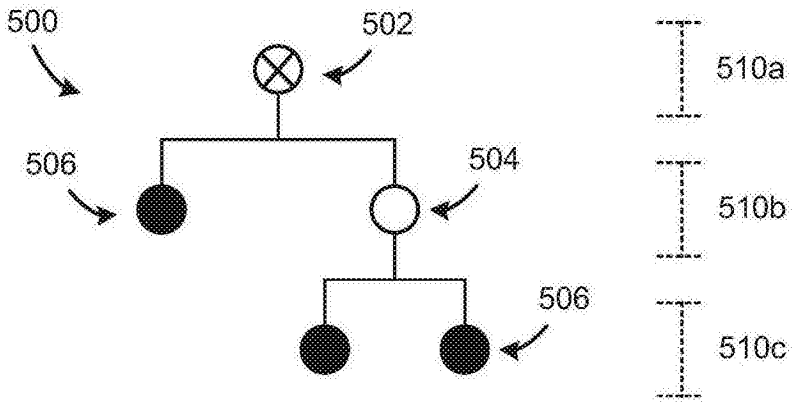

500
502
506
504
506
510a
510b
510c

图5

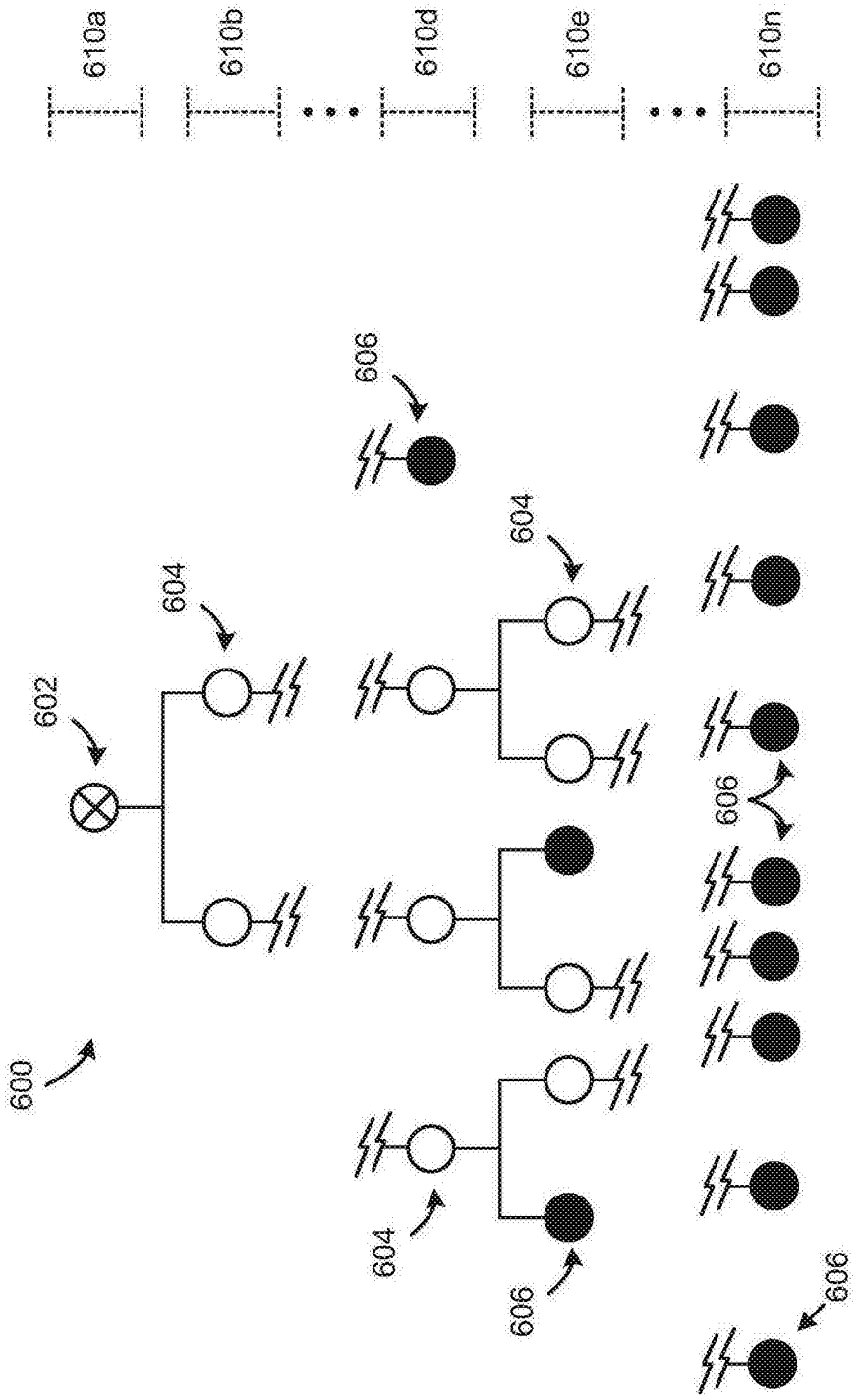


图6

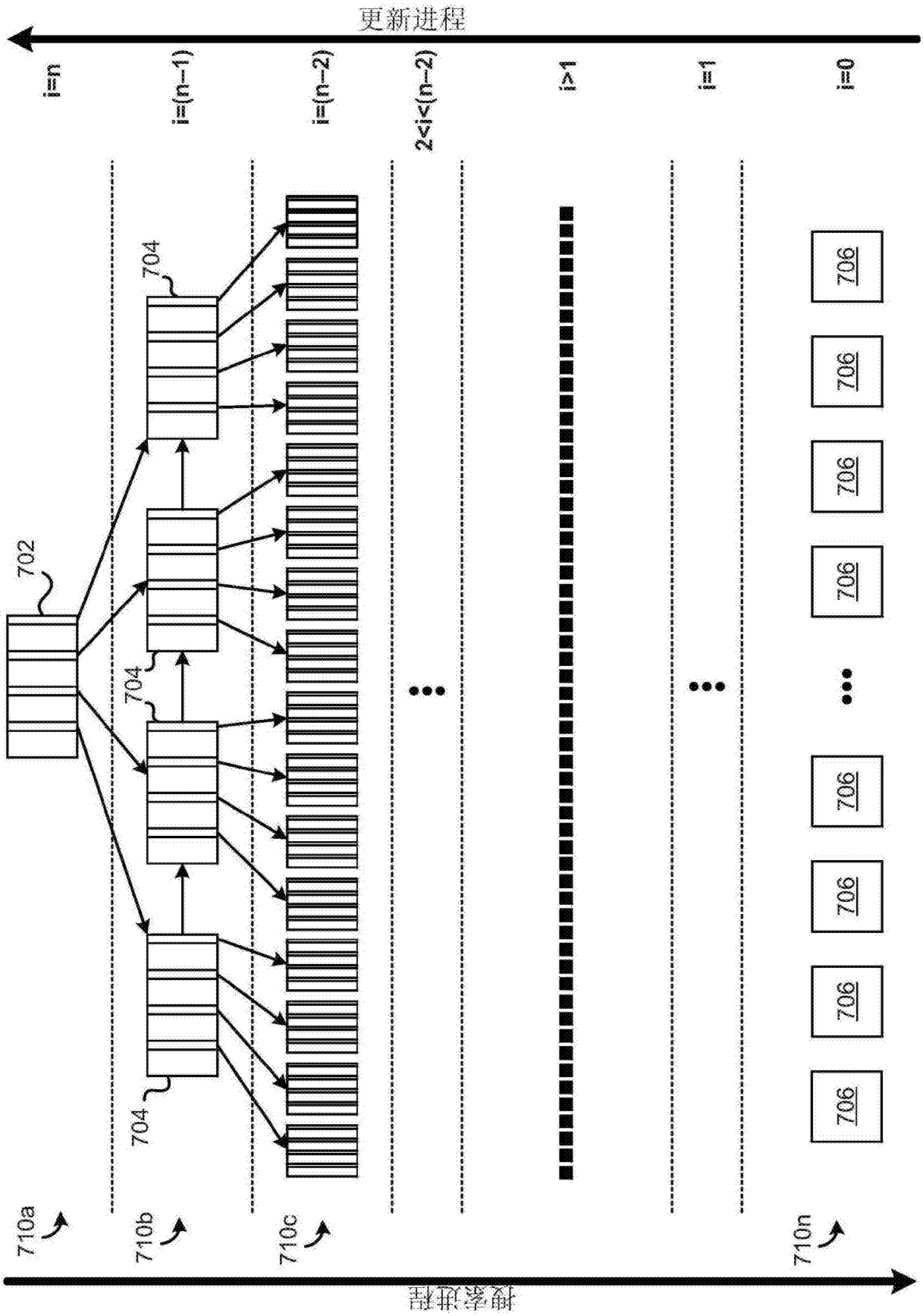

更新进程
i=n
i=(n-1)
i=(n-2)
2<i<(n-2)
i>1
i=1
i=0
702
704
704
704
706
706
706
706
706
706
706
710a
710b
710c
710n
搜索进程

图7

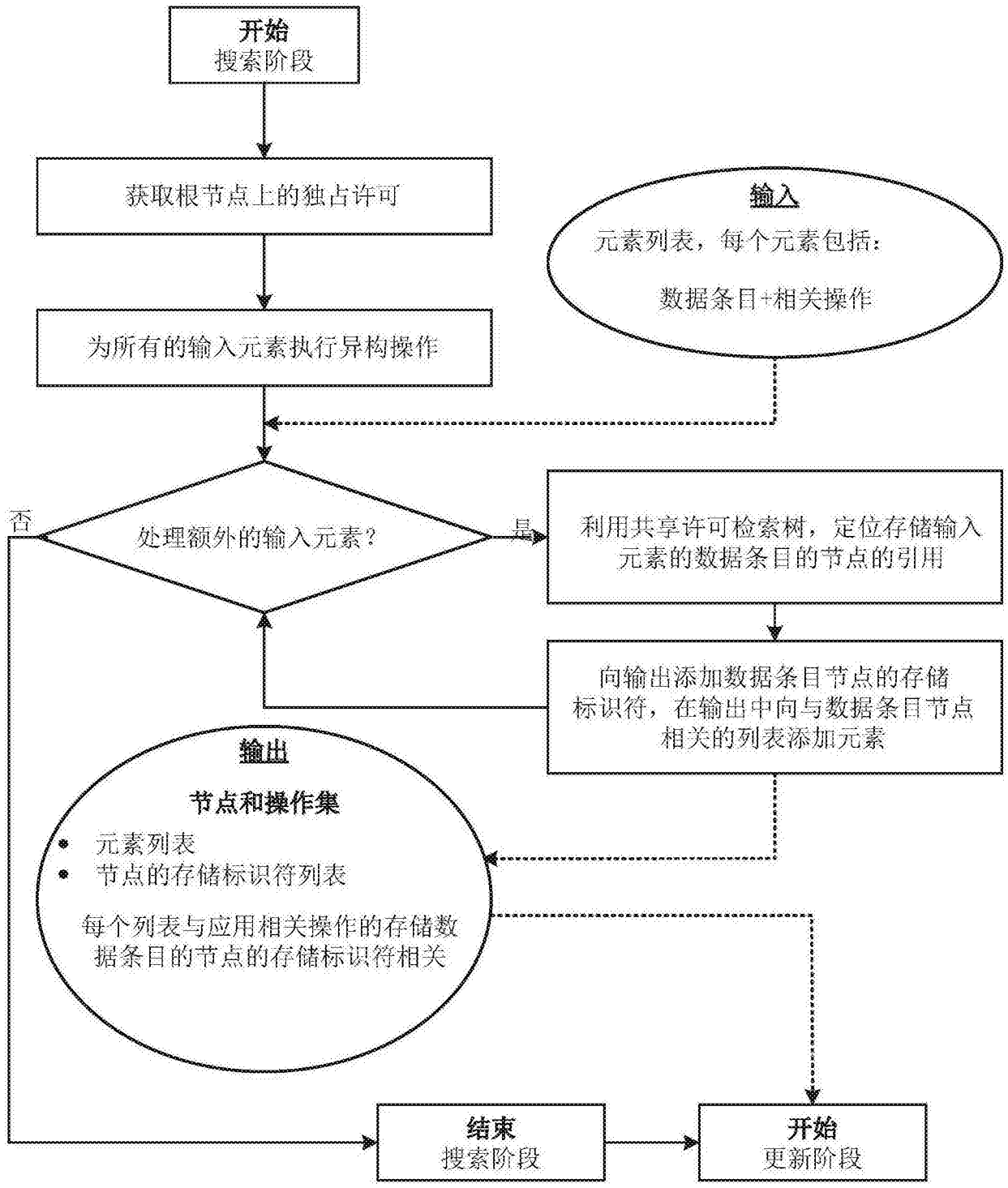

开始
搜索阶段
获取根节点上的独占许可
为所有的输入元素执行异构操作
输入
元素列表，每个元素包括：
数据条目+相关操作
处理额外的输入元素？
否
是
利用共享许可检索树，定位存储输入元素的数据条目的节点的引用
向输出添加数据条目节点的存储标识符，在输出中向与数据条目节点相关的列表添加元素
输出
节点和操作集
元素列表
节点的存储标识符列表
每个列表与应用相关操作的存储数据条目的节点的存储标识符相关
结束
搜索阶段
开始
更新阶段

图8

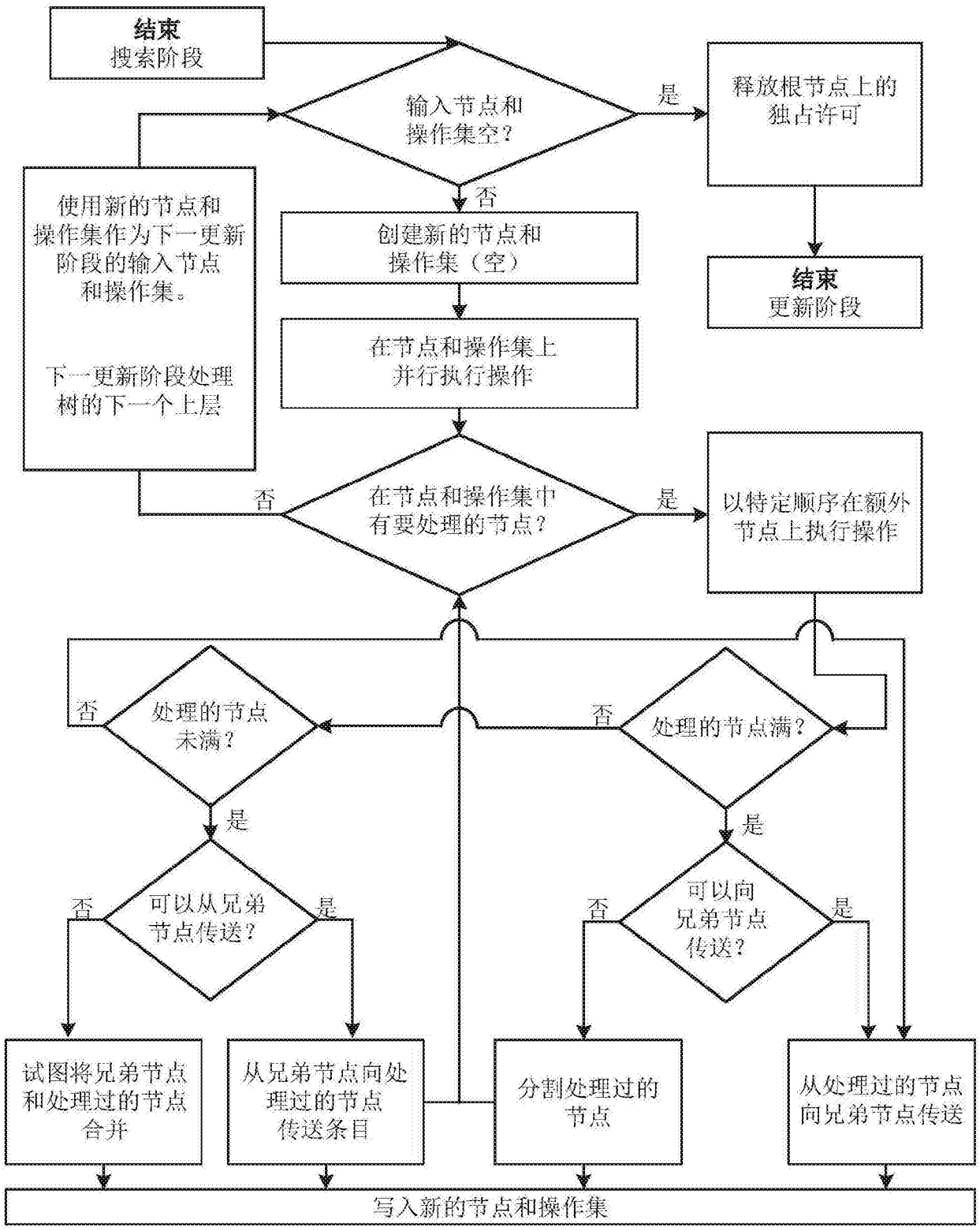

结束
搜索阶段
输入节点和
操作集空？
是
释放根节点上的
独占许可
结束
更新阶段
否
创建新的节点和
操作集（空）
在节点和操作集上
并行执行操作
使用新的节点和
操作集作为下一更新
阶段的输入节点
和操作集。
下一更新阶段处理
树的下一个上层
否
在节点和操作集中
有要处理的节点？
是
以特定顺序在额外
节点上执行操作
否
处理的节点
未满？
否
处理的节点满？
是
是
否
可以从兄弟
节点传送？
是
否
可以向
兄弟节点
传送？
是
试图将兄弟节点
和处理过的节点
合并
从兄弟节点向处
理过的节点
传送条目
分割处理过的
节点
从处理过的节点
向兄弟节点传送
写入新的节点和操作集

图9

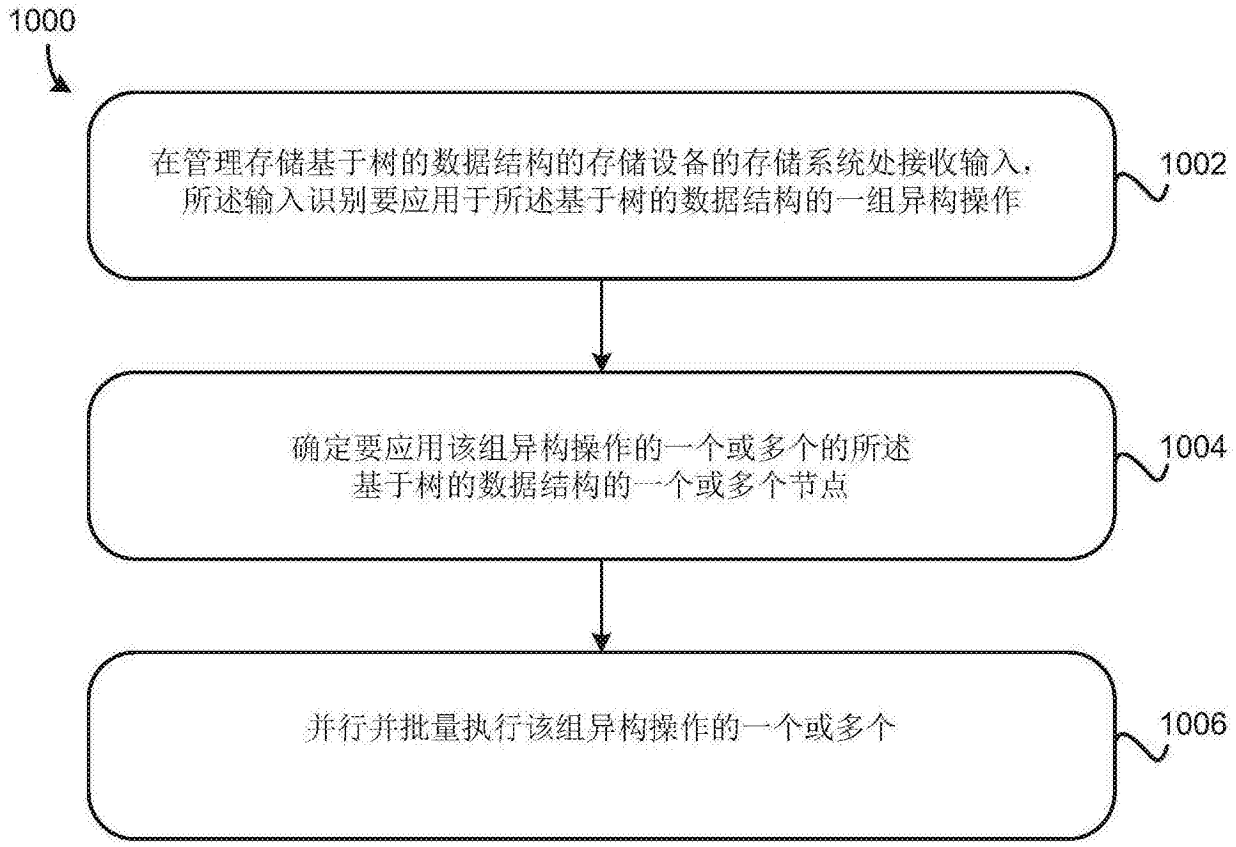

1000
在管理存储基于树的数据结构的存储设备的存储系统处接收输入，所述输入识别要应用于所述基于树的数据结构的一组异构操作
1002
确定要应用该组异构操作的一个或多个的所述基于树的数据结构的一个或多个节点
1004
并行并批量执行该组异构操作的一个或多个
1006

图10

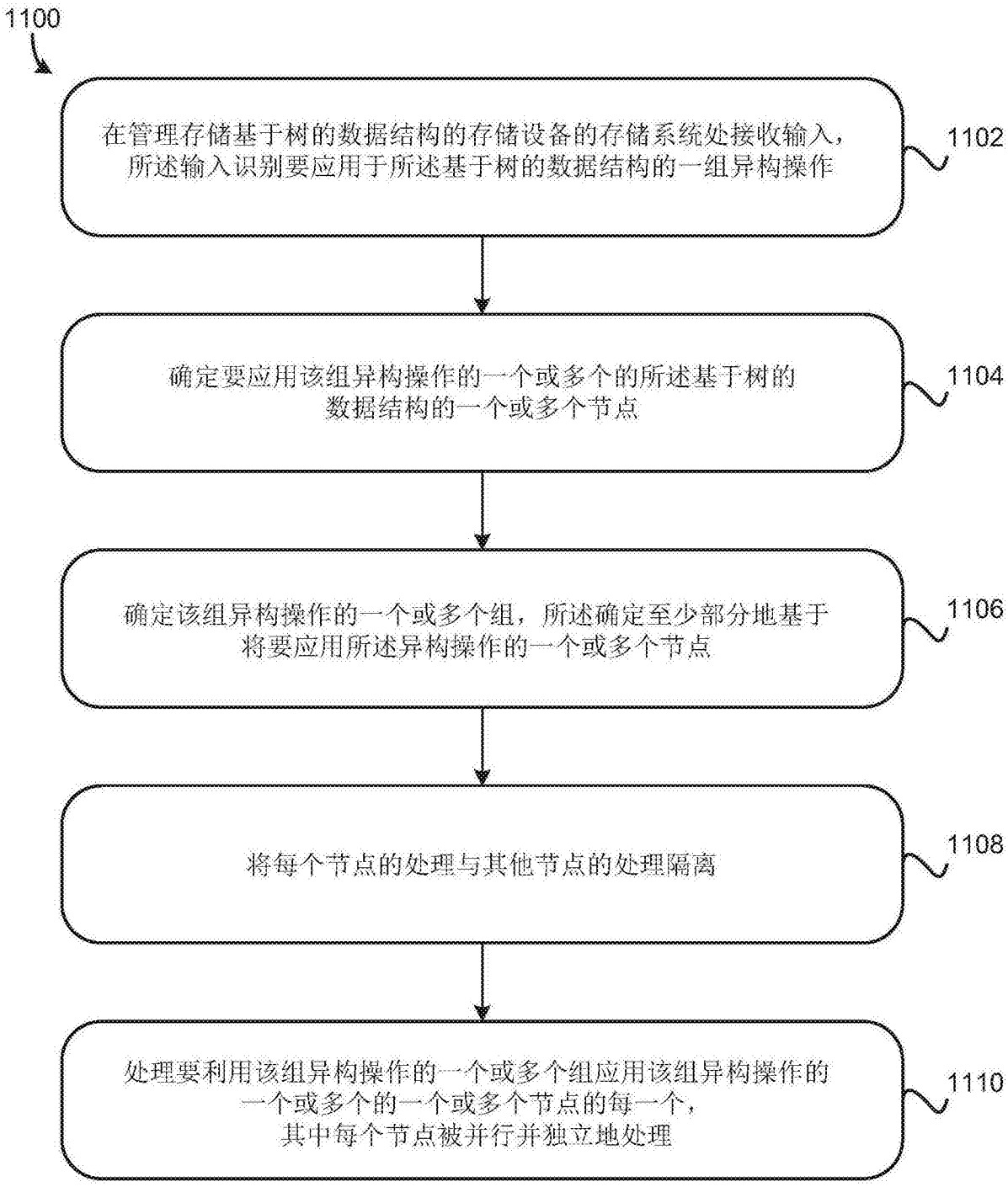

1100
在管理存储基于树的数据结构的存储设备的存储系统处接收输入，所述输入识别要应用于所述基于树的数据结构的一组异构操作
1102
确定要应用该组异构操作的一个或多个的所述基于树的数据结构的一个或多个节点
1104
确定该组异构操作的一个或多个组，所述确定至少部分地基于将要应用所述异构操作的一个或多个节点
1106
将每个节点的处理与其他节点的处理隔离
1108
处理要利用该组异构操作的一个或多个组应用该组异构操作的一个或多个的一个或多个节点的每一个，其中每个节点被并行并独立地处理
1110

图11

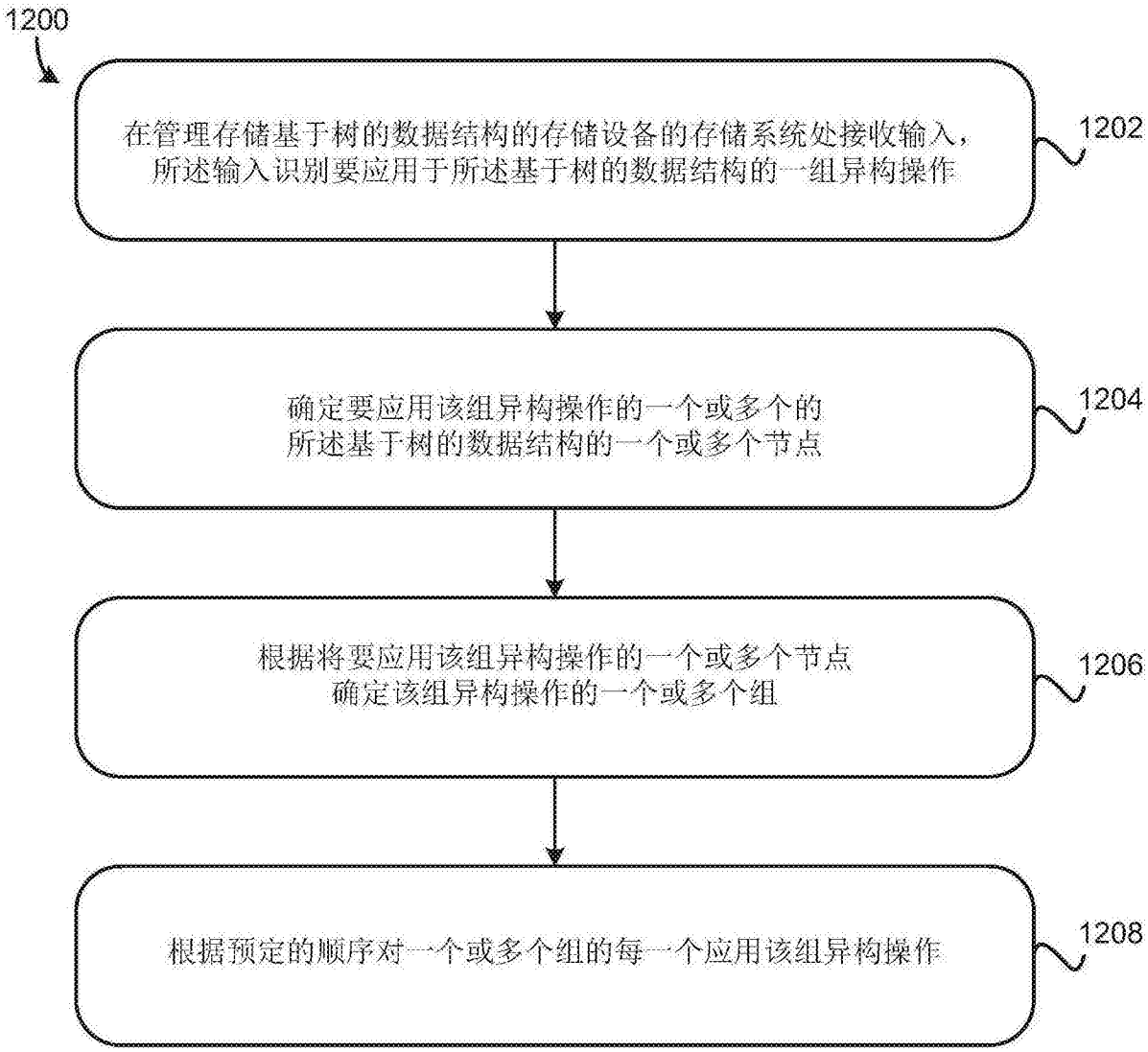

1200
在管理存储基于树的数据结构的存储设备的存储系统处接收输入，
所述输入识别要应用于所述基于树的数据结构的一组异构操作
1202
确定要应用该组异构操作的一个或多个的
所述基于树的数据结构的一个或多个节点
1204
根据将要应用该组异构操作的一个或多个节点
确定该组异构操作的一个或多个组
1206
根据预定的顺序对一个或多个组的每一个应用该组异构操作
1208

图12