



(12) 发明专利申请

(10) 申请公布号 CN 112069192 A

(43) 申请公布日 2020.12.11

(21) 申请号 202010526412.4

(22) 申请日 2020.06.10

(30) 优先权数据

16/437,852 2019.06.11 US

(71) 申请人 SAP欧洲公司

地址 德国瓦尔多夫

(72) 发明人 C.本斯伯格 S.博伊尔勒

(74) 专利代理机构 北京市柳沈律师事务所

11105

代理人 邵亚丽

(51) Int.Cl.

G06F 16/23 (2019.01)

G06F 16/27 (2019.01)

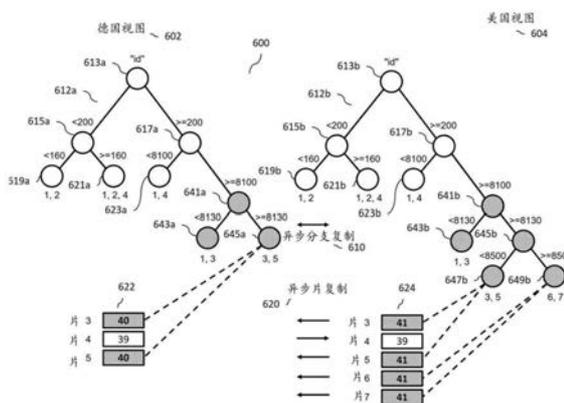
权利要求书3页 说明书13页 附图12页

(54) 发明名称

具有所有权转移的多主

(57) 摘要

一种用于执行数据所有权管理的方法、系统和计算机程序产品。选择多个索引中的索引。该索引对应于存储在数据库的多个数据库片中的多个数据值范围。该索引还对应于包括多个分层排列的节点的划分结构。每个节点对应于存储在至少一个数据库片中的数据值范围。该结构跨多个计算系统被复制。计算系统执行对一个或多个数据值范围的更新。该系统将包括更新的数据值范围的数据库片和包括更新的数据值范围的节点中的至少一个复制到另一计算系统,用于存储更新的数据值范围的副本。



1. 一种计算机实现的方法,包括:

在多个索引中选择索引,所述索引对应于存储在数据库的多个数据库片中的多个数据值范围,所述索引对应于包括多个分层排列的节点的划分结构,每个节点对应于存储在至少一个数据库片中的多个数据值范围中的数据值范围,所述划分结构跨多个计算系统被复制;

由所述多个计算系统中的计算系统执行对一个或多个数据值范围的更新;以及

由所述计算系统将包括更新的一个或多个数据值范围的数据库片和包括更新的一个或多个数据值范围的节点中的至少一个复制到所述多个计算系统中的另一计算系统,用于存储更新的一个或多个数据值范围的副本。

2. 根据权利要求1所述的方法,其中,所述复制包括基于对所述一个或多个数据值范围的更新的数量大于预定的更新阈值数量,将包括更新的一个或多个数据值范围的节点复制到所述另一计算系统。

3. 根据权利要求1所述的方法,其中,所述执行包括生成被配置为存储更新的一个或多个数据值范围的另一数据库片。

4. 根据权利要求1所述的方法,其中,所述多个片中的一个或多个片被配置为由所述多个计算系统中的一个或多个计算系统独立于所述多个分层排列的节点中的一个或多个节点来拥有,其中,一个或多个片的所有权被配置为由一个或多个计算系统独立于所述多个分层排列的节点中的一个或多个节点的所有权来转移。

5. 根据权利要求3所述的方法,还包括由所述计算系统将生成的另一数据库片复制到所述另一数据库系统。

6. 根据权利要求5所述的方法,还包括:

由所述另一计算系统存储生成的另一数据库片;

由所述另一计算系统执行对所述另一数据库片中的一个或多个数据值范围的更新;以及

由所述另一计算系统将所述另一数据库片复制到所述计算系统,并且由所述计算系统将更新的一个或多个数据值范围的副本存储在所述另一数据库片中。

7. 根据权利要求1所述的方法,其中,所述复制包括异步复制。

8. 根据权利要求1所述的方法,其中,所述执行包括以下中的至少一个:将对一个或多个数据值范围的更新插入到一个或多个数据库片中,使用对一个或多个数据值范围的更新来修改存储在一个或多个数据库片中的数据,删除一个或多个数据库片中的数据,以及它们的任意组合。

9. 一种系统,包括:

至少一个可编程处理器;和

存储指令的非暂时性机器可读介质,当所述指令被所述至少一个可编程处理器执行时,使所述至少一个可编程处理器执行包括以下步骤的操作:

在多个索引中选择索引,所述索引对应于存储在数据库的多个数据库片中的多个数据值范围,所述索引对应于包括多个分层排列的节点的划分结构,每个节点对应于存储在至少一个数据库片中的多个数据值范围中的数据值范围,所述划分结构跨多个计算系统被复制;

由所述多个计算系统中的计算系统执行对一个或多个数据值范围的更新;以及

由所述计算系统将包括更新的一个或多个数据值范围的数据库片和包括更新的一个或多个数据值范围的节点中的至少一个复制到所述多个计算系统中的另一计算系统,用于存储更新的一个或多个数据值范围的副本。

10. 根据权利要求9所述的系统,其中,所述复制包括基于对所述一个或多个数据值范围的更新的数量大于预定的更新阈值数量,将包括更新的一个或多个数据值范围的节点复制到所述另一计算系统。

11. 根据权利要求9所述的系统,其中,所述执行包括生成被配置为存储更新的一个或多个数据值范围的另一数据库片。

12. 根据权利要求9所述的系统,其中,所述多个片中的一个或多个片被配置为由所述多个计算系统中的一个或多个计算系统独立于所述多个分层排列的节点中的一个或多个节点来拥有,其中,一个或多个片的所有权被配置为由一个或多个计算系统独立于所述多个分层排列的节点中的一个或多个节点的所有权来转移。

13. 根据权利要求11所述的系统,其中,所述操作还包括由所述计算系统将生成的另一数据库片复制到所述另一数据库系统。

14. 根据权利要求13所述的系统,其中,所述操作还包括:

由所述另一计算系统存储生成的另一数据库片;

由所述另一计算系统执行对所述另一数据库片中的一个或多个数据值范围的更新;以及

由所述另一计算系统将所述另一数据库片复制到所述计算系统,并且由所述计算系统将更新的一个或多个数据值范围的副本存储在所述另一数据库片中。

15. 根据权利要求9所述的系统,其中,所述复制包括异步复制。

16. 根据权利要求9所述的系统,其中,所述执行包括以下中的至少一个: 将对一个或多个数据值范围的更新插入到一个或多个数据库片中,使用对一个或多个数据值范围的更新来修改存储在一个或多个数据库片中的数据,删除一个或多个数据库片中的数据,以及它们的任意组合。

17. 一种计算机程序产品,包括存储指令的非暂时性机器可读介质,当所述指令被至少一个可编程处理器执行时,使所述至少一个可编程处理器执行包括以下步骤的操作:

在多个索引中选择索引,所述索引对应于存储在数据库的多个数据库片中的多个数据值范围,所述索引对应于包括多个分层排列的节点的划分结构,每个节点对应于存储在至少一个数据库片中的多个数据值范围中的数据值范围,所述划分结构跨多个计算系统被复制;

由所述多个计算系统中的计算系统执行对一个或多个数据值范围的更新;以及

由所述计算系统将包括更新的一个或多个数据值范围的数据库片和包括更新的一个或多个数据值范围的节点中的至少一个复制到所述多个计算系统中的另一计算系统,用于存储更新的一个或多个数据值范围的副本。

18. 根据权利要求17所述的计算机程序产品,其中,所述复制包括基于对所述一个或多个数据值范围的更新的数量大于预定的更新阈值数量,将包括更新的一个或多个数据值范围的节点复制到所述另一计算系统。

19. 根据权利要求17所述的计算机程序产品,其中,所述执行包括生成被配置为存储更新的一个或多个数据值范围的另一数据库片。

20. 根据权利要求17所述的计算机程序产品,其中,所述多个片中的一个或多个片被配置为由所述多个计算系统中的一个或多个计算系统独立于所述多个分层排列的节点中的一个或多个节点来拥有,其中,一个或多个片的所有权被配置为由一个或多个计算系统独立于所述多个分层排列的节点中的一个或多个节点的所有权来转移。

具有所有权转移的多主

技术领域

[0001] 本公开一般涉及数据处理,尤其涉及多主(multi-master)数据管理,包括一个或多个数据划分的所有权的转移。

背景技术

[0002] 数据库管理系统已经成为许多计算机系统的组成部分。例如,一些系统每秒处理数百甚至数千个事务。另一方面,一些系统对数据执行非常复杂的多维分析。在这两种情况下,底层数据库可能需要非常快速地处理对查询的响应,以满足系统对事务时间的要求。由这种系统存储的数据可以存储在一个或多个划分中。考虑到查询的复杂性、存储的数据的量和/或它们容量,底层数据库面临着优化性能的挑战。

发明内容

[0003] 在一些实现中,当前主题涉及用于执行数据所有权管理的计算机实现的方法。该方法可以包括在多个索引中选择索引。该索引可以对应于存储在数据库的多个数据库片中的多个数据值范围。该索引还可以对应于包括多个分层排列的节点的划分结构。每个节点可以对应于存储在至少一个数据库片中的多个数据值范围中的数据值范围。划分结构可以跨多个计算系统被复制。该方法还可以包括由多个计算系统中的计算系统执行对一个或多个数据值范围的更新,以及由计算系统将包括更新的一个或多个数据值范围的数据库片和包括更新的一个或多个数据值范围的节点中的至少一个复制到多个计算系统中的另一计算系统,用于存储更新的一个或多个数据值范围的副本。

[0004] 在一些实现中,当前主题可以包括以下可选特征中的一个或多个。在一些实现中,复制可以包括基于对一个或多个数据值范围的更新的数量大于预定的更新阈值数量,将包括更新的一个或多个数据值范围的节点复制到另一计算系统。

[0005] 在一些实现中,更新执行可以包括生成被配置为存储更新的一个或多个数据值范围的另一数据库片。该方法还可以包括由计算系统将生成的另一数据库片复制到另一数据库系统。该方法还可以包括由另一计算系统存储生成的另一数据库片,由另一计算系统执行对另一数据库片中的一个或多个数据值范围的更新,以及由另一计算系统将另一数据库片复制到计算系统,以及由计算系统将更新的一个或多个数据值范围的副本存储在另一数据库片中。

[0006] 在一些实现中,多个片中的一个或多个片可以被配置为独立于多个分层排列的节点中的一个或多个节点而被多个计算系统中的一个或多个计算系统所拥有。一个或多个片的所有权可以被配置为由一个或多个计算系统独立于一个或多个节点的所有权来转移。

[0007] 在一些实现中,复制可以包括异步复制。

[0008] 在一些实现中,更新的执行可以包括以下至少一个:将对一个或多个数据值范围的更新插入到一个或多个数据库片中,使用该对一个或多个数据值范围的更新来修改存储在一个或多个数据库片中的数据,删除一个或多个数据库片中的数据,以及它们的任意组

合。

[0009] 还描述了存储指令的非暂时性计算机程序产品(即,物理体现的计算机程序产品),当由一个或多个计算系统的一个或多个数据处理器执行时,该指令使得至少一个数据处理器执行本文的操作。类似地,还描述了可以包括一个或多个数据处理器和耦合到一个或多个数据处理器的存储器的计算机系统。存储器可以临时或永久地存储指令,这些指令使得至少一个处理器执行本文描述的一个或多个操作。此外,方法可以由单个计算系统内或分布在两个或多个计算系统中的一个或多个数据处理器来实现。这种计算系统可以由一个或多个连接(包括但不限于通过网络(例如,因特网、无线广域网、局域网、广域网、有线网络等)的连接,经由多个计算系统中的一个或多个之间的直接连接等)来连接并且可以交换数据和/或命令或其他指令等。

[0010] 本文描述的主题的一个或多个变化的细节在附图和下面的描述中阐述。从说明书和附图以及权利要求书中,本文描述的主题的其他特征和优点将是显而易见的。

附图说明

[0011] 并入并构成本说明书一部分的附图显示了本文公开的主题的某些方面,并且与描述一起,有助于解释与所公开的实现相关联的一些原理。在附图中,

[0012] 图1示出了根据当前主题的一些实现的用于多主数据管理的示例性系统;

[0013] 图2示出了根据当前主题的一些实现的示例性多主数据管理系统;

[0014] 图3示出了根据当前主题的一些实现的另一示例性多主数据管理系统;

[0015] 图4示出了根据当前主题的一些实现的示例性划分规范;

[0016] 图5示出了根据当前主题的一些实现的另一示例性多主数据管理系统;

[0017] 图6a-6c示出了根据当前主题的一些实现的示例性多主数据管理系统;

[0018] 图7是示出根据当前主题的一些实现的包括数据存储应用的示例性系统的图;

[0019] 图8是示出图7的系统的细节的图;

[0020] 图9是根据当前主题的一些实现的示例性系统;和

[0021] 图10是根据当前主题的一些实现的示例性方法。

具体实施方式

[0022] 为了解决当前可用解决方案的这些和潜在的其他缺陷,当前主题的一个或多个实现涉及方法、系统、制品等,除了其他可能的优点之外,这些方法、系统、制品等可以提供多主数据管理,包括一个或多个数据划分的所有权的转移。

[0023] 数据库管理系统和对数据库管理系统管理的数据执行的操作变得越来越复杂。例如,数据库管理系统(或简称为数据库)可以支持相对复杂的在线分析处理(Online analytical processing,OLAP,它可以执行多维分析)到更直接的基于事务的在线事务处理(online transaction processing,OLTP)。此外,数据库可以被配置为行存储数据库或列存储数据库,其中的每一个都可以具有关于数据库处的查询和其他操作的某些方面。例如,数据库可能使用字典对数据进行编码,而有些数据库可能不使用。除了这些各种数据库层差异之外,在数据库处执行的查询可能包括复杂的操作序列,以便生成相应的响应。为了实现复杂的序列,可以实现查询执行计划(或简称为查询计划)。查询计划可以表示访问数

数据库中的数据的一系列操作,诸如指令、命令等。数据库还可以包括查询计划优化器,以确定执行查询计划的有效方式。

[0024] 从应用或客户端的角度来看,访问数据库可能非常麻烦。例如,应用可能需要使用复杂的查询来查询不同类型的数据库。因此,应用层可能需要被配置为处理各种类型的数据库和各种查询类型。附加地或可替换地,每个数据库可能需要将来自应用的查询处理成给定数据库可以处理的格式和结构。将复杂的操作和对各种不同数据库类型的支持推送到应用层可能会违背具有相对较轻的重量和/或易于部署的应用的需求。另一方面,将复杂的操作推送到存储数据的数据库层可能会在数据库处消耗处理和/或存储器资源,因此可能会降低该数据库层上查询的性能和响应时间。

[0025] 数据库系统可以使用一个或多个划分配置来存储数据。数据库中的划分可以指将逻辑数据库或其元素划分成独立的部分。划分允许改进的可管理性、性能、负载平衡等。在一些情况下,划分可以分布在多个节点上,其中每个节点可以允许用户在划分上执行各种操作(例如,事务的执行等)。这种分布可以提高节点的性能,这些节点可能经历频繁的事务,这些事务可能涉及数据的检索、插入、修改、视图生成等,同时保持数据的可用性和安全性。数据划分可以通过建立单独的较小数据库、分割选择的元素等来执行。可以使用水平或垂直划分方法对数据进行划分。水平划分可以将不同的行放入不同的表中(例如,分割不同年龄组的用户)。垂直划分可以创建具有较少列的新表,并且可以使用附加的表来存储任何剩余的列。

[0026] 在一些实现中,当前主题可以被配置成为可以存储在数据库系统中的数据生成划分规范。可以使用包括在数据库和/或其划分中的数据的片来“即时”(on the fly)定义划分规范(可以假设存储在数据库中的数据是隐式划分的)。当前主题可以使用树结构生成划分规范,其中树中的节点可以对应于数据库中的特定数据片。数据片可能相对较小。这样,如果划分方案不理想,在性能方面就没有大的缺点,因为小片的处理可能相对较快。这种方法的优点之一是不需要昂贵的数据重新组织。

[0027] 在一些实现中,(多个)数据库的一个或多个划分可以位于或存储在一个或多个服务器上,这些服务器可以位于不同的位置(例如,在不同的地理区域)。服务器可以“拥有”特定的数据划分,诸如,例如通过在该服务器上创建和存储划分,更频繁地在该服务器上访问(例如,更频繁地访问特定划分的用户在地理上位于该服务器附近)。然而,在一些情况下,划分的所有权可能需要(例如,临时地、永久地等)从一个服务器转移到另一服务器。这可能是为了提供更新、执行他服务器本地的事务、执行该服务器本地的写入等目的。在一些示例性实现中,一旦另一服务器完成任务,它可以将划分的所有权转移回原始服务器。在一些实现中,可以转移整个划分的所有权。或者,可以仅转移划分的一部分(例如,树的“分支”,如下面将要讨论的)。

[0028] 图1示出了根据当前主题的一些实现的用于多主数据管理的示例性系统100。系统100可以包括一个或多个用户(用户1、用户2、...用户n) 102、执行引擎104和数据库系统106,数据库系统106可以将数据存储在一个或多个片108中。用户102、执行引擎104和数据库系统106可以使用任何类型的网络彼此通信耦合,包括但不限于有线、无线和/或两者的组合。用户102可以包括以下中的至少一个:计算机处理器、计算网络、软件应用、服务器、用户界面和/或硬件和/或软件组件的任意组合。数据库系统106可以包括以下至少一个:数据库、

存储位置、存储器位置、列存储、文档存储和/或硬件和/或软件组件的任意组合。在一些实现中,数据库系统106可以是由德国沃尔多夫(Walldorf,Germany)的SAP SE开发的高性能分析设备(“High Performance Analytic Appliance,HANA”)系统,如下所述。

[0029] 执行引擎104可以包括软件和/或硬件组件的任意组合,并且可以被配置为接收和执行来自一个或多个用户102的查询,以从数据库系统106中的一个或多个片108获得数据,将数据插入数据库系统106中的一个或多个片108,修改从数据库系统106中的一个或多个片108存储的数据,删除从数据库系统106中的一个或多个片108存储的数据,生成一个或多个新片108(例如,用于插入新数据)等,及其任意组合。在一些实现中,执行引擎106可以被包括在数据库系统106中。

[0030] 查询的执行通常可能需要生成查询计划或查询执行计划,查询计划或查询执行计划可以是可用于访问存储的数据(例如,访问SQL关系数据库管理系统中的数据)的有序操作的集合。在向数据库系统106提交查询后,可以基于查询的参数检索所请求的数据。检索的数据可以与用户可能寻找的任何其他数据聚集/结合。数据库系统106中数据的插入、修改、删除等可以使用各种SQL或其他语句来执行。

[0031] 图2示出了根据当前主题的一些实现的示例性多主数据管理系统200。系统200可以包括经由联网连接206通信耦合的第一计算系统202和第二计算系统204。系统202、204可以包括以下中的至少一个:一个或多个服务器、一个或多个计算用户、一个或多个数据库、一个或多个存储器位置、计算机处理器、计算网络、软件应用、服务器、用户界面、和/或硬件和/或软件组件的任意组合、和/或任何其他计算组件。通信链路206可以包括任何类型的网络,包括但不限于有线、无线和/或两者的组合。可以布置在系统202、204处的数据库可以包括以下中的至少一个:任何存储位置、任何存储器位置、任何列存储、任何文档存储和/或硬件和/或软件组件的任何组合。

[0032] 系统202、204可以被配置为对可能存储在任一系统中的数据执行本地事务。这可能包括此类数据的任何更新、删除、修改、写入等。每个系统可以(临时地、或永久地等)拥有由系统的用户正在访问的任何数据。在一些实现中,数据的所有权可以从一个系统转移到另一系统。可以为数据的特定划分(例如,片)、特定划分的一部分(例如,划分规范中的树的分支)转移数据的所有权。在一些实现中,一旦数据的所有权被转移,接收系统可以成为该数据的新所有者,直到它被请求或者可以被确定所有权可能需要被转移回最初拥有该数据的系统。或者,可能提供更新的系统可能永远不拥有来自另一系统的数据,而是简单地对数据提供任何必要的更新。

[0033] 图3示出了根据当前主题的一些实现的另一示例性多主数据管理系统300。系统300可以类似于图2所示的系统200。系统300可以包括通信耦合到第二系统304(例如,类似于系统204(如图2所示)并位于美国)的第一系统302(例如,类似于系统202(如图2所示)并位于德国)。每个系统可以存储一个或多个数据片(或划分)306。每个位置还可以“拥有”特定的数据片(和/或其版本)。如图3所示,未被着色的数据片可由系统302拥有,而被着色的片可由系统304拥有。

[0034] 每个数据片也可以具有与之相关联的片的适当版本。例如,由位置302拥有的片1可以具有与其相关联的“版本40”。如果系统302提供对片1的版本40的更新,则这种更新可以被传送(如箭头所示)到存储在系统304中的相应片1的版本(如存储在系统302中的片12-

14到版本40的更新所示,其被传送以更新存储在系统304中的相应片版本39)。类似地,对由系统304拥有并在系统304执行的片的任何更新可以被传送到系统302处的相应片(如对片4、7-10和15的更新所示)。如上所述,可以对特定片和/或片的部分执行更新。这种更新可以允许用户对同一片或表(其可以跨两个系统被复制)的不同部分进行操作(例如,更新、写入等),由此两个独立的系统302、304可以在逻辑上表现为单个系统。

[0035] 图4示出了与片1402-片5410一起生成的示例性划分规范412和414(例如,以树结构的形式)。如图4所示,片402-405可以是包括索引的表。例如,片1402可以包括标识符(identifier)列以及最小值(“min”)和最大值(“max”)列。标识符列可以包括“id”、“城市(city)”和“订单_日期(order_date)”。这些标识符中的每一个都包括值的范围,例如,“id”标识符可以包括“100-8100”的范围,“城市”标识符可以包括“Heidelberg-New York”的范围,“订单_日期”标识符可以包括“2018-04-01-2018-04-02”的范围。

[0036] 类似地,片2404可以包括其标识符的不同范围。例如,“id”标识符范围可以是“110-200”;“城市”标识符可以包括“Aachen-Wuerzburg”的范围,“订单_日期”标识符可以包括“2018-04-01-2018-05-01”的范围。在片3406中,“id”标识符可以包括“8100-8200”的范围,“城市”标识符可以包括“Chicago-Washington”的范围,并且“订单_日期”标识符可以包括“2018-04-02-2018-05-01”的范围。在片4408中,“id”标识符可以包括“180-250”的范围,“城市”标识符可以包括“Bonn-Wolfsburg”的范围,“订单_日期”标识符可以包括“2018-04-15-2018-05-15”的范围。最后,在片5410中,“id”标识符可以包括“8150-8250”的范围,“城市”标识符可以包括“Denver-Washington”的范围,并且“订单_日期”标识符可以包括“2018-04-16-2018-05-16”的范围。在一些实现中,作为隐式划分的结果,数据可以以这样的方式排列,即片可以包含可以分别高度相关的数据,并且因此按照主题隐式地组织(例如,片2和4可以包括与德国订单相关的数据;片3和5可以包括与美国订单相关的数据,片1可以包括与美国和德国订单相关的所有数据)。

[0037] 在一些实现中,片可以用于生成划分规范(例如,树412和414),其中可以根据标识符、基于片中标识的范围来组织数据。划分规范可用于存储在片中的数据插入、检索、修改、删除等。此外,如果必要或需要,这些规范可用于创建新的数据片。在一些实现中,片和划分规范可以同时增长,并且可以基于可能插入的实际数据值范围而相互影响。后者可能导致片的生成,由此在每个片内,一个或多个标识符/字段(例如,“id”)之间的相关性可能是高的。

[0038] 如图4所示,树412可以基于片1-5中的标识符“id”分层地组织,其中标识符“id”是父节点413,并且可以链接到一个或多个子节点。例如,“id”父节点413可以链接到对应于“id”在小于200(“<200”)的范围内的节点415和对应于“id”在大于或等于200(“≥200”)的范围内的节点417。可以理解,任何其他数量或多个数量可以用于基于该树生成划分规范。此外,超过一个节点可以链接到父节点413。

[0039] 节点415可进一步链接到子节点419和421,其中节点419可包括对应于小于160(“<160”)的“id”标识符的数据,并且节点421可包括对应于大于或等于160(“≥160”)的“id”标识符的数据。此外,节点417可以链接到子节点423和441,其中子节点423可以包括对应于小于8100(“<8100”)的“id”标识符的数据,并且节点441可以包括对应于大于或等于8100(“≥8100”)的“id”标识符的数据。此外,节点441可以包括其自己的子节点443(值小于

8130)和445(值大于或等于8130)。

[0040] 基于这种划分,节点419可以对应于片1、2中的一些或全部数据;节点421可以对应于片1、2、4中的一些或全部数据;节点423可以对应于片1和4中的一些或全部数据;节点443可以对应于片1和3中的一些或全部数据;并且节点445可以包括片3和5中的一个或多个数据值。

[0041] 对应于树412中的节点的数据值(或数据值的范围)可以由不同的系统拥有(例如,图3中所示的系统302(德国)、304(美国))。如图4所示,对应于节点413、415、417、419、421和423的数据值可以由德国系统302拥有,而对应于节点441、443和445的数据值可以由美国系统304拥有。

[0042] 类似地,树414可以基于片1-5中的标识符“订单_日期”分层地组织,其中标识符“订单_日期”是父节点427。“订单_日期”父节点427可以链接到节点429和431,其中节点429包括对应于“订单_日期”在小于2018-04-10(“<2018-04-10”)的范围内的数据,并且节点431对应于“订单_日期”在大于或等于2018-04-10(“≥2018-04-10”)的范围内。如可以理解的,任何其他的一个或多个订单日期可以用于基于该树生成划分规范,和/或一个以上的节点可以链接到父节点427。

[0043] 节点429可以进一步链接到子节点433和435,其中节点433可以包括对应于小于2018-04-03(“<2018-04-03”)的“订单_日期”标识符的数据,并且节点435可以包括对应于大于或等于2018-04-03(“≥2018-04-03”)的“订单_日期”标识符的数据。此外,节点431可以链接到子节点437和439,其中子节点437可以包括对应于小于2018-05-01(“<2018-05-01”)的“订单_日期”标识符的数据,并且节点439可以包括对应于大于或等于2018-05-01(“≥2018-05-01”)的“订单_日期”标识符的数据。

[0044] 基于这种划分,节点433可以对应于片1、2和3中的一些或全部数据;节点435可以对应于片2和3中的一些或全部数据;节点437可以对应于片2、3、4和5中的一些或全部数据;并且节点439可以对应于片4和5中的一些或全部数据。类似于划分树412,对应于树414中的节点的数据值(或数据值的范围)可以由图3所示的不同系统拥有。例如,对应于节点427、431、437和439的数据值可由德国系统302拥有,并且对应于节点429、433和435的数据值可由美国系统304拥有。

[0045] 在一些实现中,片1-5可以接收各种更新,例如,数据插入、数据删除、数据修改等。例如,典型的数据插入可以使用INSERT语句(e.g.,INSERT{“id”:100,“城市”：“Heidelberg”,“订单_日期”：“2018-04-01”})来执行。这种插入可以基于与拥有数据的位置相对应的位置(例如,插入到德国系统302拥有的片2中)、由不拥有数据的位置(例如,可以在片的副本或片的一部分上执行更新)和/或由两个位置来执行。

[0046] 图5示出了根据当前主题的一些实现的另一示例性多主数据管理系统500。系统500可以分别类似于图2、3所示的系统200和300。系统500可以包括通信耦合到第二系统504(例如,类似于系统304(如图3所示)并位于美国)的第一系统502(例如,类似于系统302(如图3所示)并位于德国)。同样,每个系统可以拥有一个或多个数据片(或划分)。如图5所示,未被着色的数据片可由系统502拥有,而被着色的片可由系统504拥有。

[0047] 如图5所示,连接到德国系统502的用户可以请求对数据的片9的更新。可以确定片9由美国系统504拥有(如阴影框所示)。更新请求可以被发送到系统504用于实际执行。更新

可以包括生成片的新版本(例如,“版本41”)。系统500可以进一步确定系统502可以包括片9的较旧版本(例如,“版本39”),因此,系统504可以将更新的版本(即,“版本41”)复制到系统502用于存储。通过这种方式,远程系统可以对其不拥有的数据执行更新——通过让所有者执行实际操作。所有者有责任执行数据检查、处理事务性快照(transactional snapshot)等。

[0048] 在一些实现中,系统500可以使用系统的更新频率(例如,特定片或其部分被更新的频率)来确定片的特定版本的所有权是否需要被改变或保留在原始系统中。阈值更新频率可用于这种确定。作为非限制性示例,如果片的更新频率超过系统每分钟500次更新,则该片的所有权(假设它不被更新系统所拥有)可以被转移到该系统。参考图5,如果系统502每分钟更新片9的次数少于500次,则片9的所有权可以由系统504保留。否则,它可以被切换到系统502。

[0049] 在一些实现中,系统500可以确定片的所有权切换过于频繁(例如,超过另一阈值)。在这种情况下,系统500可以确定对应于特定分支中被频繁更新的节点的树划分(例如,图4所示的树412、414)的该分支的所有权可以被切换(例如,永久地、临时地等),而不是切换片所有权。下面结合图6a-6c进一步讨论分支所有权的转换。

[0050] 图6a-6c示出了根据当前主题的一些实现的示例性多主数据管理系统600。具体而言,图6a-6c示出了类似于图4所示的“id”划分规范的“id”划分规范,其基于图4所示的片1-5。如图6a所示,系统600可以包括第一系统602(即,“德国视图”)和第二系统604(即,“美国视图”),它们类似于图2-5所示的相应系统。图6a-6c示出了特定时间点的各个系统状态。由于系统之间复制的异步性质,它们在这些时间点可能具有不同的状态。这些不同的状态被图示为各自的“视图”。

[0051] 每个系统602、604可以包括“id”划分规范或树612的副本,即,系统602可以包括树612a,系统604可以包括树612b。如图6a所示,划分规范612可以由系统602拥有(即,如未阴影的圆圈所示)。每个划分规范612可以包括链接到子节点615和617的父节点613,反过来,子节点615和617又可以链接到另外的子节点。

[0052] 图6a示出了可以由系统604执行的数据的示例性更新(例如,插入)。具体地,如图6a所示,系统602的节点617a可以链接到包括链接到两个子节点643a和645a的节点641a的分支。由节点641a领导的分支可以由系统604(即,美国系统)拥有。节点643a可以包括小于8130的数据值范围,对应于片1和3,节点645a可以包括大于或等于8130的数据值范围,并且对应于片3和5。在系统602,片3和5可以对应于包含在这些片中的数据值的版本40。

[0053] 系统604可以通过更新由节点645b领导的分支中的数据值(例如,插入)来执行更新过程。具体地,两个节点647b和649b可以链接到节点645b。节点647b可对应于小于8500的数据值,并且节点649b可对应于大于或等于8500的数据值。节点647b可以对应于新的数据值范围,该数据值范围可以用于用新版本(例如,片3和5中的“版本41”)更新现有的数据值范围,如图6a所示。节点649b可以对应于不存在先前数据范围或相应片的新数据值范围。因此,可能需要生成新的片(片6和7)。片6和7可以对应于数据值范围的新版本(即,“版本41”)。

[0054] 如图6a所示,为了使用系统604中的数据执行对系统602的更新,可以执行异步分支和/或片复制过程。这可以允许两个系统在更新过程正在进行时操作。在系统602的更新

期间,新分支数据值(对应于节点647b)可以从系统604复制到系统602(如图6a中的箭头所示)。对应于节点649b的新片6和7可以异步复制到系统602。

[0055] 图6b示出了系统602执行数据更新过程的示例性过程,该过程可以发生在图6a所示的更新过程完成之后。如图6所示,在图6a中的更新过程完成之后,系统602可以包括节点645a、647a和649a,它们可以分别对应于系统604的节点(并且因此对应于数据值的范围)645b、647b和649b。

[0056] 在一些实现中,系统602可以确定已经针对片5和6(例如,“版本43”)提供了更新的数据值范围。更新的数据值范围可以异步复制到系统604(通过反转复制方向)。此外,更新可以使系统602承担片5和6的所有权,如未阴影的片所示。虽然片的所有权发生了改变,但这并不一定意味着划分规范中分支的所有权发生了改变。如图6b所示,分支641-649仍然由系统604拥有。这意味着系统602可以使用给定范围内的值(即使它们由另一系统拥有),但是它不能扩展范围、添加子节点等。

[0057] 图6c示出了在确定一个系统更新了更多片时改变分支所有权的示例性过程。当一个系统(例如,系统602)对另一系统(例如,系统604)拥有的分支(或节点)(例如,节点649)的更新可能超过预定阈值时,这可能发生,其中该阈值对应于特定分支中被更新的片的数量(例如,超过50%)。

[0058] 作为非限制性示例,系统602可以用数据值范围的更新版本(例如,对于片6为“版本43”,和对于片7为“版本44”)来执行对片6和7的更新。作为该更新的结果,片4-7现在由系统602拥有,其中更新被异步复制到系统604。因为系统602现在拥有的片的数量,节点649b的所有权可以从系统604改变到系统602。这可能是由于一个系统(例如系统602)更频繁地更新另一系统(例如系统604)存储的数据而导致的。

[0059] 在一些实现中,当前主题可以在各种内存中(in-memory)数据库系统中实现,诸如由德国沃尔多夫的SAP SE开发的高性能分析设备(“HANA”)系统。例如,各种系统,诸如企业资源规划(“enterprise resource planning,ERP”)系统、供应链管理系统(“supply chain management system,SCM”)系统、供应商关系管理(“supplier relationship management,SRM”)系统、客户关系管理(“customer relationship management,CRM”)系统和其他系统,可以为了访问数据的目的而与内存系统交互。其他系统和/或系统的组合可以用于当前主题的实现。以下是对示例性内存系统的讨论。

[0060] 图7示出了根据当前主题的一些实现的示例性系统700,其中计算系统702执行数据存储应用704的一个或多个模块、软件组件等,该计算系统702可以包括一个或多个通过一个或多个网络配置、链接的一个或多个可编程处理器。数据存储应用704可以包括数据库、企业资源程序、分布式存储系统(例如,可从加利福尼亚州森尼维尔(Sunnyvale,CA)的NetApp获得的NetApp文件管理器(NetApp Filer))等中的一个或多个。

[0061] 计算系统702的本地用户以及通过网络连接710从一个或多个客户端机器706访问计算系统702的远程用户可以访问一个或多个模块、软件组件等。由一个或多个第一模块产生的一个或多个用户界面屏幕可以经由本地显示器或经由与客户端机器706之一相关联的显示器显示给用户。数据存储应用704的数据单元可以暂时存储在持久性层712(例如,页面缓冲器或其他类型的临时持久性层)中,持久性层712可以将数据以存储页面的形式写入一个或多个存储装置714,例如经由输入/输出组件716。一个或多个存储装置714可以包括一

个或多个物理存储介质或设备(例如,硬盘驱动器、持久闪存、随机存取存储器、光学介质、磁介质等),其被配置用于写入数据用于长期存储。应当注意,存储装置714和输入/输出组件716可以被包括在计算系统702中,尽管它们在图7中被示为在计算系统702的外部。

[0062] 保存在长期存储装置714中的数据可以被组织成页面,每个页面已经分配给它一个定义量的存储空间。在一些实现中,分配给每个页面的存储空间量可以是恒定的和固定的。然而,分配给每个页面的存储空间量可以变化的其他实现也在当前主题的范围之内。

[0063] 图8示出了根据当前主题的一些实现的示例性软件架构800。可以在硬件和软件中的一个或多个中实现的数据存储应用704可以包括数据库应用、网络连接的存储系统等中的一个或多个。根据当前主题的至少一些实现,这种数据存储应用704可以包括持久性层712或其他类型的存储器缓冲器,或者以其他方式与持久性层712或其他类型的存储器缓冲器接口,例如经由持久性接口802。持久性层712内的页面缓冲器804可以存储一个或多个逻辑页面806,并且可选地可以包括影子页面(shadow page)、活动页面(active page)等。保留在持久性层712中的逻辑页面806可以经由输入/输出组件716被写入存储装置(例如,长期存储装置等)714,输入/输出组件716可以是软件模块、在软件和硬件中的一个或多个中实现的子系统等。存储装置714可以包括一个或多个数据体810,其中存储的页面812在物理存储块处被分配。

[0064] 在一些实现中,数据存储应用704可以包括页面管理器814和/或保存点管理器816,或者以其他方式与页面管理器814和/或保存点管理器816通信。页面管理器814可以与持久性层712处的页面管理模块820通信,该页面管理模块820可以包括空闲块管理器822,该空闲块管理器822监视页面状态信息824,例如存储装置714内的物理页面和持久性层712内(以及可选地在页面缓冲器804中)的逻辑页面的状态。保存点管理器816可以与持久性层712处的保存点协调器826通信,以处理保存点,保存点用于创建数据库的一致持久状态,以便在可能的崩溃之后重新启动。

[0065] 在数据存储应用704的一些实现中,持久性层712的页面管理模块可以实现影子分页(shadow paging)。页面管理模块820内的空闲块管理器822可以维护物理页面的状态。页面缓冲器804可以包括如本文讨论的那样操作的固定的页面状态缓冲器。转换器组件840可以是页面管理模块820的一部分或者与页面管理模块820通信,转换器组件840可以负责写入存储装置714的逻辑页面和物理页面之间的映射。转换器840可以在转换器表842中维护逻辑页面到相应物理页面的当前映射。转换器840可以维护逻辑页面806到一个或多个转换器表842中的相应物理页面的当前映射。当从存储装置714读取逻辑页806时,可以使用转换器840从一个或多个转换器表842中查找要加载的存储页面。当在保存点之后第一次将逻辑页面写入存储装置714时,新的空闲物理页面被分配给该逻辑页面。空闲块管理器822将新的物理页面标记为“已使用”,并且新的映射被存储在一个或多个转换器表842中。

[0066] 持久性层712可以确保在数据存储应用704中做出的改变是耐久的,并且数据存储应用704可以在重新启动后恢复到最近提交的状态。向存储装置714写入数据不需要与写入事务的结束同步。因此,未提交的改变可能写入磁盘,并且当写入事务完成时,提交的改变可能尚未写入磁盘。系统崩溃后,未完成的事务所做的改变可以回滚(rolled back)。已经提交的事务发生的改变不应在此过程中丢失。还可以包括记录器组件844,以在线性日志(linear log)中存储对数据存储应用的数据所做的改变。记录器组件844可在恢复期间用

于重现(replay)自最后保存点以来的操作,以确保所有操作都应用于数据,并且具有记录的“提交”记录的事务在恢复过程结束时回滚仍然打开的事务之前被提交。

[0067] 对于一些数据存储应用,向磁盘写入数据不一定与写入事务的结束同步。可能会出现这样的情况:未提交的改变被写入磁盘,同时,当写入事务完成时,提交的改变尚未写入磁盘。系统崩溃后,未完成的事务所做的改变必须回滚,并且提交的事务所做的改变不能丢失。

[0068] 为了确保提交的改变不会丢失,每当做出改变时,记录器组件844可以写入重做日志信息(redo log information)。该信息最迟可以在事务结束时写入磁盘。当正常数据写入数据体时,日志条目可以持久保存在单独的日志体中。使用重做日志,即使相应的数据页面没有写入磁盘,提交的改变也可以恢复。为了撤销(undoing)未提交的改变,持久性层712可以使用撤销日志条目(来自一个或多个日志)和影子分页的组合。

[0069] 持久性接口802可以处理存储装置(例如,内存存储装置等)的读取和写入请求。持久性接口802还可以提供用于在记录和不记录的情况下写入数据的写入方法。如果使用记录的写入操作,则持久性接口802调用记录器844。此外,记录器844提供允许存储装置(例如,内存中存储装置等)直接将日志条目添加到日志队列中的接口。记录器接口还提供请求将内存中日志队列中的日志条目刷新到磁盘的方法。

[0070] 日志条目包含日志序列号、日志条目类型和事务标识符。根据操作类型,记录器844记录附加信息。例如,对于“更新”类型的条目,这将是受影响记录的标识和被修改的数据的余像(after image)。

[0071] 当数据应用704重新启动时,需要处理日志条目。为了加快这个过程,重做日志并不总是从头开始处理。相反,如上所述,可以周期性地执行保存点,将自最后保存点以来(例如,在存储器中等)所做的所有改变写入磁盘。当启动系统时,只需要处理最后保存点之后创建的日志。在下一备份操作之后,可以移除保存点位置之前的旧日志条目。

[0072] 当记录器844被调用来写入日志条目时,它不会立即写入磁盘。相反,它可以将日志条目放入存储器中的日志队列中。日志队列中的条目最迟可以在相应的事务完成(提交或中止)时写入磁盘。为了保证提交的改变不会丢失,在相应的日志条目刷新到磁盘之前,提交操作不会成功完成。将日志队列条目写入磁盘也可能由其他事件触发,例如当日志队列页面已满或执行保存点时。

[0073] 通过当前主题,记录器844可以以自然顺序(例如,有序顺序等)将数据库日志(或在本文简称为“日志”)顺序地写入存储器缓冲器。如果使用几个物理硬盘/存储设备来存储日志数据,则可以定义几个日志划分。此后,记录器844(如上所述,其用于生成和组织日志数据)可以对所有可用的日志划分上的日志缓冲区进行负载平衡写入。在一些情况下,负载平衡是根据轮询分配(round-robin distribution)方案进行的,其中各种写入操作以顺序和连续的方式指向日志缓冲区。通过这种安排,写入多划分日志的特定划分的单个日志段的日志缓冲区不是连续的。但是,日志缓冲区可以在恢复期间从所有划分的日志段重新排序到正确的顺序。

[0074] 如上所述,数据存储应用704可以使用影子分页,使得保存点管理器816可以写入事务一致的保存点。在这种安排下,数据备份包括特定保存点中包含的所有数据页面的副本,这是数据备份过程的第一步。当前主题也可以应用于其他类型的数据页面存储装置。

[0075] 在一些实现中,当前主题可以被配置为在系统900中实现,如图9所示。系统900可以包括处理器910、存储器920、存储设备930和输入/输出设备940。组件910、920、930和940中的每一个都可以使用系统总线950互连。处理器910可以被配置为处理用于在系统900内执行的指令。在一些实现中,处理器910可以是单线程处理器。在替代实现中,处理器910可以是多线程处理器。处理器910可以进一步被配置为处理存储在存储器920中或存储设备930上的指令,包括通过输入/输出设备940接收或发送信息。存储器920可以在系统900内存储信息。在一些实现中,存储器920可以是计算机可读介质。在替代实现中,存储器920可以是易失性存储器单元。在又一些实现中,存储器920可以是非易失性存储器单元。存储设备930能够为系统900提供大容量存储。在一些实现中,存储设备930可以是计算机可读介质。在替代实现中,存储设备930可以是软盘设备、硬盘设备、光盘设备、磁带设备、非易失性固态存储器或任何其他类型的存储设备。输入/输出设备940可以被配置为向系统900提供输入/输出操作。在一些实现中,输入/输出设备940可以包括键盘和/或定点设备。在替代实现中,输入/输出设备940可以包括用于显示图形用户界面的显示单元。

[0076] 图10示出了根据当前主题的一些实现的用于执行数据所有权管理的示例性方法1000。在1002,可以选择多个索引中的索引(例如,图6a-6c所示的系统602或604中的索引“id”)。该索引可以对应于存储在数据库的多个数据库片(例如,图4所示的片1-5)中的多个数据值范围。对应于划分结构(例如,图4所示的结构412)的索引可以包括多个分层排列的节点(例如,节点413、415等)。每个节点可以对应于存储在至少一个数据库片中的多个数据值范围中的数据值范围。划分结构可以跨多个计算系统(例如,系统602、604,如图6a-6c所示)被复制。在1004,计算系统(例如,系统604)可以执行对一个或多个数据值范围的更新(例如,对节点647b、新分支或节点649b、新片6、7等中的数据值范围的更新)。在1006,计算系统可以将包括更新的一个或多个数据值范围的数据库片和包括更新的一个或多个数据值范围的节点中的至少一个复制到多个计算系统中的另一计算系统(例如,系统602),用于存储更新的一个或多个数据值范围的副本。

[0077] 在一些实现中,当前主题可以包括以下可选特征中的一个或多个。在一些实现中,复制可以包括基于对一个或多个数据值范围的更新的数量大于预定的更新阈值数量,将包括更新的一个或多个数据值范围的节点复制到另一计算系统。

[0078] 在一些实现中,更新执行可以包括生成被配置为存储更新的一个或多个数据值范围的另一数据库片。该方法还可以包括由计算系统将生成的另一数据库片复制到其他数据库系统。该方法还可以包括由另一计算系统存储生成的其他数据库片,由另一计算系统执行对另一数据库片中的一个或多个数据值范围的更新,以及由另一计算系统将另一数据库片复制到计算系统,以及由计算系统将更新的一个或多个数据值范围的副本存储在另一数据库片中。

[0079] 在一些实现中,多个片中的一个或多个片可以被配置为独立于多个分层排列的节点中的一个或多个节点而被多个计算系统中的一个或多个计算系统所拥有。一个或多个片的所有权可以被配置为由一个或多个计算系统独立于一个或多个节点的所有权来转移。

[0080] 在一些实现中,复制可以包括异步复制。

[0081] 在一些实现中,更新的执行可以包括以下至少一个:将对一个或多个数据值范围的更新插入到一个或多个数据库片中,使用对一个或多个数据值范围的更新来修改存储在

一个或多个数据库片中的数据,删除一个或多个数据库片中的数据,以及它们的任意组合。

[0082] 本文公开的系统和方法可以以各种形式来体现,包括例如数据处理器,诸如也包括数据库的计算机、数字电子电路、固件、软件或它们的组合。此外,本公开实现的上述特征和其他方面和原理可以在各种环境中实现。这种环境和相关应用可以被专门构造用于执行根据所公开的实现的各过程和操作,或者它们可以包括由代码选择性激活或重新配置的通用计算机或计算平台,以提供必要的功能。本文公开的过程并不固有地与任何特定的计算机、网络、架构、环境或其他装置相关,并且可以通过硬件、软件和/或固件的适当组合来实现。例如,各种通用机器可以与根据所公开的实现的教导编写的程序一起使用,或者构建专用装置或系统来执行所需的方法和技术会更方便。

[0083] 本文公开的系统和方法可以被实现为计算机程序产品,即,有形地体现在信息载体中的计算机程序,例如,在机器可读存储设备中或者在传播信号中,用于由数据处理设备执行或者控制数据处理装置(例如,可编程处理器、计算机或者多个计算机)的操作。计算机程序可以用任何形式的编程语言编写,包括编译或解释语言,并且它可以以任何形式部署,包括作为独立程序或作为模块、组件、子例程或适合在计算环境中使用的其他单元。计算机程序可以部署成在一个计算机上或在一个站点的多个计算机上执行,或者分布在多个站点上并通过通信网络互连。

[0084] 如本文所使用的,术语“用户”可以指任何实体,包括人或计算机。

[0085] 尽管在一些情况下,诸如第一、第二等序数可以与顺序相关;如本文档所使用的,序数不一定暗示顺序。例如,序数仅用来区分一个项目和另一项目。例如,区分第一事件和第二事件,但不需要暗示任何时间顺序或固定的参考系统(使得描述的一个段落中的第一事件可以不同于描述的另一段落中的第一事件)。

[0086] 前面的描述旨在说明而不是限制本发明的范围,本发明的范围由所附权利要求的范围限定。其他实现在所附权利要求的范围内。

[0087] 这些计算机程序也可以指程序、软件、软件应用、应用、组件或代码,包括用于可编程处理器的机器指令,并且可以用高级过程和/或面向对象的编程语言和/或汇编/机器语言来实现。如本文所使用的,术语“机器可读介质”是指用于向可编程处理器提供机器指令和/或数据的任何计算机程序产品、装置和/或设备(诸如例如磁盘、光盘、存储器和可编程逻辑器件(PLD)),包括接收机器指令作为机器可读信号的机器可读介质。术语“机器可读信号”是指用于向可编程处理器提供机器指令和/或数据的任何信号。机器可读介质可以非瞬态地存储这种机器指令,诸如例如非瞬态固态存储器或磁性硬盘驱动器或任何等效的存储介质。机器可读介质可以替代地或附加地以瞬态方式存储这种机器指令,诸如例如与一个或多个物理处理器核相关联的处理器高速缓存或其他随机存取存储器。

[0088] 为了提供与用户的交互,本文描述的主题可以在计算机上实现,该计算机具有显示设备,诸如例如阴极射线管(CRT)或液晶显示器(LCD)监视器,用于向用户显示信息,以及键盘和定点设备,诸如例如鼠标或轨迹球,用户可以通过它们向计算机提供输入。也可以使用其他种类的设备来提供与用户的交互。例如,提供给用户的反馈可以是任何形式的感觉反馈,诸如例如视觉反馈、听觉反馈或触觉反馈;并且来自用户的输入可以以任何形式接收,包括但不限于声音、语音或触觉输入。

[0089] 本文描述的主题可以在计算系统中实现,该计算系统包括后端组件,诸如例如一

个或多个数据服务器,或者包括中间件组件,诸如例如一个或多个应用服务器,或者包括前端组件,诸如例如具有图形用户界面或网络浏览器的一个或多个客户端计算机,用户可以通过该图形用户界面或网络浏览器与本文描述的主题的实现进行交互,或者这种后端、中间件或前端组件的任意组合。系统的组件可以通过任何形式或介质的数字数据通信互连,诸如例如通信网络。通信网络的示例包括但不限于局域网(“LAN”)、广域网(“WAN”)和因特网。

[0090] 计算系统可以包括客户端和服务端。客户端和服务端通常但不排他地彼此远离,并且通常通过通信网络进行交互。客户端和服务端之间的关系是通过在各自计算机上运行并彼此具有客户端-服务端关系的计算机程序产生的。

[0091] 前面描述中阐述的实现并不代表与本文描述的主题一致的所有实现。相反,它们仅仅是与所描述的主题相关的方面一致的一些示例。尽管上面已经详细描述了一些变化,但是其他修改或添加也是可能的。特别地,除了本文阐述的那些之外,还可以提供进一步的特征和/或变化。例如,上述实现可以针对所公开的特征的各种组合和子组合和/或上面公开的几个进一步特征的组合和子组合。此外,附图中描绘的和/或本文中描述的逻辑流程不一定需要所示的特定顺序或有序顺序来实现期望的结果。其他实现可以在所附权利要求的范围内。

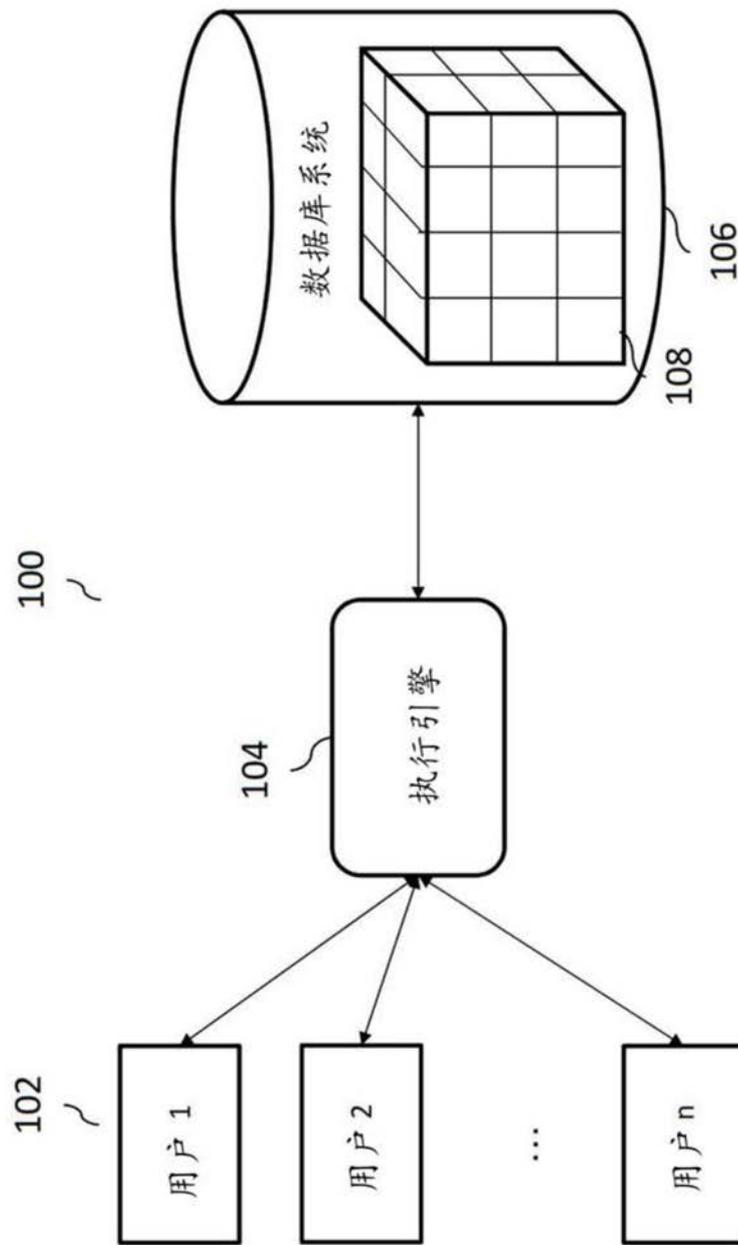


图1

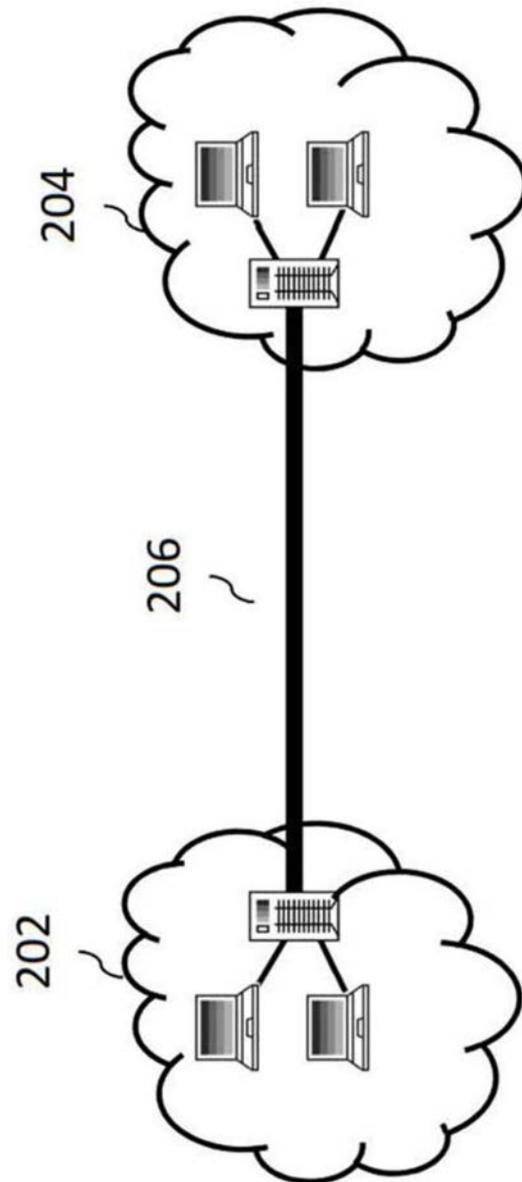


图2

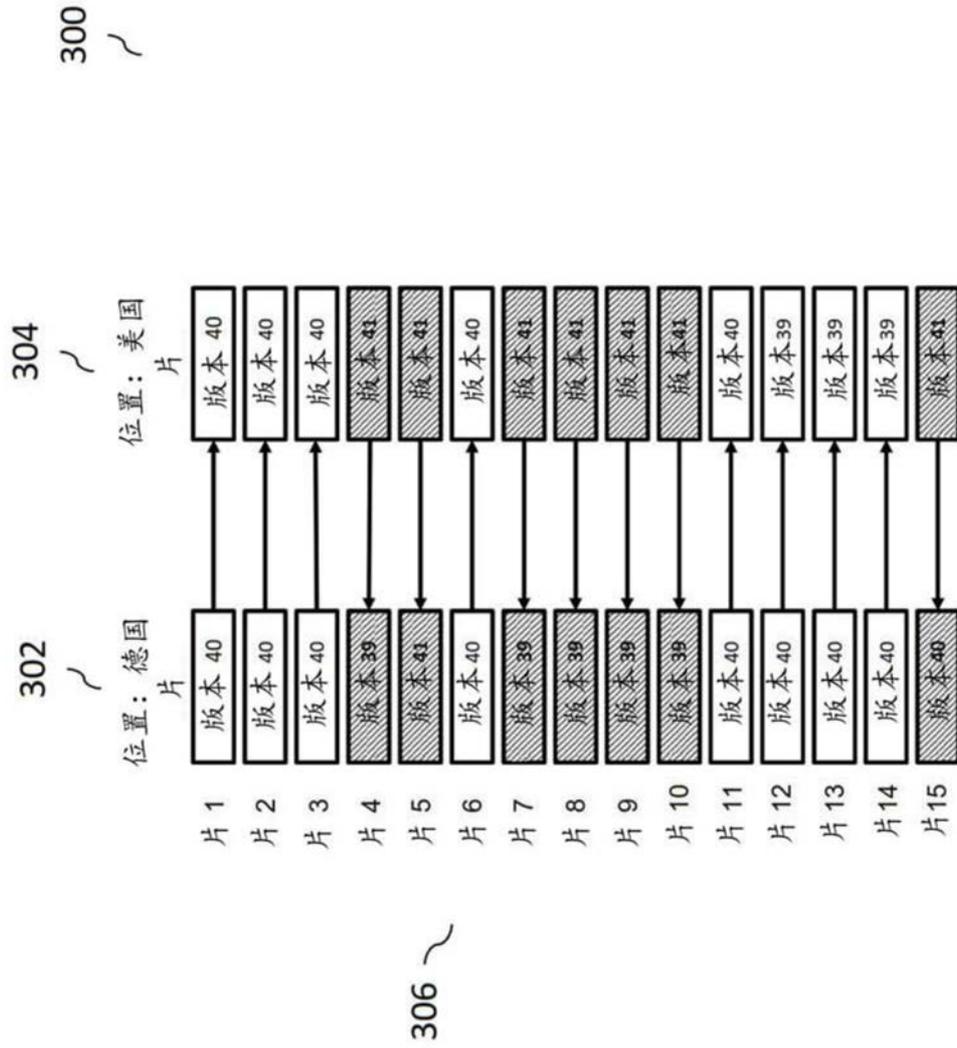


图3

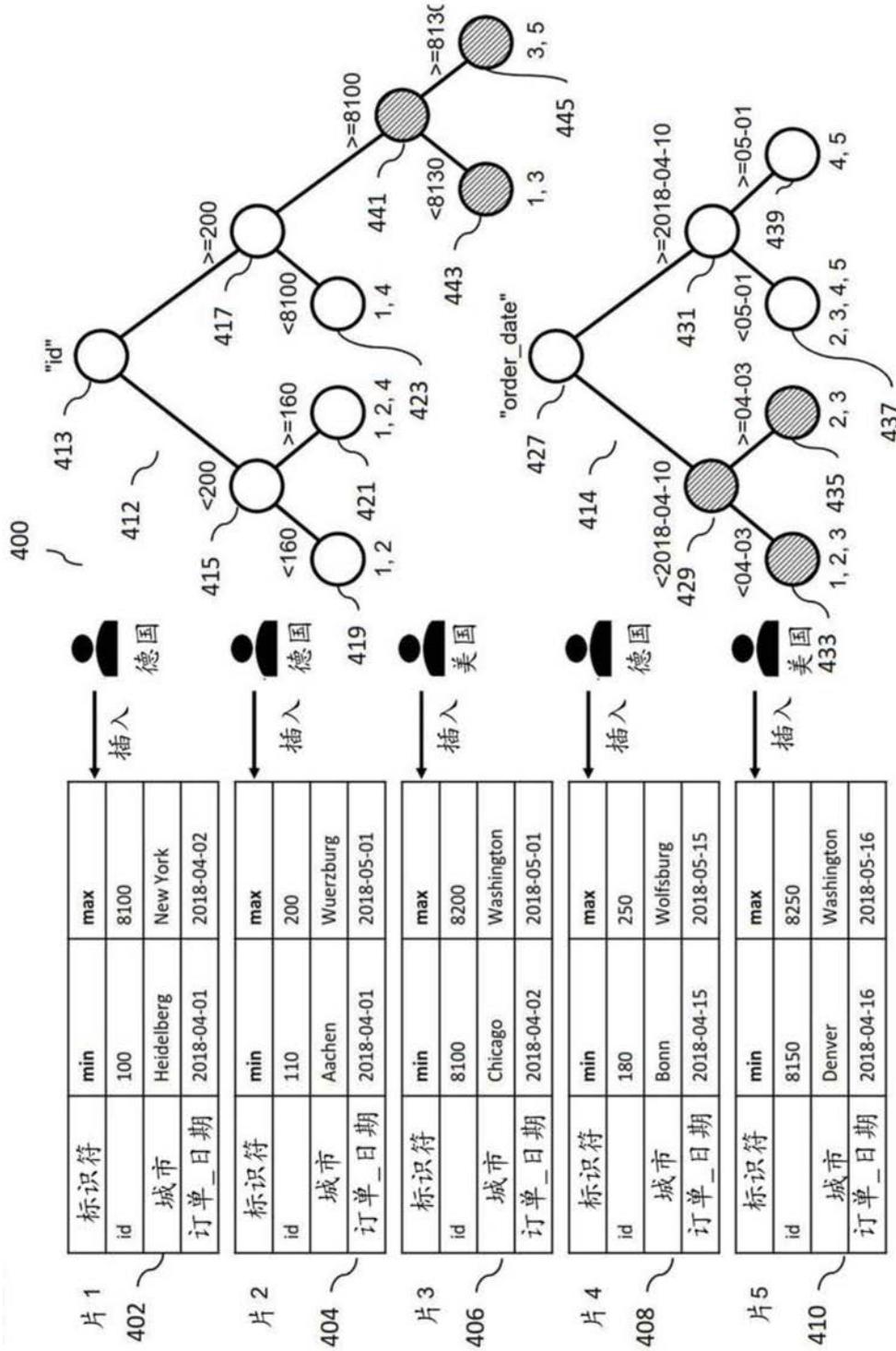


图4

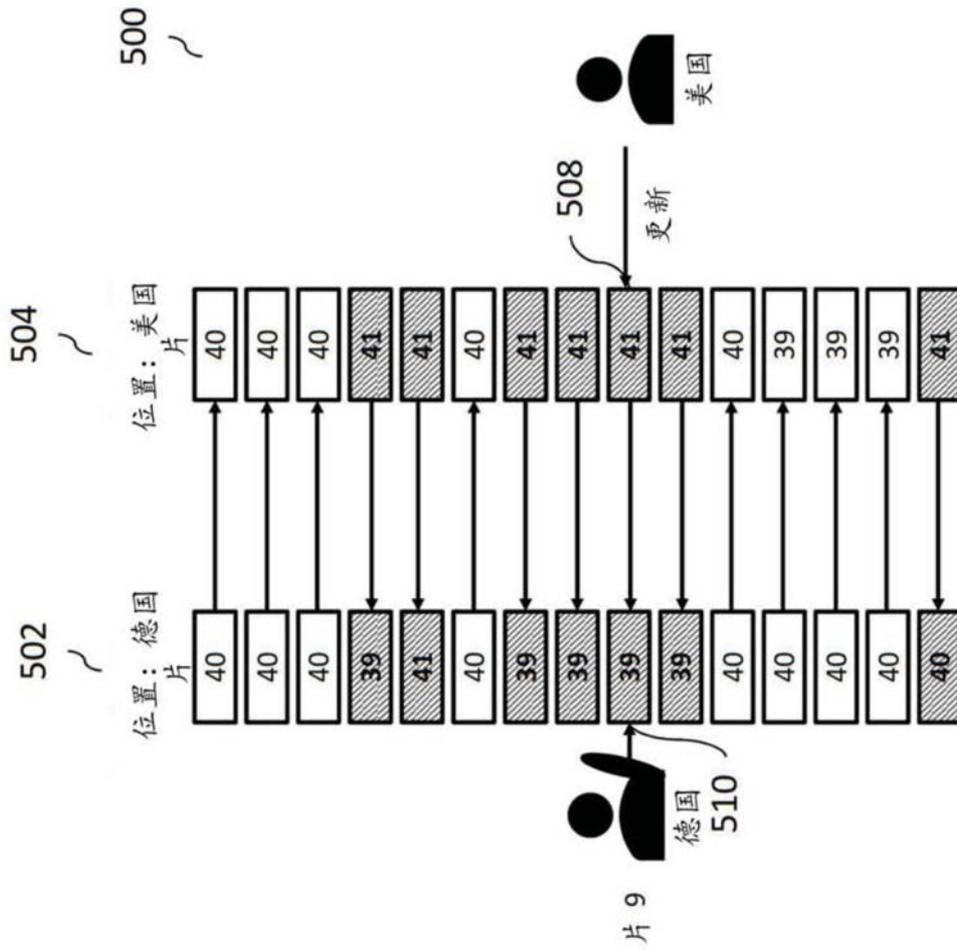


图5

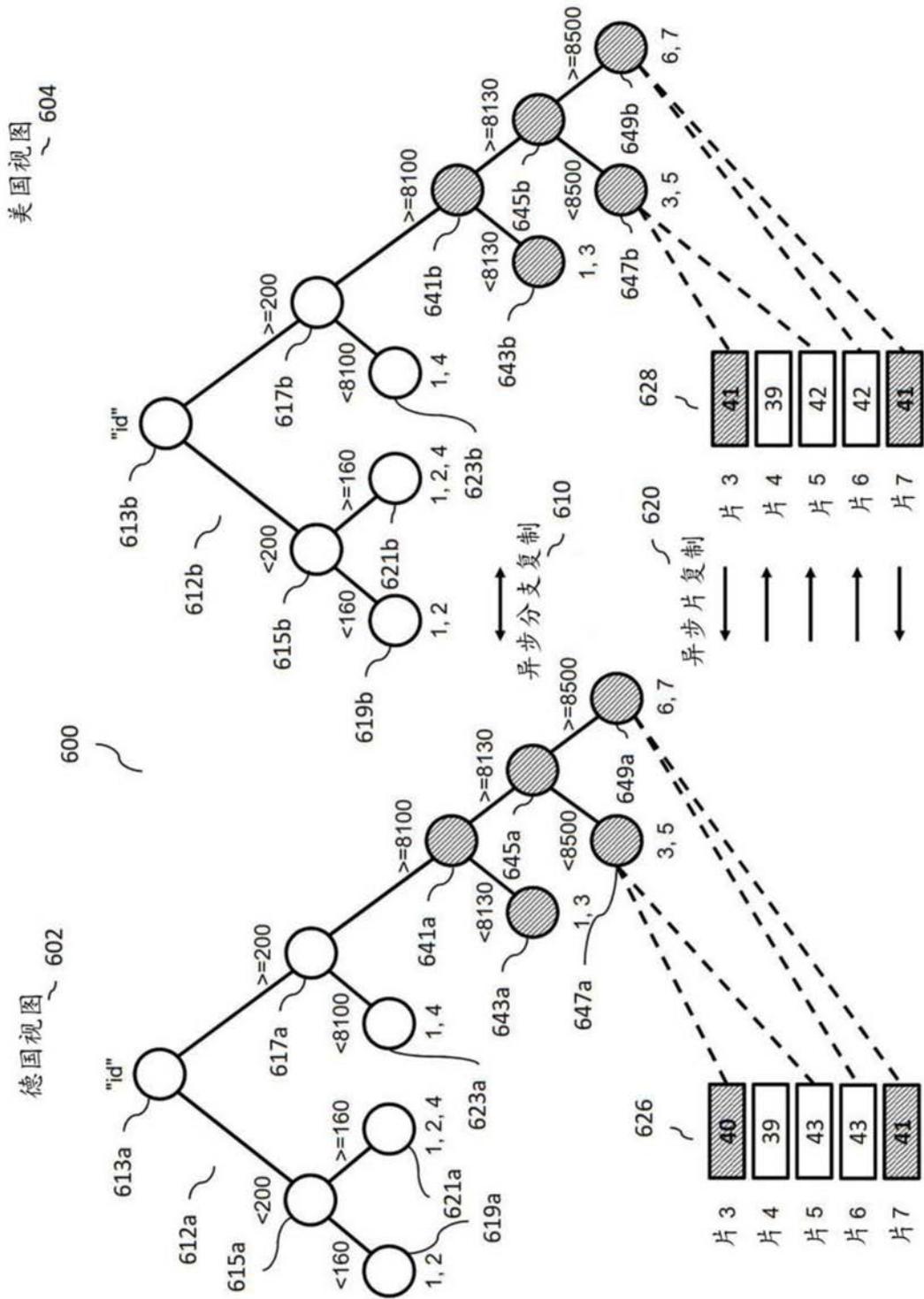


图6b

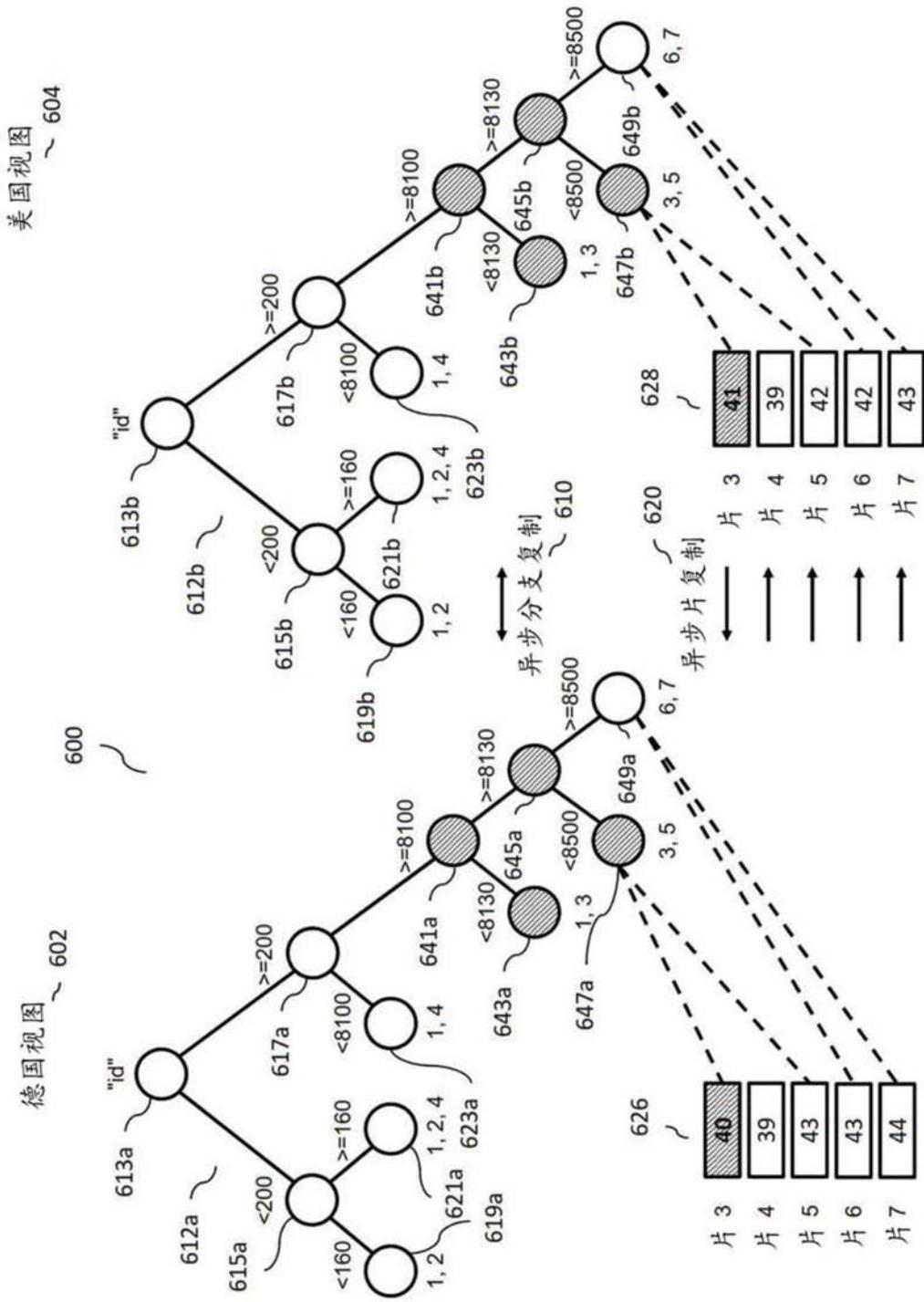


图6c

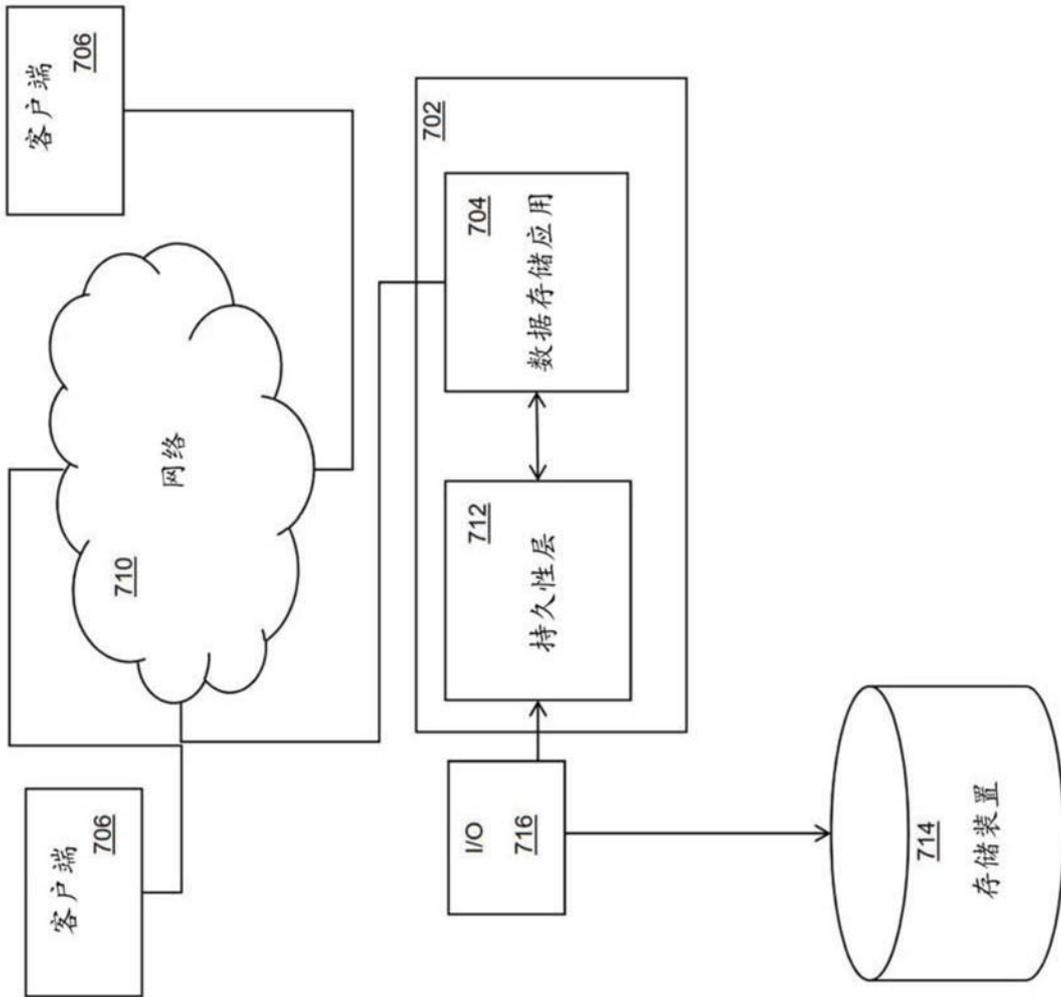


图7

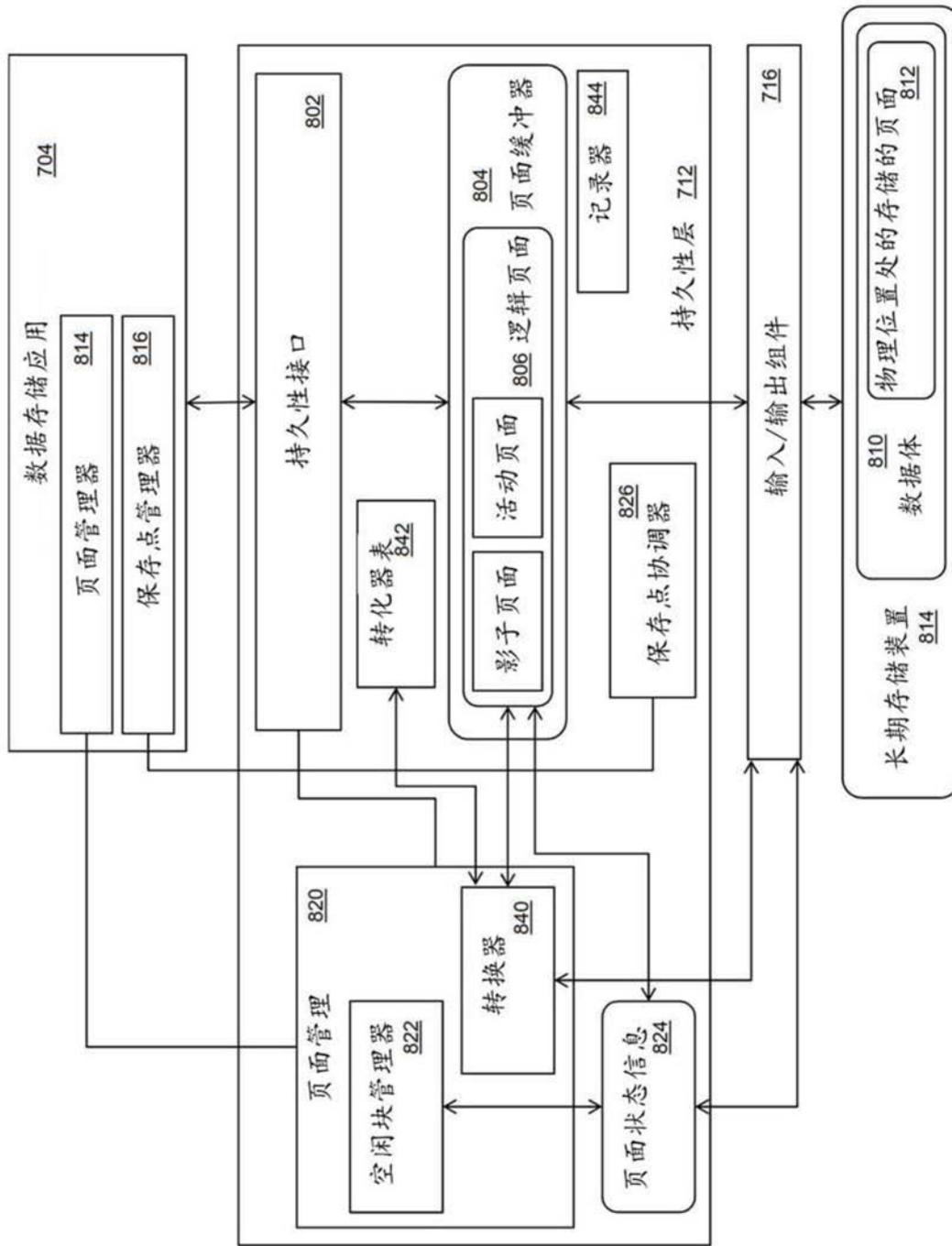


图8

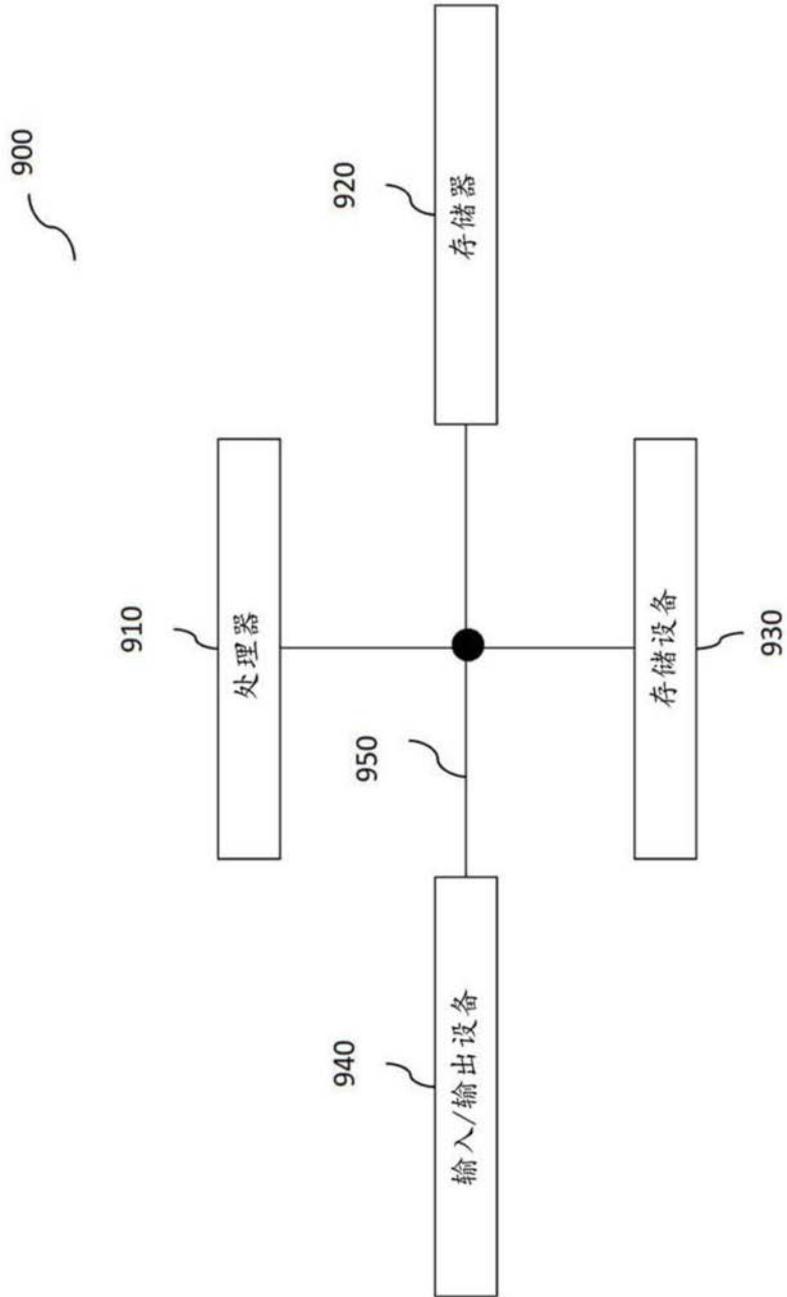


图9

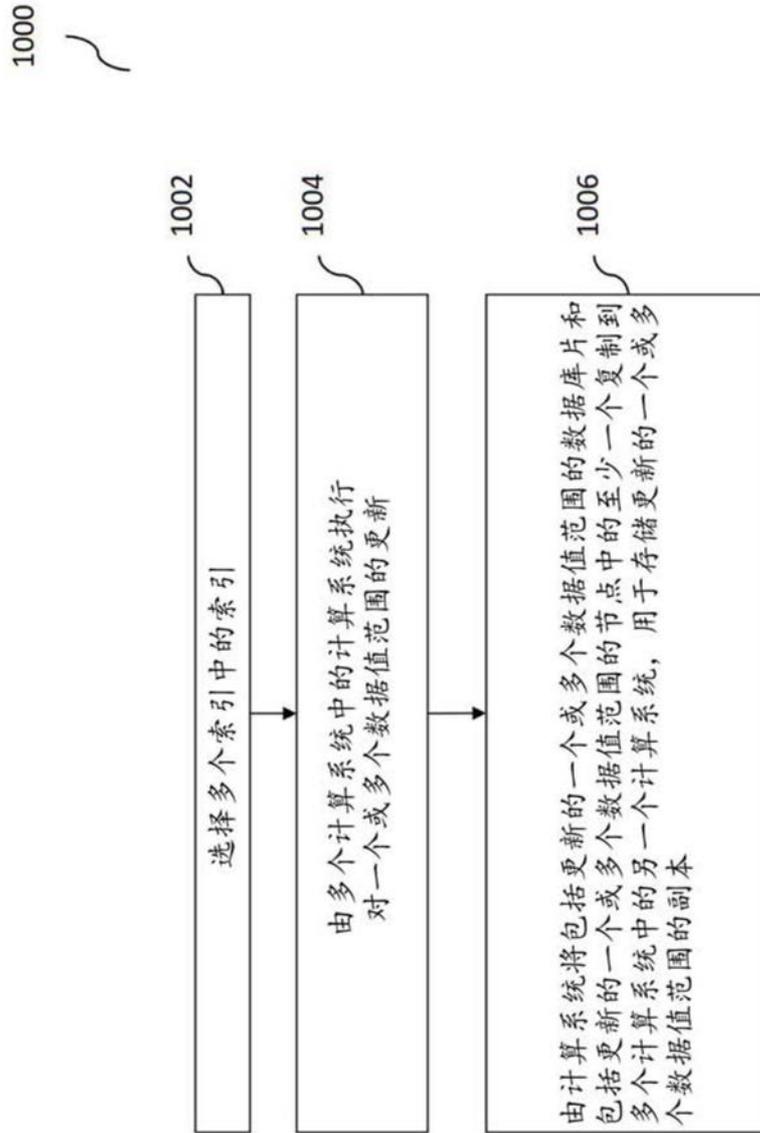


图10