



(12) 发明专利

(10) 授权公告号 CN 102142024 B

(45) 授权公告日 2016. 04. 06

(21) 申请号 201110037158. 2

(22) 申请日 2011. 01. 31

(30) 优先权数据

12/697, 302 2010. 02. 01 US

(73) 专利权人 微软技术许可有限责任公司

地址 美国华盛顿州

(72) 发明人 H·H·胡 吴中伟 O·N·泽利格
S·帕尔 C·罗伯 B·H·M·德尼
L·诺维克 N·R·埃利斯
T·塔留斯

(74) 专利代理机构 上海专利商标事务所有限公司 31100

代理人 陈斌 钱静芳

(51) Int. Cl.

G06F 17/30(2006. 01)

(56) 对比文件

US 20100005124 A1, 2010. 01. 07,
US 20100005124 A1, 2010. 01. 07,
US 7107419 B1, 2006. 09. 12,
US 7222141 B2, 2007. 05. 22,
US 20050138081 A1, 2005. 06. 23,
US 20050240633 A1, 2005. 10. 27,
US 20070130226 A1, 2007. 06. 07,

审查员 李楠

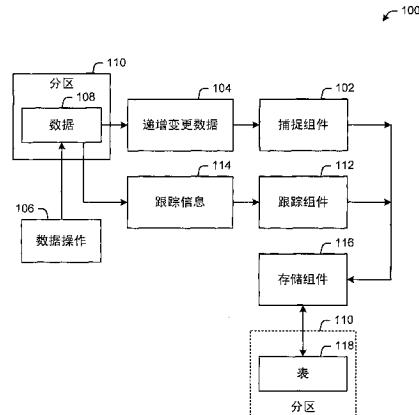
权利要求书2页 说明书9页 附图8页

(54) 发明名称

在分布式数据库中使用递增捕捉来进行逻辑
数据备份和回退

(57) 摘要

一种消除了对数据的盘上完整备份的需求而
只在单独的表中保留已发生的变更的体系结构。
因此，该体系结构允许对关系数据库（例如，SQL）
中的递增变更进行递增还原。该体系结构提供经
改进的还原时间和还原点目标。通过使用对已变
更数据（例如，以 XML 为格式）的递增捕捉，提供
了捕捉模式变更、查询递增变更数据、并将用户数
据高效地还原到更早时间点状态的能力。通过一
组触发器来（例如，连续地）跟踪变更（例如，
插入、更新以及删除操作），并且以人类可读格式
(例如，XML) 将递增地捕捉的已变更的行插入数
据捕捉表中（差分变更“增量”表）。还提供了回
退。



1. 一种具有物理介质的计算机实现的数据管理系统 (100), 包括 :

捕捉递增变更数据的捕捉组件 (102), 所述递增变更数据与在分布式数据库的一分区中的数据上的数据操作相关联 ;

创建与所述递增变更数据的数据操作相关联的跟踪信息的跟踪组件 (112) ;

将所述递增变更数据和相关联的跟踪信息存储在所述分区的一个表中的存储组件 (116), 所述递增变更数据可用, 所述表包括递增变更数据的变更的历史, 所述变更与数据操作发生时间、事务发生时间、以及行创建时间相关联, 其中所述递增变更数据和所述跟踪信息以包括所述表中的已变更数据的自描述模式的人类可读的格式存储在所述表中, 所述递增变更数据和所述跟踪信息在与将所述数据变更提交给所述数据库的同一事务中被存储在所述表中 ; 以及

用于将所述行的状态还原到先前时间点的回退组件, 其中在回退期间, 在所请求的回退时间之前的最后提交的行状态被用来覆盖基行。

2. 如权利要求 1 所述的系统, 其特征在于, 还包括自动地启动所述递增变更数据的捕捉、所述跟踪信息的创建以及所述递增变更数据和所述跟踪信息两者在所述表中的存储的一个或多个触发器。

3. 如权利要求 1 所述的系统, 其特征在于, 所述递增变更数据被持久存储在所述数据所处的同一分区中。

4. 如权利要求 1 所述的系统, 其特征在于, 所述递增变更数据是可用的并且可根据查询语言来搜索。

5. 如权利要求 1 所述的系统, 其特征在于, 还包括保留策略组件, 所述保留策略组件便利于对保留策略的创建以及所述保留策略到所述递增变更数据和相关联的跟踪信息的应用。

6. 一种使用处理器和存储器的计算机实现的数据管理方法, 包括 :

接收与分布式数据库中的数据的递增变更数据相关的数据操作 (600) ;

响应于所述数据操作来捕捉所述递增变更数据 (602) ;

创建与所述递增变更数据相关联的跟踪信息 (604) ;

将所述递增变更数据和跟踪信息存储在表中 (606), 所述表包括递增变更数据的变更的历史, 所述变更与数据操作发生时间、事务发生时间、以及行创建时间相关联, 其中所述递增变更数据和所述跟踪信息以包括所述表中的已变更数据的自描述模式的人类可读的格式存储在所述表中, 所述递增变更数据和所述跟踪信息在与将所述数据变更提交给所述数据库的同一事务中被存储在所述表中 ; 以及

基于所述递增变更数据将回退操作应用于所述数据库以将所述数据回退到先前时间点, 其中在回退期间, 在所请求的回退时间之前的最后提交的行状态被用来覆盖基行。

7. 如权利要求 6 所述的方法, 其特征在于, 还包括将所述递增变更数据作为所述表中的一行来存储。

8. 如权利要求 6 所述的方法, 其特征在于, 还包括使用结构化查询语言来查询所述递增变更数据。

9. 如权利要求 6 所述的方法, 其特征在于, 还包括将保留策略应用于所述递增变更数据和跟踪信息。

10. 如权利要求 6 所述的方法,其特征在于,还包括与所述递增变更数据的事务相关联地存储事务时间戳和数据操作时间戳。

在分布式数据库中使用递增捕捉来进行逻辑数据备份和回退

技术领域

[0001] 本发明涉及数据备份和回退技术,更具体地涉及递增的数据备份和回退技术。

背景技术

[0002] 在大型分布式数据库系统中,可以使用传统 SQL(结构化查询语言)备份和还原来进行数据保护。然而,尚有缺点存在。需要完整数据库的备份的盘上副本,这需要与所备份的数据库一样多的存储。另外,需要对事务日志进行周期性备份来用于较低的还原点目标(RPO)。此外,还原时间目标(RTO)很差,因为不管所涉及的数据的大小,任何还原操作都将需要还原整个备份并随后应用一系列事务日志备份文件。该过程非常耗时并且劳动力密集的,并且备份是以不可被直接查询的二进制格式来存储的。

发明内容

[0003] 下面提供了简化的概述,以便提供对此处所描述的一些新颖实施方式的基本理解。本“发明内容”不是详尽的综述,并且它不旨在标识关键 / 重要元素或描绘本发明的范围。其唯一的目的是以简化形式提出一些概念,作为稍后提出的更详细描述的序言。

[0004] 所公开的体系结构是一种消除了对数据的盘上完整备份的需求的有成本竞争力的方法。通过只在单独的表中保留已发生的变更来优化存储。因此,该体系结构允许对关系数据库(例如,SQL)中的递增变更进行递增还原。该体系结构提供经改进的还原时间和还原点目标。通过使用对已变更数据(例如,以 XML 为格式)的递增捕捉,提供了捕捉模式变更、查询递增地捕捉的数据、并将用户数据高效地还原到更早时间点状态并且没有停机时间的能力。

[0005] 通过一组触发器来(例如,连续地)跟踪变更(例如,插入、更新以及删除操作),并且根据某一格式(例如,XML)将递增地捕捉的已变更的行插入数据捕捉表中(差分变更“增量”表)。该格式是自描述的并且包含该格式内的行的模式。

[0006] 数据回退将来自数据捕捉表的适当行的递增变更解压到更早时间点,并随后将这些行覆盖到生产数据。优化插入操作以不在数据捕捉表中创建各行而是在基表中维护变更跟踪信息(例如,插入的协调世界时间(UTC))。

[0007] 为了为实现上述及相关目的,本文结合下面的描述和附图来描述某些说明性方面。这些方面指示了可以实践本文所公开的原理的各种方式,并且所有方面及其等效方面旨在落入所要求保护的主题的范围内。结合附图阅读下面的详细描述,其他优点和新颖特征将变得显而易见。

附图说明

[0008] 图 1 示出根据所公开的体系结构的计算机实现的数据管理系统。

[0009] 图 2 示出数据管理系统的一替换实施方式。

- [0010] 图 3 示出数据管理系统的一替换表示。
- [0011] 图 4 示出一系统, 该系统示出为了更新和删除数据操作而对基表进行的修改。
- [0012] 图 5 示出示例性捕捉和变更跟踪表。
- [0013] 图 6 示出根据所公开的体系结构的计算机实现的数据管理方法。
- [0014] 图 7 示出图 6 的方法的其他方面。
- [0015] 图 8 示出可用于根据所公开的体系结构来使用镜像之前 (before-image) 捕捉执行逻辑数据备份和回退的计算系统的框图。
- [0016] 图 9 示出使用镜像之前捕捉来执行逻辑数据备份和回退的计算环境的示意框图。

具体实施方式

[0017] 所公开的体系结构提供高效地捕捉数据操作的“镜像之前”(递增) 数据变更以及变更跟踪信息并随后使用这些变更和信息来高效地回退更新 / 删除 / 插入操作的能力。镜像之前数据是对其进行了变更,但在该已变更数据被处理来覆盖生产数据之前的数据。递增数据的存储空间是用于整个备份和事务日志备份的空间的一小部分。此外,从数据捕捉表回退更新 / 删除 / 插入操作的时间相对于从整个备份和事务日志备份进行还原的时间而言是很小的。

[0018] 如本文所使用的,表是采用控制跨服务器的划分的划分键并且还采用控制在一服务器内各行的排序的群集键的逻辑关系。表群是具有同一划分键的表的集合。行群是具有同一划分键值的表群中的行的集合。行群完全处于一个服务器上,但可以不被群集在一起。每一表群可以跨节点分布。每一存储节点是被分配了一定范围(分区)的键值,并且出于耐久性目的而复制了每一分区。

[0019] 镜像之前数据可以用包含这些行的自描述模式的 XML(可扩展标记语言) 格式来持久存储。因此,该解决方案对模式评估是有效的。另外,镜像之前数据可以被持久存储在同一分区中(例如,表群),并且因而是高度可用的。此外,自动地管理并维护镜像之前保留策略,并且该镜像之前数据可以经由诸如 TSQL(事务结构化查询语言(SQL)) 等传统关系语言来查询。

[0020] 选择镜像之前优于镜像之后,使得变更可以从该分区内的当前数据向后应用(撤消而非重做),并且因此消除了对完整分区备份的需求以节省存储空间。

[0021] 现在将参考附图,全部附图中相同的附图标记用于表示相同的元件。在下面的描述中,为了进行说明,阐述了很多具体细节以便提供对本发明的全面理解。然而,显而易见,可以没有这些具体细节的情况下实施各新颖实施例。在其他情况下,以框图形式示出了公知的结构和设备以便于描述它们。本发明将涵盖落入所要求保护的主题的精神和范围内的所有修改、等效方案和替换方案。

[0022] 图 1 示出了根据所公开的体系结构的计算机实现的数据管理系统 100。系统 100 包括捕捉与对分布式数据库的分区 110 中的数据 108 的数据操作 106 相关联的递增变更数据 104 的捕捉组件 102。系统 100 还可包括创建与递增变更数据 104 的数据操作 106 相关联的跟踪信息 114 的跟踪组件 112 和将递增变更数据 104 及相关联的跟踪信息 114 存储在分区 110 的表 118 中的存储组件 116。在需要时,递增变更数据 104 高度可用于访问。捕捉组件 102 和跟踪组件 112 将数据和信息传递到存储组件 116。

[0023] 递增变更数据 104 和跟踪信息 114 以人类可读的格式（例如，XML——可扩展标记语言）存储在表 118 中，该格式在表 118 中包括各行的自描述模式。递增变更数据 104 和跟踪信息 114 在与将数据 108 提交给数据库的同一事务中被存储在表 118 中。递增变更数据 104 被持久存储在与数据 108 所处的相同分区（分区 110）中。递增变更数据 104 是高度可用和可根据查询语言搜索的。表 118 包括递增变更数据 104 的变更的历史。变更与数据操作发生的时间、事务发生的时间、或行创建的时间中的至少一个相关联。

[0024] 图 2 示出数据管理系统 200 的一替换实施方式。系统 200 包括捕捉与对分布式数据库的分区 110 中的数据 108 的数据操作 106 相关联的递增变更数据 104 的捕捉组件 102、创建与数据操作 106 相关联的跟踪信息 114 的跟踪组件 112、将递增变更数据 104 和相关联的跟踪信息 114 存储在表 118 中的存储组件 116。在此，各组件（102, 112, 116, 204 以及 206）与触发器 202 之间的通信由双向箭头来表示，因为这些实体之间可在两个方向上都存在通信。系统 200 还可包括自动地启动对递增变更数据 104 和跟踪信息 114 的捕捉、跟踪以及将它们存储在表 118 中的一个触发器（一个或多个触发器 202）。

[0025] 递增变更数据 104 和跟踪信息 114 以人类可读的格式（例如，XML——可扩展标记语言）存储在表 118 中，该格式包括来自基表 108 的各行的自描述模式。递增变更数据 104 和跟踪信息 114 在与将数据 108 提交给数据库的同一事务中被存储在表 118 中。递增变更数据 104 被持久存储在数据 108 所处的分区 110 中，是高度可用的，并且是可以使用查询语言来搜索的。表 118 包括递增变更数据的变更的历史。变更与数据操作发生的时间（数据操作时间戳）、事务发生的时间（事务时间戳）以及行创建的时间（协调世界时间——UTC）相关联。

[0026] 系统 200 还可包括便于创建保留策略并将其应用于递增变更数据 104 和相关联的跟踪信息 114 的保留策略组件 204 以及用于将数据 108 的状态还原到先前时间点的回退组件 206。

[0027] 图 3 示出数据管理系统 300 的一替换表示。可以在云计算环境中将系统 300（以及例如，系统 100 和 200）用于备份和还原功能。对于数据库中的表群 302 中的（包括主副本和多个次副本的多个副本中的）每一分区，创建变更捕捉和跟踪表 304（例如，图 1 是表 118）。在对原始表群 302 进行某些变更时，该表 304（也被称为递增变更表或“增量”表）存储来自原始表群 302 的行值。

[0028] 系统 300 可被认为是与表群 302 进行接口的两个不同的应用。备份服务 306 安装数据捕捉和跟踪表、触发器 202（例如，SQL 更新和删除触发器），并向基表添加三个跟踪列。这些附加列被用来跟踪插入操作。这些组件备份该表群的数据。还原工具 308 允许用户将表群 302 的先前状态还原到分阶段数据库 310。

[0029] 为了执行备份功能，为每一表群中的每一个表的删除和更新操作安装触发器 202。触发器 202 捕捉对基行进行的变更并将记录插入数据捕捉和跟踪表 304 中。这些值随后可从数据捕捉和跟踪表 304 被插回（或，还原）到原始数据库或分阶段数据库。在还原时，获取在时间上最接近所请求的还原时间的记录并且将变更应用回基行。以此方式，数据库的所有者可以在设定期限内撤消对数据库进行的动作。

[0030] 触发器捕捉对基行进行的变更并且将镜像之前（递增变更数据）插入到该系统所维护的数据捕捉和跟踪表 304。表 304 被用作用来回退（还原）操作的变更的历史。数据

捕捉和跟踪表 304 被持久存储在与用户数据相同的表群中，使得表 304 与多个副本高度可用。此外，因为镜像之前是 XML 的格式，所以它是易于查询的。

[0031] 图 4 示出一系统 400，该系统示出为了更新和删除数据操作而对基表 402 进行的修改。存在着被用来跟踪插入、删除、以及更新操作的三个列。这些列包括 UTC 时间戳和两个 DBTS（数据库时间戳）列（TX DBTS 和 OP DBTS）。TX DBTS 列跟踪插入、删除以及更新操作的较大事务上下文，而 OP DBTS 列跟踪单个插入、删除以及更新操作。

[0032] 在变更变得永久（覆盖在生产数据上）之前，来自基表的数据被存储在捕捉和变更跟踪表 304 中。对于每一表群每一分区都存在一个活动的捕捉和变更跟踪表 304。各列由基表数据的 XML 版本、用于帮助还原操作的 UTC 和 DBTS 跟踪列、以及与备份和还原两者都相关的其他元数据所组成。在对基表执行更新和删除操作时，对应触发器（删除触发器 404 和更新触发器 406）激发并将旧数据复制到捕捉和变更跟踪表 304 中。

[0033] 存在着关于捕捉和变更跟踪表 304 的视图 408。触发器（404 和 406）引用这一视图 408 来间接引用大多数最近（活动）捕捉和变更跟踪表。捕捉和变更跟踪表 304 不能被直接引用，因为它的实际名称在清除旧捕捉和变更跟踪表期间变化。

[0034] 可以提供存储基表的各模式版本的信息表。这对于跟踪跨还原操作的模式变更而言是有用的。备份服务 306 安装所有以上组件，并且还清除旧的（过期）捕捉和变更跟踪表。

[0035] 还原所存储的过程 410 基于给定 UTC 时间将该分区还原到先前状态。过程 410 与其他组件一起安装。一个过程还原行群，并且另一过程还原整个分区。

[0036] 在执行还原操作时，使用三个列来跟踪先前插入、删除以及更新操作的次序。这些列具有在每一插入、删除、以及更新操作时填充的默认值。每一列的默认值调用在插入、删除或更新操作期间自动地填充正确的值的函数（内部函数）。

[0037] TX DBTS 允许通过事务来对基表及捕捉和变更跟踪表 304 中的各条目进行分类和编组。这使用返回该单独操作的当前事务的 DBTS 的内部函数。在服务器上，这一内部函数存储整个事务的当前 DBTS 并随后递增该 DBTS。在这一内部函数被调用时，给定事务中的任何操作将具有同一 DBTS 值。任何其他 DBTS 引用将返回已递增的、当前非事务 DBTS 值。

[0038] OP DBTS 展示每一操作的次序，并使用返回当前 DBTS 并递增该 DBTS 的内部函数。对于基表中的每一条目，这是不同的，因为每一插入操作经由该列的默认值来调用内部函数。

[0039] UTC 时间戳是创建该行的插入操作的当前 UTC 时间。这一时间戳用于帮助还原操作。用户可以选择需要还原到的 UTC 日期或时间。还原基于 DBTS 值使用 UTC 时间戳作为指导来确定实际还原点。这被用来将数据库还原到一致的事务状态。

[0040] 使用被该系统添加到基表中的各列来跟踪基表中的插入操作。这些列存储该操作的序列号、在该分区的主副本处测量的该操作的 UTC 时间、以及事务标识符。（这三个字段也被添加到数据捕捉表。）对于插入操作，不向数据捕捉表插入行。这一优化以稍微增加回退逻辑的复杂性为代价而减少了存储要求。

[0041] 在回退期间，行（它已被保存在数据捕捉表中）在所请求的回退时间之前的最后提交的状态被用来覆盖基行。

[0042] 插入操作没有触发器。在对备份的基表进行插入时，在捕捉和变更跟踪表中不存

储信息。自动地填充三个附加列的默认值。这一新数据允许还原操作考虑插入操作并对其进行排序。

[0043] 为了在更新或删除操作期间对数据进行备份,备份服务 306 为相关基表安装触发器。在对基表的更新或删除操作时,触发器存储预更新或预删除基表行的 XML 表示(没有三个跟踪列)。这捕捉用于备份的数据以及数据库的“模式”。其次,它存储分区键。基于插入触发器的 DBTS 记录不更改基表。在备份服务 306 不运行的情况下,触发器继续将已变更的数据存储到捕捉和变更跟踪表 304 中。

[0044] 使用捕捉和变更跟踪表 304 上的视图 408,以使得在为行群内的给定分区做出新捕捉和变更跟踪表并且该表自身的名称变更时,触发器不必变更。

[0045] 更新和删除触发器两者都将必要的备份数据插入已安装的视图中。无论何时创建带有基于其期满日期的新名称的新捕捉和变更跟踪表,该视图被重建为指向该新表。

[0046] 备份服务 306 在初始化时从配置文件读取信息,这些文件包含关于要由捕捉和变更跟踪表来进行备份的每一数据库的信息。服务 306 随后建立捕捉和变更跟踪表和表群的触发器。服务 306 随后调度两个操作:检查基表模式变更和升级以及清除捕捉和变更跟踪表。

[0047] 为了将一个分区(例如,表群)或行群还原到特定时间点,还原工具 308 调用在备份服务初始化或在模式变更之后安装的一个或多个所存储的过程 410。一个过程还原整个分区中的每一个表。另一过程还原该分区内的行群。这些过程取要还原的所需时间点作为自变量。

[0048] 对于实现事务一致性,在还原(回退)操作期间,如果所需还原点落入所存储(备份)的事务的中间则可能创建不一致的状态。对于对数据库的每一所存储的变更,为事务中的每一操作都存储事务开始 DBTS 和操作 DBTS。

[0049] 展示了返回当前事务中的第一操作的 DBTS 的内部函数 TX_START_DBTS()。如果在该内部函数被调用之前该事务没有执行操作,则递增 @@DBTS 并且将该值记录在事务记录中。

[0050] 对于事务开始时间,观察到如下结果:回退给定事务操作中的一些而留下同一事务的其他操作可能是不合需要的。考虑一系统,在该系统中,在任何给定时间都有许多事务在运行并且这些事务中的每一个都修改不止一行。DBTS 中的每一个值可以处于一些事务的中间。因此,正确答案是对于没有 DBTS 值将回退晚于该值的所有操作。因此,使用操作的 DBTS 作为唯一准则是受限的。

[0051] 该能力被提供给属于已经开始回退的事务的回退操作。为了对其进行支持,事务标识符与记录存储在一起。此外,事务标识符是跨故障唯一的。DBTS 具有这一特性并且被保证在故障中前进。因此,事务开始 DBTS 是这一需求的候选。

[0052] 考虑事务 T1、T2 和 T3 的以下时间线。

[0053]

时间	操作
5	5 :X = 0

10	T1 开始 (做某事)
15	T1 开始 (做某事)
20	T2 将 X 更新成 2
25	T2 提交
30	T1 将 X 更新成 1

[0054]

35	T1 提交
40	T3 开始
41	T3 将 X 更新成 3
42	T3 提交

[0055] 现在,还原到 DBTS = 28。因为事务 T2 那时已经提交,并且事务 T1 尚未提交,所以 X 被还原到 2。然而,为实现这一点,注意,事务 T1 更新发生在事务 T2 更新之后,但事务 T1 开始时间早于事务 T2 开始时间。因此,要解决这一问题不止需要事务开始时间。

[0056] 使用主表和捕捉和变更跟踪表中的各列来优化插入繁重工作负载。所公开的体系结构不涉及在插入时写入第二行(或甚至使触发器执行)。

[0057] 对于配置备份策略,在策略文件中指定关键客户可见的策略参数。

[0058] 保留时间段确定该数据在多长时间内可用于恢复。这基本上管控跟踪数据将被保持在数据捕捉和跟踪表中的持续时间。数据捕捉表中的旧数据可由该系统懒散地删除。

[0059] 回退过程作为回退命令来执行,它取要被回退的行群和要回退所述行群回到的 UTC 时间作为参数。(行群是具有相同的 [分区] 键值的行的集合。)该命令能够将生产数据覆盖到在指定回退时间处存在的状态。命令的变型可以将多个行群回退到某一时间点。也可以用各种其他方式来扩展该命令。

[0060] 图 5 示出示例性捕捉和变更跟踪表 500。表 500(类似于表 118 和 304)存储基表的分区键和类型、事务开始 DBTS(TX START DBTS)、操作 DBTS(OP DBTS)、基于 UTC 的时间戳(用于找出还原点)、基表中的行的主键(作为 XML)、来自该基表行的“旧”值(数据库中的数据在删除或更新操作之前的 XML 版本)以及从中学到值和主键列的基表的名称。在清除方面,捕捉和变更跟踪表 500 的名称可以是期满日期。捕捉和变更跟踪表 500 的创建日期和期满日期可被附加到该表的名称之后,这消除了对额外字段的需求。

[0061] 此处所包括的是一组代表用于执行所公开的体系结构的新颖方面的示例性方法的流程图。尽管出于解释简明的目的,此处例如以流图或流程图形式示出的一个或多个方法被示出并描述为一系列动作,但是可以理解和明白,各方法不受动作的次序的限制,因为根据本发明,某些动作可以按与此处所示并描述的不同的次序和 / 或与其他动作同时发生。例如,本领域技术人员将会明白并理解,方法可被替换地表示为一系列相互关联的状态

或事件,诸如以状态图的形式。此外,并非在一方法中示出的所有动作都是新颖实现所必需的。

[0062] 图 6 示出根据所公开的体系结构的计算机实现的数据管理方法。在 600,接收与分布式数据库中的数据的递增变更数据相关的数据操作。在 602,响应于数据操作来捕捉递增变更数据。在 604,创建与递增变更数据相关联的跟踪信息。在 606,存储递增变更数据和跟踪信息。

[0063] 图 7 示出了图 6 的方法的其他方面。在 700,将递增变更数据作为表中的一行并以包括自描述模式的 XML 格式来存储。在 702,使用结构化查询语言来查询递增变更数据。在 704,将保留策略应用于递增变更数据和跟踪信息。在 706,将事务时间戳和数据操作时间戳与递增变更数据的事务相关联地存储。在 708,基于递增变更数据将回退操作应用于数据库以将数据回退到先前时间点。

[0064] 如在本申请中所使用的,术语“组件”和“系统”旨在表示计算机相关的实体,其可以是硬件、硬件和软件的组合、软件、或者执行中的软件。例如,组件可以是,但不限于,诸如处理器、芯片存储器、大容量存储设备(例如,光驱、固态驱动器、和 / 或磁存储介质驱动器)、以及计算机等有形组件,以及诸如运行在处理器上的进程、对象、可执行码、模块、执行的线程和 / 或程序等软件组件。作为说明,在服务器上运行的应用程序和服务器两者都可以是组件。一个或多个组件可以驻留在进程和 / 或执行的线程内,且组件可以位于一台计算机上和 / 或分布在两台或更多的计算机之间。词语“示例性”此处可用于表示用作示例、实例或说明。在此被描述为“示例性”的任何方面或设计并不一定要被解释为相比其他方面或设计更优选或有利。

[0065] 现在参考图 8,示出可用于根据所公开的体系结构来使用镜像之前捕捉执行逻辑数据备份和回退的计算系统 800 的框图。为了提供用于其各方面的附加上下文,图 8 及以下讨论旨在提供对其中可实现各方面的合适的计算系统 800 的简要概括描述。尽管以上描述是在可在一或多个计算机上运行的计算机可执行指令的一般上下文中进行的,但是本领域的技术人员将认识到,新颖实施例也可结合其他程序模块和 / 或作为硬件和软件的组合来实现。

[0066] 用于实现各方面的计算系统 800 包括计算机 802,其具有处理单元 804、诸如系统存储器 806 等的计算机可读存储、以及系统总线 808。处理单元 804 可以是各种市场上可买到的处理器中的任一种,包括单处理器、多处理器、单核单元以及多核单元。此外,本领域的技术人员可以理解,各新颖方法可用其他计算机系统配置来实施,包括小型机、大型计算机、以及个人计算机(例如、台式、膝上型等)、手持式计算设备、基于微处理器的或可编程消费电子产品等,其每一个都可在操作上耦合到一个或多个相关联的设备。

[0067] 系统存储器 806 可包括计算机可读存储,如易失性(VOL)存储器 810(例如,随机存取存储器(RAM))和非易失性存储器(NON-VOL)812(如 ROM、EPROM、EEPROM 等)。基本输入 / 输出系统(BIOS)可被存储在非易失性存储器 812 中,并且包括诸如在启动期间便于在计算机 802 内的组件之间传递数据和信号的基本例程。易失性存储器 810 还可包括诸如静态 RAM 等高速 RAM 来用于高速缓存数据。

[0068] 系统总线 808 提供了用于包括,但不限于系统存储器 806 的系统组件到处理单元 804 的接口。系统总线 808 可以是若干种总线结构中的任一种,这些总线结构还可使用各类

可购买到的总线体系结构中的任一种互连到存储器总线（带有或没有存储器控制器）以及外围总线（例如，PCI、PCIe、AGP、LPC 等）。

[0069] 计算机 802 还包括用于机器可读存储子系统 814 以及将存储子系统 814 接口到系统总线 808 和其他所需计算机组件的存储接口 816。存储子系统 814 可包括例如硬盘驱动器 (HDD)、磁软盘驱动器 (FDD) 和 / 或光盘存储驱动器（例如，CD-ROM 驱动器、DVD 驱动器）中的一种或多种。存储接口 816 可包括诸如，例如 EIDE、ATA、SATA 和 IEEE 1394 等接口技术。

[0070] 一个或多个程序和数据可被存储在存储器子系统 806、机器可读和可移动存储器子系统 818（例如，闪存驱动器形状因子技术）和 / 或存储子系统 814（例如，光、磁、固态）中，包括操作系统 820、一个或多个应用程序 822、其他程序模块 824 以及程序数据 826。

[0071] 一个或多个应用程序 822、其他程序模块 824 以及程序数据 826 可包括例如图 1 的系统 100 的实体和组件、图 2 的系统 200 的实体和组件、图 3 的系统 300 的实体和组件、图 4 的系统 400 的实体和组件、表 500、以及图 6-7 的流程图所表示的方法。

[0072] 一般而言，程序包括执行特定任务或实现特定抽象数据类型的例程、方法、数据结构、其他软件组件等等。操作系统 820、应用程序 822、模块 824 和 / 或数据 826 的全部或部分也可被高速缓存在诸如易失性存储器 810 等存储器中。应该明白，所公开的体系结构可以用各种市场上可购得的操作系统或操作系统的组合（例如，作为虚拟机）来实施。

[0073] 存储子系统 814 和存储器子系统 (806 和 818) 用作用于数据、数据结构、计算机可执行指令等的易失性和非易失性存储的计算机可读介质。计算机可读介质可以是可由计算机 802 访问的任何可用介质，且包括可移动和不可移动的易失性和非易失性、内部和 / 或外部介质。对于计算机 802，介质以任意适当的数字格式容纳数据的存储。本领域的技术人员应当理解，可使用其他类型的计算机可读介质，如 zip 驱动器、磁带、闪存卡、闪存驱动器、磁带盒等来存储用于执行所公开的体系结构的新颖方法的计算机可执行指令。

[0074] 用户可以使用诸如键盘和鼠标等外部用户输入设备 828 来与计算机 802、程序和数据交互。其他外部用户输入设备 828 可包括话筒、IR (红外) 遥控器、操纵杆、游戏手柄、照相机识别系统、指示笔、触摸屏、姿势系统（例如，眼移动、头移动等）和 / 或类似物。在计算机 802 是例如便携式计算机的情况下，用户可以使用诸如触摸垫、话筒、键盘等板载用户输入设备 830 来与计算机 802、程序和数据交互。这些和其他输入设备通过输入 / 输出 (I/O) 设备接口 832 经由系统总线 808 连接到处理单元 804，但也可通过其它接口连接，如并行端口、IEEE 1394 串行端口、游戏端口、USB 端口、IR 接口等等。I/O 设备接口 832 还便于使用输出外围设备 834，如打印机、音频设备、照相机设备等，如声卡和 / 或板载音频处理能力。

[0075] 一个或多个图形接口 836（通常也称为图形处理单元 (GPU)）提供计算机 802 和外部显示器 838（例如，LCD、等离子）和 / 或板载显示器 840（例如，对于便携式计算机）之间的图形和视频信号。图形接口 836 也可作为计算机系统板的一部分来制造。

[0076] 计算机 802 可以使用经由有线 / 无线通信子系统 842 到一个或多个网络和 / 或其他计算机的逻辑连接在联网环境（例如，基于 IP 的）中操作。其他计算机可包括工作站、服务器、路由器、个人计算机、基于微处理器的娱乐设备、对等设备或其他常见的网络节点，并且通常包括以上相对于计算机 802 描述的许多或所有元件。逻辑连接可包括到局域网

(LAN)、广域网 (WAN) 热点等的有线 / 无线连接。LAN 和 WAN 联网环境常见于办公室和公司，并且方便了诸如内联网等企业范围计算机网络，所有这些都可连接到例如因特网等全球通信网络。

[0077] 当在联网环境中使用时，计算机 802 经由有线 / 无线通信子系统 842（例如，网络接口适配器、板载收发机子系统等）连接到网络来与有线 / 无线网络、有线 / 无线打印机、有线 / 无线输入设备 844 等通信。计算机 802 可包括用于通过网络建立通信的调制解调器或其他装置。在联网环境中，相对于计算机 802 的程序和数据可被存储在远程存储器 / 存储设备中，如与分布式系统相关联。应该理解，所示网络连接是示例性的，并且可以使用在计算机之间建立通信链路的其他手段。

[0078] 计算机 802 可用于使用诸如 IEEE 802. xx 标准家族等无线电技术来与有线 / 无线设备或实体通信，例如在操作上安置在与例如打印机、扫描仪、台式和 / 或便携式计算机、个人数字助理 (PDA)、通信卫星、任何一件与无线可检测标签相关联的设备或位置（例如，电话亭、报亭、休息室）以及电话的无线通信（例如，IEEE 802.11 空中调制技术）中的无线设备。这至少包括用于热点的 Wi-Fi（即无线保真）、WiMax 和蓝牙™ 无线技术。由此，通信可以是如对于常规网络那样的预定义结构，或者仅仅是至少两个设备之间的自组织 (ad hoc) 通信。Wi-Fi 网络使用称为 IEEE 802.11x(a,b,g 等等) 的无线电技术来提供安全、可靠、快速的无线连接。Wi-Fi 网络可用于将计算机彼此连接、连接到因特网以及连接到有线网络（使用 IEEE802.3 相关介质和功能）。

[0079] 现在参考图 9，示出使用镜像之前捕捉来执行逻辑数据备份和回退的计算环境 900 的示意框图。环境 900 包括一个或多个客户机 902。客户机 902 可以是硬件和 / 或软件（例如，线程、进程、计算设备）。例如，客户机 902 可容纳 cookie 和 / 或相关联的上下文信息。

[0080] 环境 900 还包括一个或多个服务器 904。服务器 904 也可以是硬件和 / 或软件（例如，线程、进程、计算设备）。服务器 904 可以例如通过使用本体系结构来容纳线程以执行变换。客户机 902 和服务器 904 之间的一种可能的通信可以是适用于在两个或更多计算机进程之间传输的数据包的形式。数据分组可包括例如 cookie 和 / 或相关联的上下文信息。环境 900 包括可以用来使客户机 902 和服务器 904 之间通信更容易的通信框架 906（例如，诸如因特网等全球通信网络）。

[0081] 通信可经由有线（包括光纤）和 / 或无线技术来促进。客户机 902 操作上被连接到可以用来存储对客户机 902 本地的信息（例如，cookie 和 / 或相关联的上下文信息）的一个或多个客户机数据存储 908。同样地，服务器 904 可在操作上连接到可以用来存储对服务器 904 本地的信息的一个或多个服务器数据存储 910。

[0082] 上面描述的包括所公开的体系结构的各示例。当然，描述每一个可以想到的组件和 / 或方法的组合是不可能的，但本领域内的普通技术人员应该认识到，许多其他组合和排列都是可能的。因此，该新颖体系结构旨在涵盖所有这些落入所附权利要求书的精神和范围内的更改、修改和变化。此外，就在说明书或权利要求书中使用术语“包括”而言，这一术语旨在以与术语“包含”在被用作权利要求书中的过渡词时所解释的相似的方式为包含性的。

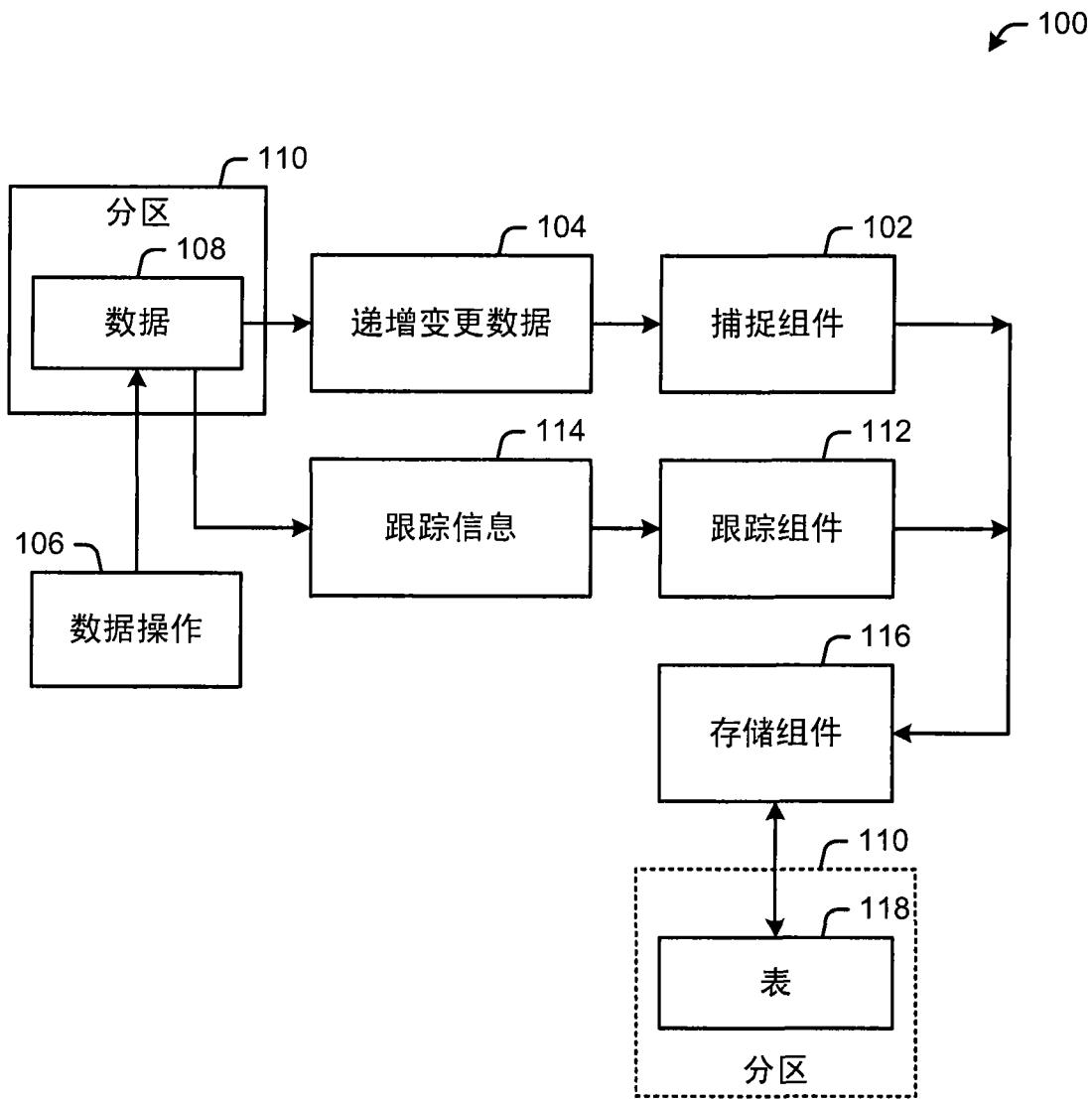


图 1

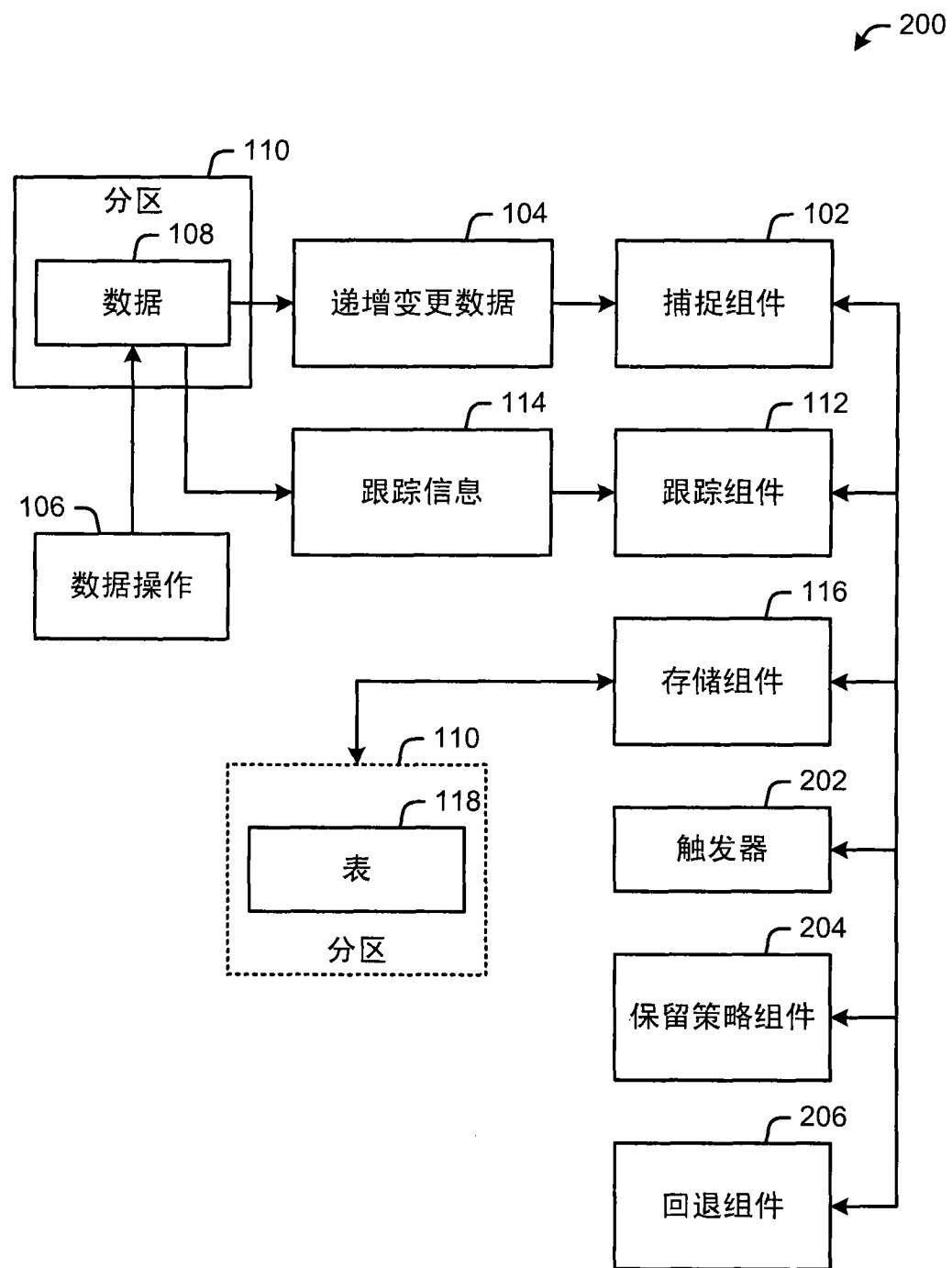


图 2

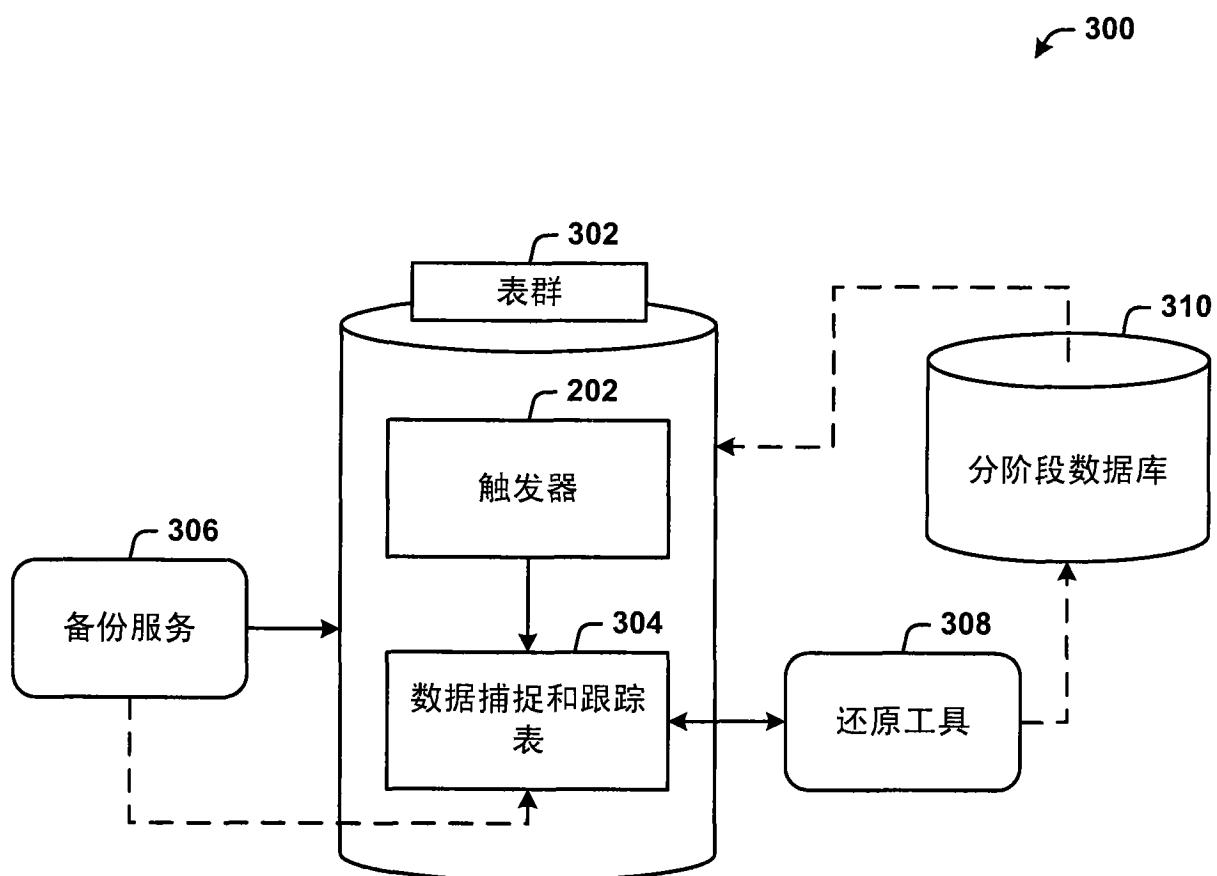


图 3

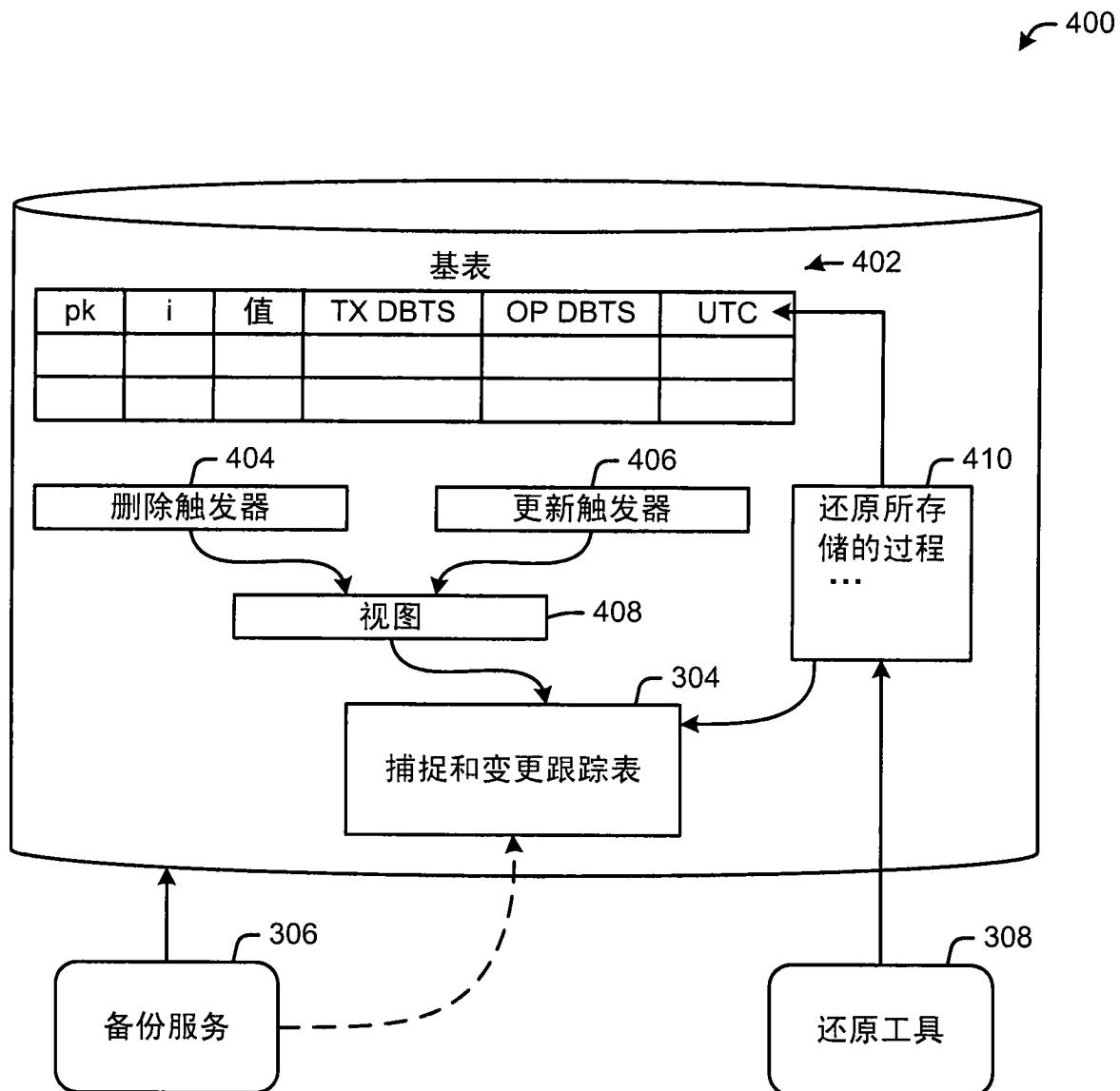


图 4

500 ↙

分区键	TX开始 DBTS	OP DBTS	UTC	主键	值	表名称

图 5

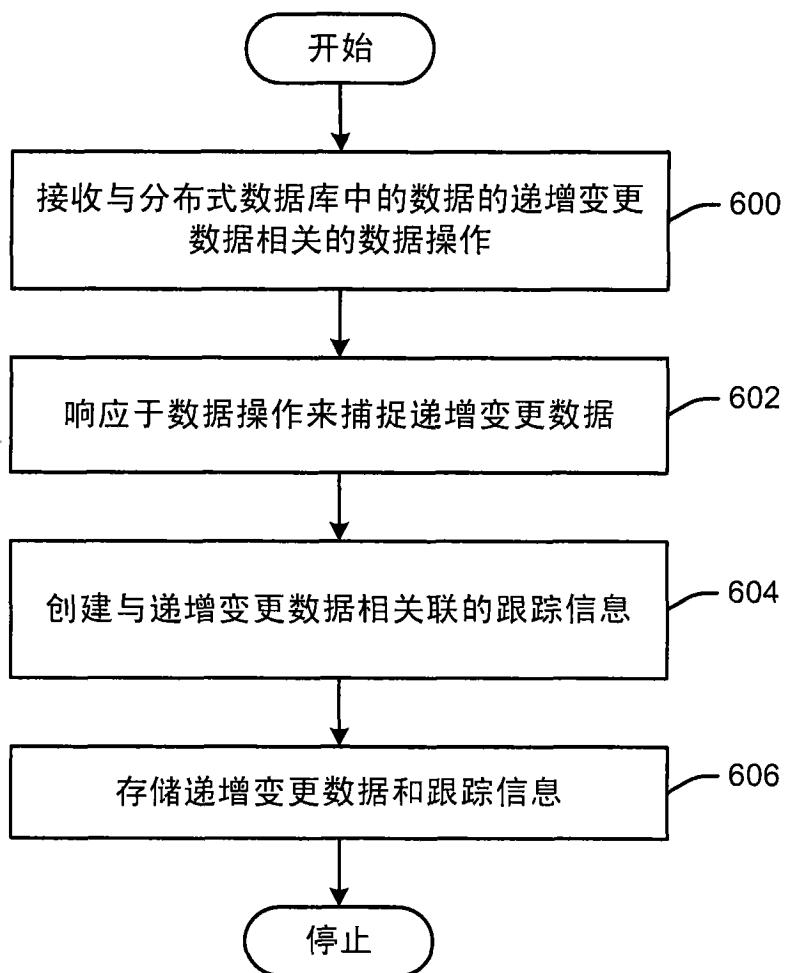


图 6

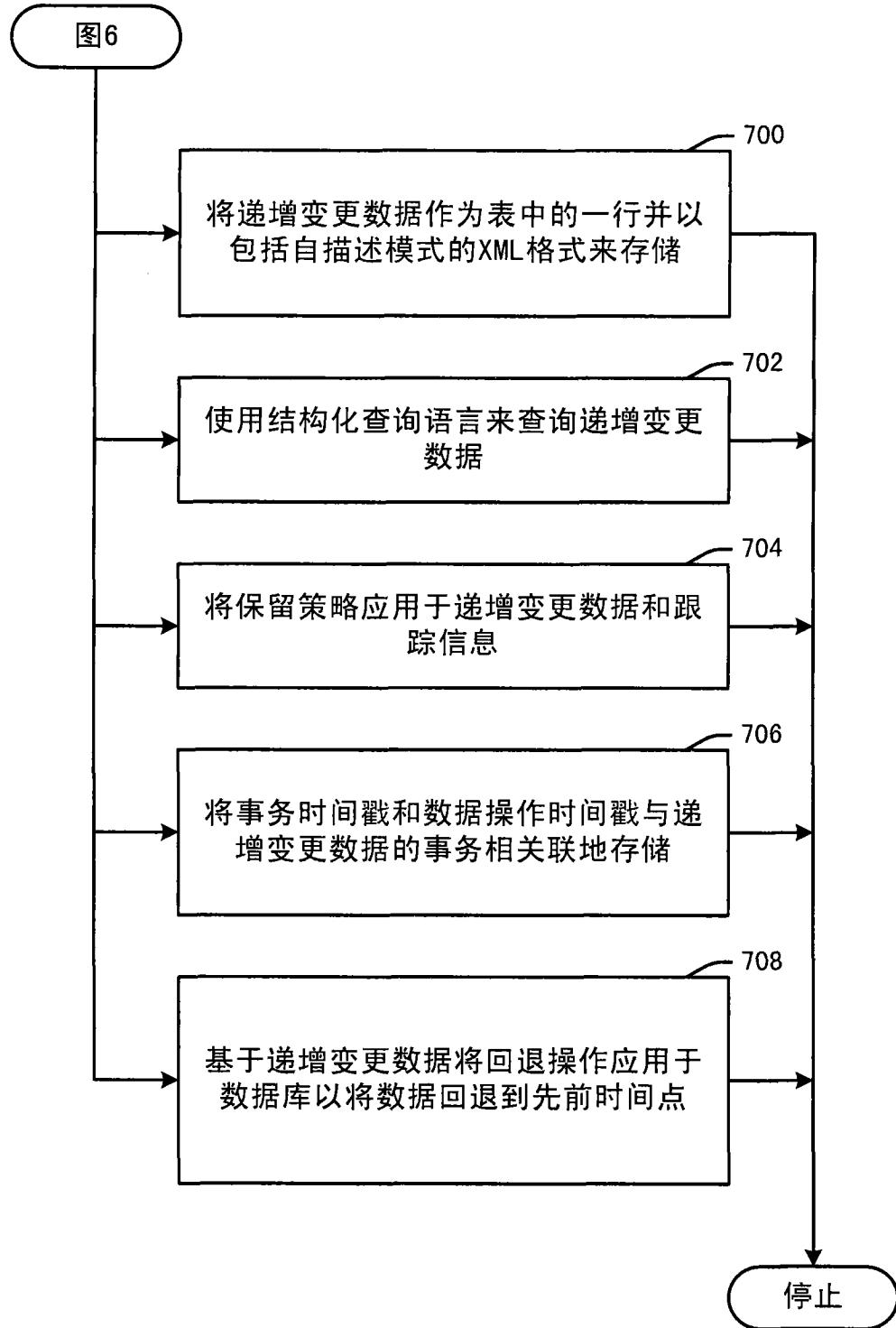


图 7

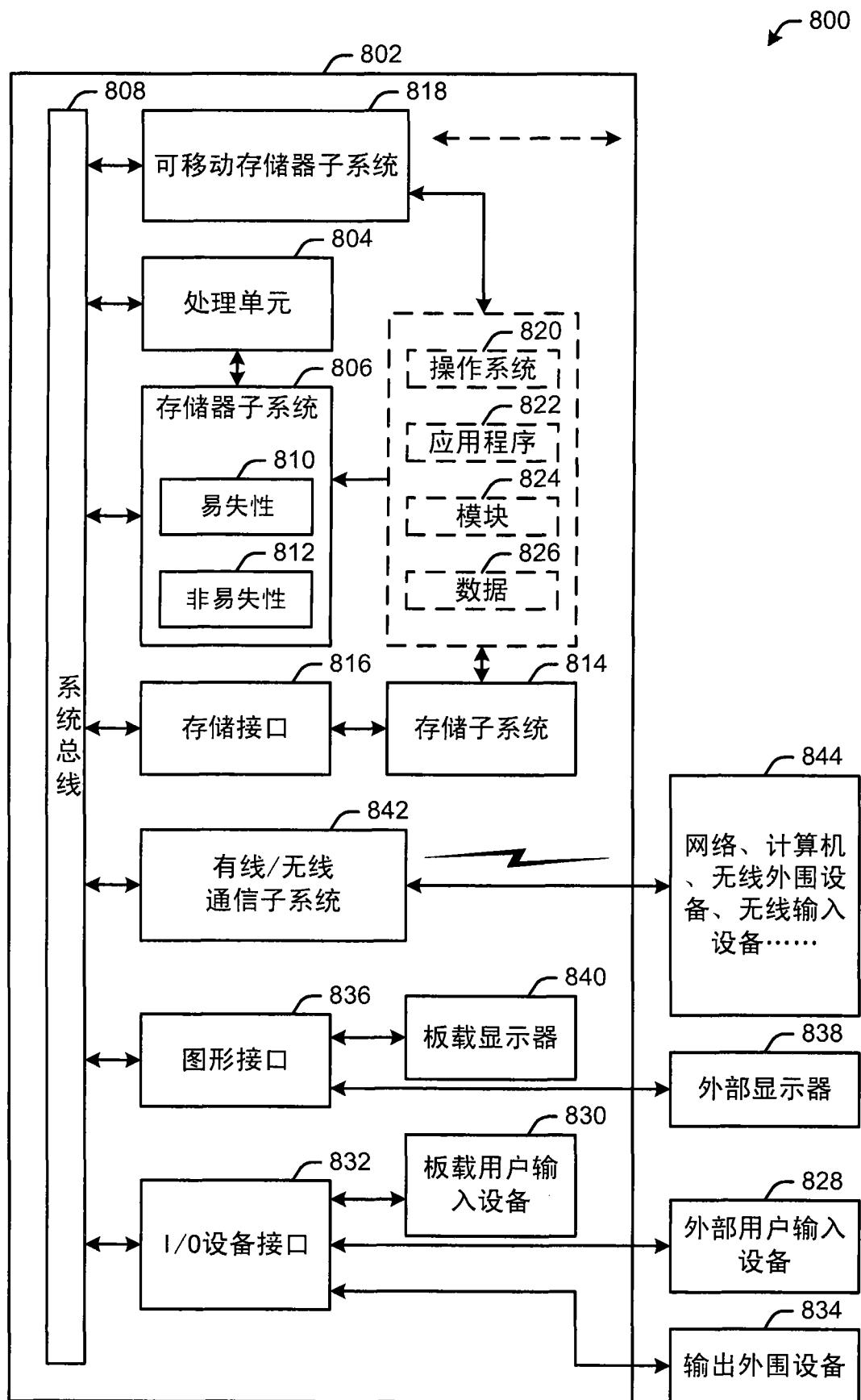


图 8

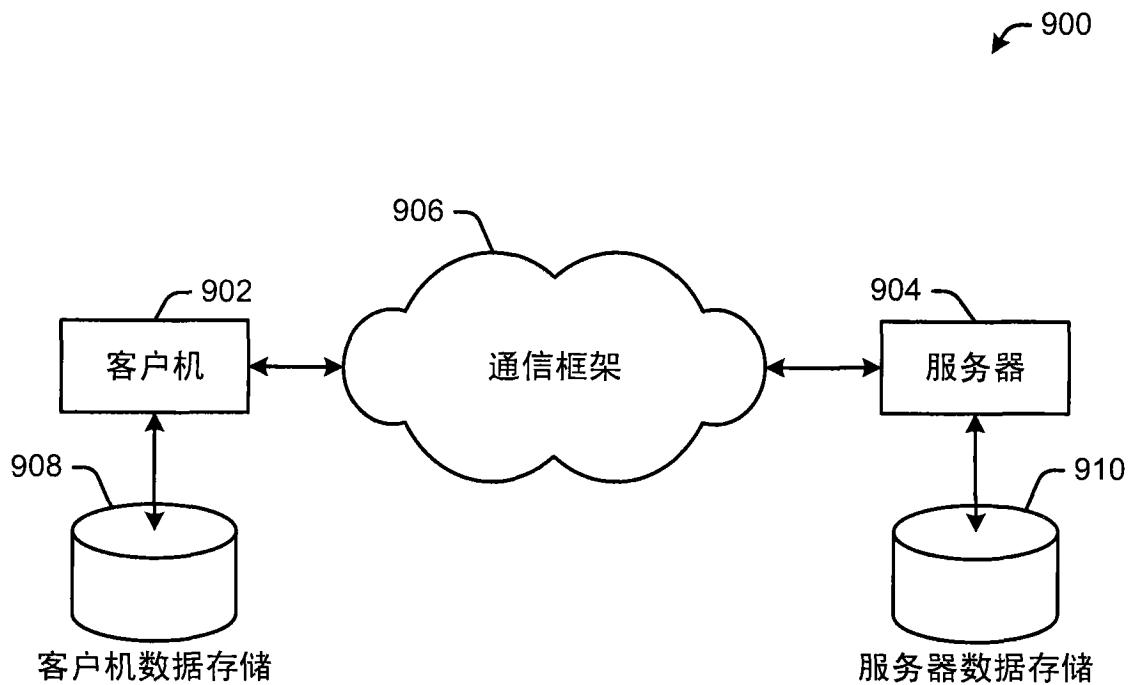


图 9