(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 109690525 B
(45)授权公告日 2023.05.30

(21)申请号 201780055809.6
(22)申请日 2017.09.15
(65)同一申请的已公布的文献号
申请公布号 CN 109690525 A
(43)申请公布日 2019.04.26
(30)优先权数据
62/395,204 2016.09.15 US
(85)PCT国际申请进入国家阶段日
2019.03.12
(86)PCT国际申请的申请数据
PCT/US2017/051868 2017.09.15
(87)PCT国际申请的公布数据
WO2018/053320 EN 2018.03.22
(73)专利权人 甲骨文国际公司
地址 美国加利福尼亚
(72)发明人 H•帕克 S•比施诺伊 P•图卡拉姆 I•施米莱瓦
(74)专利代理机构 中国贸促会专利商标事务所有限公司 11038
专利代理师 周磊
(51)Int.Cl.
G06F 16/2455(2019.01)
G06F 16/29(2019.01)
H04W 4/021(2018.01)
(56)对比文件
US 2015148077 A1,2015.05.28
US 9298788 B1,2016.03.29
CN 102067130 A,2011.05.18
王钢林等.一种Cartesian气动网格的自适应划分算法.《应用力学学报》.2008,(第02期),102-106+187.
审查员 李天浩

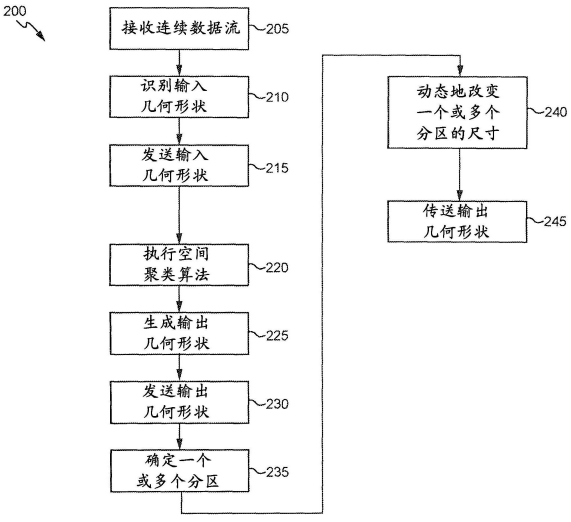
权利要求书3页 说明书28页 附图13页

(54)发明名称

形状的流数据的自动分区

(57)摘要

公开了用于处理事件流中的事件的事件处理系统。该系统可以执行指令以接收与应用相关的连续数据流,识别与连续数据流相关联的输入几何形状,至少部分地基于输入几何形状生成几何形状聚类,至少部分地基于几何形状聚类和几何形状聚类中的每个聚类中的几何形状的数量来生成输出几何形状,基于几何形状聚类和几何形状聚类中的每个聚类中的几何形状的数量确定输出几何形状的一个或多个分区,动态地改变输出几何形状的一个或多个分区的尺寸,以及传送与连续数据流相关联的输出几何形状。



CN 109690525 B

1.一种用于动态地确定分区的尺寸的方法,包括:

由计算设备接收与应用相关的连续数据流;

由计算设备识别与连续数据流相关联的输入几何形状,所述输入几何形状指定要被分区的连续数据流中的区域;

由计算设备至少部分地基于所述输入几何形状生成几何形状聚类;

由计算设备至少部分地基于所述几何形状聚类和所述几何形状聚类中的每个聚类中的几何形状的数量来生成输出几何形状;

由计算设备基于所述几何形状聚类和所述几何形状聚类中的每个聚类中的几何形状的数量来确定所述输出几何形状的一个或多个分区;

由计算设备检测所述几何形状聚类中的变化,其中检测所述变化包括:

将所述几何形状聚类的边界框加载到索引;

比较所述几何形状聚类与索引内的其他几何形状聚类;

确定所述几何形状聚类是否已从索引内的所述其他几何形状聚类改变;

响应于从索引中没有找到所述几何形状聚类,做出所述几何形状聚类是新聚类的确定,并且在没有来自所述其他几何形状聚类的一个或多个参数的情况下,基于所述几何形状聚类和所述几何形状聚类中的每个聚类中的几何形状的数量来确定所述输出几何形状的一个或多个分区;以及

响应于从索引中找到所述几何形状聚类,做出所述几何形状聚类是现有聚类的确定,并且识别尺寸阈值以动态地改变所述输出几何形状的所述一个或多个分区的尺寸;

由计算设备动态地改变所述输出几何形状的所述一个或多个分区的尺寸;以及

由计算设备传送与连续数据流相关联的所述输出几何形状。

2.如权利要求1所述的方法,其中,改变所述一个或多个分区的尺寸包括确定网格范围,创建到分区索引表的网格索引,以及将到分区索引表的网格索引设置为存储在索引中的聚类对象,使得所述聚类对象可以用在分配分区标识符中。

3.如权利要求2所述的方法,其中,网格范围的决定由配置参数、网格尺寸、网格尺寸影响因子以及所述几何形状聚类中的每个聚类中的几何形状的数量来实现。

4.一种用于动态地确定分区的尺寸的系统,包括:

存储器,被配置为存储计算机可执行指令;以及

处理器,被配置为访问存储器并执行计算机可执行指令,以:

接收与应用相关的连续数据流;

识别与连续数据流相关联的输入几何形状,所述输入几何形状指定要被分区的连续数据流中的区域;

至少部分地基于所述输入几何形状生成几何形状聚类;

至少部分地基于所述几何形状聚类和所述几何形状聚类中的每个聚类中的几何形状的数量来生成输出几何形状;

基于所述几何形状聚类和所述几何形状聚类中的每个聚类中的几何形状的数量来确定所述输出几何形状的一个或多个分区;

检测所述几何形状聚类中的变化,其中检测所述变化包括:

将所述几何形状聚类的边界框加载到索引;

比较所述几何形状聚类与索引内的其他几何形状聚类；

确定所述几何形状聚类是否已从索引内的所述其他几何形状聚类改变；

响应于从索引中没有找到所述几何形状聚类，做出所述几何形状聚类是新聚类的确定，并且在没有来自所述其他几何形状聚类的一个或多个参数的情况下，基于所述几何形状聚类和所述几何形状聚类中的每个聚类中的几何形状的数量来确定所述输出几何形状的一个或多个分区；以及

响应于从索引中找到所述几何形状聚类，做出所述几何形状聚类是现有聚类的确定，并且识别尺寸阈值以动态地改变所述输出几何形状的所述一个或多个分区的尺寸；

动态地改变所述输出几何形状的所述一个或多个分区的尺寸；以及

传送与连续数据流相关联的所述输出几何形状。

5.如权利要求4所述的系统，其中，改变所述一个或多个分区的尺寸包括确定网格范围，创建到分区索引表的网格索引，以及将到分区索引表的网格索引设置为存储在索引中的聚类对象，使得所述聚类对象可以用在分配分区标识符中。

6.如权利要求5所述的系统，其中，网格范围的决定由配置参数、网格尺寸、网格尺寸影响因子以及所述几何形状聚类中的每个聚类中的几何形状的数量来实现。

7.一种存储计算机可执行代码的计算机可读介质，当所述计算机可执行代码由处理器执行时使所述处理器执行包括以下的操作：

接收与应用相关的连续数据流；

识别与连续数据流相关联的输入几何形状，所述输入几何形状指定要被分区的连续数据流中的区域；

至少部分地基于所述输入几何形状生成几何形状聚类；

至少部分地基于所述几何形状聚类和所述几何形状聚类中的每个聚类中的几何形状的数量来生成输出几何形状；

基于所述几何形状聚类和所述几何形状聚类中的每个聚类中的几何形状的数量来确定所述输出几何形状的一个或多个分区；

检测所述几何形状聚类中的变化，其中检测所述变化包括：

将所述几何形状聚类的边界框加载到索引；

比较所述几何形状聚类与索引内的其他几何形状聚类；

确定所述几何形状聚类是否已从索引内的所述其他几何形状聚类改变；

响应于从索引中没有找到所述几何形状聚类，做出所述几何形状聚类是新聚类的确定，并且在没有来自所述其他几何形状聚类的一个或多个参数的情况下，基于所述几何形状聚类和所述几何形状聚类中的每个聚类中的几何形状的数量来确定所述输出几何形状的一个或多个分区；以及

响应于从索引中找到所述几何形状聚类，做出所述几何形状聚类是现有聚类的确定，并且识别尺寸阈值以动态地改变所述输出几何形状的所述一个或多个分区的尺寸；

动态地改变所述输出几何形状的所述一个或多个分区的尺寸；以及

传送与连续数据流相关联的所述输出几何形状。

8.如权利要求7所述的计算机可读介质，其中，改变所述一个或多个分区的尺寸包括确定网格范围，创建到分区索引表的网格索引，以及将到分区索引表的网格索引设置为存储

在索引中的聚类对象，使得所述聚类对象可以用在分配分区标识符中。

9.如权利要求8所述的计算机可读介质，其中，网格范围的决定有配置参数、网格尺寸、网格尺寸影响因子以及所述几何形状聚类中的每个聚类中的几何形状的数量来实现。

10.一种用于动态地确定分区的尺寸的装置，包括用于执行权利要求1-3中任一项所述的方法的操作的单元。

形状的流数据的自动分区

[0001] 相关申请的交叉引用

[0002] 本申请要求于2016年9月15日提交的标题为“AUTOMATIC PARALLELIZATION FOR GEOMENCE APPLICATIONS”的美国临时申请No.62/395,204的优先权和权益,其全部内容通过引用并入本文,用于所有目的。

背景技术

[0003] 在传统的数据库系统中,数据通常以表的形式存储在一个或多个数据库中。然后使用诸如结构化查询语言(SQL)之类的数据管理语言来查询和操纵所存储的数据。例如,可以定义并执行SQL查询,以从存储在数据库中的数据中识别相关数据。因此,在存储在数据库中的有限数据集上执行SQL查询。另外,当执行SQL查询时,它在有限数据集上被执行一次并产生有限静态结果。因此,数据库被进行最佳装备以在有限存储数据集上运行查询。

[0004] 但是,许多现代应用和系统以连续数据或事件流而不是有限数据集的形式生成数据。此类应用的示例包括但不限于传感器数据应用、金融报价机、网络性能测量工具(例如,网络监视和流量管理应用)、点击流分析工具、汽车交通监视等。此类应用产生了对可以处理数据流的新型应用的需求。例如,温度传感器可以被配置为发出温度读数。

[0005] 管理和处理用于这些类型的基于事件流的应用的数据涉及以强时间焦点构建数据管理和查询能力。需要不同种类的查询机制,该查询机制包括对连续无界数据集的长时间运行的查询。虽然一些供应商现在提供面向事件流处理的产品套件,但这些产品提供仍然缺乏处置当今事件处理需求所需的处理灵活性。

发明内容

[0006] 提供了用于处理事件流的事件的技术(例如,方法、系统、存储可由一个或多个处理器执行的代码或指令的非瞬态计算机可读介质)。一个或多个计算机的系统可以被配置为借助于在系统上安装软件、固件、硬件或它们的组合来执行特定的操作或动作,这些软件、固件、硬件或它们的组合在操作中使系统执行动作。一个或多个计算机程序可以被配置为借助于包括当由数据处理装置执行时使装置执行动作的指令来执行特定的操作或动作。一个一般方面包括一种用于动态确定分区的尺寸的方法,包括:由计算设备接收与应用相关的连续数据流。该方法还包括由计算设备识别与连续数据流相关联的输入几何形状,输入几何形状指定要被分区的连续数据流中的区域。该方法还包括由计算设备至少部分地基于输入几何形状生成几何形状聚类。该方法还包括由计算设备至少部分地基于几何形状聚类和几何形状聚类中的每个聚类中的几何形状的数量来生成输出几何形状。该方法还包括由计算设备基于几何形状聚类和几何形状聚类中的每个聚类中的几何形状的数量来确定输出几何形状的一个或多个分区。该方法还包括由计算设备动态地改变输出几何形状的一个或多个分区的尺寸。该方法还包括由计算设备传送与连续数据流相关联的输出几何形状。该方面的其它实施例包括对应计算机系统、装置和记录在一个或多个计算机存储设备上的计算机程序,计算机系统、装置和计算机程序中的每一个被配置为执行该方法的动作。

[0007] 实现可以包括以下特征中的一个或多个。其中确定输出几何形状的一个或多个分区的方法包括检测几何形状聚类中的变化。其中检测几何形状聚类中的变化的方法包括：将几何形状聚类的边界框加载到索引；比较几何形状聚类与索引内的其他几何形状聚类；以及确定该几何形状聚类是否已从索引内的所述其他几何形状聚类变化。其中检测几何形状聚类中的变化的方法还包括：当从索引中没有找到几何形状聚类时，进行几何形状聚类是新聚类的确定；而当从索引中找到几何形状聚类时，进行几何形状聚类是现有聚类的确定。其中检测几何形状聚类中的变化的方法还包括：当几何形状聚类是新聚类时，在没有来自其他几何形状聚类的一个或多个参数的情况下，基于几何形状聚类和几何形状聚类中的每个聚类中的几何形状的数量来确定输出几何形状的一个或多个分区；而当几何形状聚类是现有聚类时，识别尺寸阈值以动态地改变输出几何形状的一个或多个分区的尺寸。其中改变一个或多个分区的尺寸的方法包括确定网格范围，创建到分区索引表的网格索引，以及将到分区索引表的网格索引设置为存储在索引中的聚类对象，使得该聚类对象可以用在分配分区标识符中。其中决定网格范围的方法通过配置参数、网格尺寸、网格尺寸影响因子以及几何形状聚类中的每个聚类中的几何形状的数量来实现。所描述的技术的实现可以包括硬件、方法或处理、或计算机可访问介质上的计算机软件。

[0008] 一个一般方面包括一种系统，该系统包括：存储器，被配置为存储计算机可执行指令；以及处理器，被配置为访问存储器并执行计算机可执行指令。该系统还包括接收与应用相关的连续数据流。该系统还包括识别与连续数据流相关联的输入几何形状，输入几何形状指定要被分区的连续数据流中的区域。该系统还包括至少部分地基于输入几何形状生成几何形状聚类。该系统还包括至少部分地基于几何形状聚类和几何形状聚类中的每个聚类中的几何形状的数量来生成输出几何形状。该系统还包括基于几何形状聚类和几何形状聚类中的每个聚类中的几何形状的数量来确定输出几何形状的一个或多个分区。该系统还包括动态地改变输出几何形状的一个或多个分区的尺寸。该系统还包括传送与连续数据流相关联的输出几何形状。该方面的其它实施例包括对应计算机系统、装置和记录在一个或多个计算机存储设备上的计算机程序，计算机系统、装置和计算机程序中的每一个被配置为执行该方法的动作。

[0009] 实现可以包括以下特征中的一个或多个。其中确定输出几何形状的一个或多个分区的系统包括检测几何形状聚类中的变化。其中检测几何形状聚类中的变化的系统包括：将几何形状聚类的边界框加载到索引；比较几何形状聚类与索引内的其他几何形状聚类；以及确定几何形状聚类是否已从索引内的所述其他几何形状聚类变化。其中检测几何形状聚类中的变化的系统还包括：当从索引中没有找到几何形状聚类时，进行几何形状聚类是新聚类的确定；而当从索引中找到几何形状聚类时，进行几何形状聚类是现有聚类的确定。其中检测几何形状聚类中的变化的系统还包括：当几何形状聚类是新聚类时，在没有来自其他几何形状聚类的一个或多个参数的情况下，基于几何形状聚类和几何形状聚类中的每个聚类中的几何形状数量来确定输出几何形状的一个或多个分区；而当几何形状聚类是现有聚类时，识别尺寸阈值以动态地改变输出几何形状的一个或多个分区的尺寸。其中改变一个或多个分区的尺寸的系统包括确定网格范围，创建到分区索引表的网格索引，以及将到分区索引表的网格索引设置为存储在索引中的聚类对象，使得该聚类对象可以用在分配分区标识符中。其中决定网格范围的系统通过配置参数、网格尺寸、网格尺寸影响因子以及

几何形状聚类中的每个聚类中的几何形状的数量来实现。所描述的技术的实现可以包括硬件、方法或处理、或计算机可访问介质上的计算机软件。

[0010] 一个一般方面包括存储计算机可执行代码的计算机可读介质，当所述计算机可执行代码由处理器执行时使处理器执行包括以下的操作：接收与应用相关的连续数据流。该计算机可读介质还包括识别与连续数据流相关联的输入几何形状，输入几何形状指定要被分区的连续数据流中的区域。该计算机可读介质还包括至少部分地基于输入几何形状生成几何形状聚类。该计算机可读介质还包括至少部分地基于几何形状聚类和几何形状聚类中的每个聚类中的几何形状的数量来生成输出几何形状。该计算机可读介质还包括基于几何形状聚类和几何形状聚类中的每个聚类中的几何形状的数量来确定输出几何形状的一个或多个分区。该计算机可读介质还包括动态地改变输出几何形状的一个或多个分区的尺寸。该计算机可读介质还包括传送与连续数据流相关联的输出几何形状。该方面的其它实施例包括对应计算机系统、装置和记录在一个或多个计算机存储设备上的计算机程序，计算机系统、装置和计算机程序中的每一个被配置为执行该方法的动作。

[0011] 实现可以包括以下特征中的一个或多个。其中确定输出几何形状的一个或多个分区的计算机可读介质包括检测几何形状聚类中的变化。其中检测几何形状聚类中的变化的计算机可读介质包括：将几何形状聚类的边界框加载到索引；比较几何形状聚类与索引内的其他几何形状聚类；以及确定几何形状聚类是否已从索引内的所述其他几何形状聚类变化。其中检测几何形状聚类中的变化的计算机可读介质还包括：当几何形状聚类是新聚类时，在没有来自其他几何形状聚类的一个或多个参数的情况下，基于几何形状聚类和几何形状聚类中的每个聚类中的几何形状的数量确定输出几何形状的一个或多个分区；而当几何形状聚类是现有聚类时，识别尺寸阈值以动态地改变输出几何形状的一个或多个分区的尺寸。其中改变一个或多个分区的尺寸的计算机可读介质包括确定网格范围，创建到分区索引表的网格索引，以及将到分区索引表的网格索引设置为存储在索引中的聚类对象，使得该聚类对象可以用在分配分区标识符中。其中决定网格范围的计算机可读介质通过配置参数、网格尺寸、网格尺寸影响因子以及几何形状聚类中的每个聚类中的几何形状的数量来实现。所描述的技术的实现可以包括硬件、方法或处理、或计算机可访问介质上的计算机软件。

[0012] 以上和以下描述的技术可以以多种方式并在许多上下文中实现。参考以下附图提供若干示例实现和上下文，如下面更详细地描述的。但是，以下实现和上下文只是其中的一小部分。

附图说明

[0013] 图1描绘根据本公开的实施例的示例事件处理系统架构的各方面，该示例事件处理系统架构提供可以通过其针对不同的执行环境处理事件处理应用的环境。

[0014] 图2是根据本公开的实施例的总体工作流程的示例流程图。

[0015] 图3是根据本公开的实施例的聚类变化检测处理的示例流程图。

[0016] 图4是根据本公开的实施例的聚类移除处理、分区的改变处理以及分区标识符的分配处理的示例流程图。

[0017] 图5是根据本公开的实施例的输入几何形状、索引和网格的图示。

[0018] 图6和图7是根据本公开的实施例的用于执行流中移动对象的接近度检测和检查的技术的图示。

[0019] 图8是根据本公开的实施例的其中可以实现用于执行流中移动对象的接近度检测和检查以及检查和设置操作的技术的示例系统或架构的图示。

[0020] 图9是根据本公开的实施例的事件处理系统的简化高级图的图示。

[0021] 图10是根据本公开的实施例的事件处理的示例流程图。

[0022] 图11描绘用于实现本公开的实施例的分布式系统的简化图。

[0023] 图12是根据本公开的实施例的系统环境的一个或多个部件的简化框图，通过该系统环境，由实施例系统的一个或多个部件提供的服务可以作为云服务提供。

[0024] 图13图示了可以用于实现本公开的实施例的示例计算机系统。

具体实施方式

[0025] 在以下描述中，将描述各种实施例。出于解释的目的，阐述具体配置和细节以便提供对实施例的透彻理解。但是，对于本领域技术人员来说显而易见的是，可以在没有具体细节的情况下实践这些实施例。此外，可以省略或简化众所周知的特征，以免模糊所描述的实施例。

[0026] 复杂事件处理(CEP)概述

[0027] 复杂事件处理(CEP)提供用于基于事件驱动的架构来构建应用的模块化平台。CEP平台的核心是连续查询语言(CQL)，这允许应用使用声明性的类似SQL的语言对数据流过滤、查询和执行模式匹配操作。开发者可以将CQL与轻量级Java编程模型结合使用来编写应用。其它平台模块包括特征丰富的IDE、管理控制台、聚类、分布式高速缓存、事件储存库和监视等。

[0028] 随着事件驱动的架构和复杂事件处理已变成企业计算领域的突出特征，越来越多的企业已开始使用CEP技术来构建关键任务应用。如今，关键任务CEP应用可以在许多不同的行业中找到。例如，CEP技术在电力行业中使用，以通过允许它们立即对电力需求的变化做出反应而使公用事业更高效。CEP技术在信用卡行业中使用，以在潜在欺诈性交易发生时实时地检测它们。关键任务CEP应用的列表继续增长。使用CEP技术构建关键任务应用导致需要使CEP应用具有高可用性和容错性。

[0029] 当今的信息技术(IT)环境为每样事物生成连续数据流，从监视金融市场和网络性能到业务处理执行和跟踪RFID标记的资产。CEP为开发事件处理应用提供丰富的声明性环境，以提高业务操作的有效性。CEP可以处理多个事件流，以实时检测模式和趋势并为企业提供必要的可见性以利用新出现的机会或减轻发展风险。

[0030] 连续数据流(也被称为事件流)可以包括本质上可以是连续的或无界的没有明确结束的数据或事件的流。在逻辑上，事件或数据流可以是数据元素(也被称为事件)的序列，每个数据元素具有相关联的时间戳。连续事件流可以在逻辑上被表示为元素(s,T)的包或集合，其中“s”表示数据部分，并且“T”在时域中。“s”部分一般被称为元组或事件。因此事件流可以是带时间戳的元组或事件的序列。

[0031] 在一些方面，与流中的事件相关联的时间戳可以等同于时钟时间。但是，在其它示例中，与事件流中的事件相关联的时间可以由应用域定义，并且可以不与时钟时间对应，而

是可以代替地例如由序列号表示。因而，与事件流中的事件相关联的时间信息可以由数字、时间戳或表示时间概念的任何其它信息表示。对于接收输入事件流的系统，事件以递增时间戳的次序到达系统。可以存在具有相同时间戳的多于一个事件。

[0032] 在一些示例中，事件流中的事件可以表示某个世间(worldly)事件的发生(例如，当温度传感器将值改变为了新值，当股票代号的价格改变了)并且与事件相关联的时间信息可以指示由数据流事件表示的世间事件何时发生。

[0033] 对于经由事件流接收的事件，可以使用与事件相关联的时间信息来确保事件流中的事件以递增的时间戳值的次序到达。这可以使事件流中接收到的事件能够基于它们相关联的时间信息被排序。为了使这种排序成为可能，时间戳可以以非递减的方式与事件流中的事件相关联，使得较晚生成的事件具有比较早生成的事件晚的时间戳。作为另一个示例，如果序列号被用作时间信息，那么与较晚生成的事件相关联的序列号可以大于与较早生成的事件相关联的序列号。在一些示例中，例如，当由数据流事件表示的世间事件同时发生时，多个事件可以与相同的时间戳或序列号相关联。属于相同事件流的事件一般可以以由相关联的时间信息施加在事件上的次序来处理，其中较早的事件在较晚的事件之前被处理。

[0034] 与事件流中的事件相关联的时间信息(例如，时间戳)可以由流的源来设置，或者可替代地可以由接收流的系统来设置。例如，在某些实施例中，可以在接收事件流的系统上维持心跳，并且与事件相关联的时间可以基于由心跳测量的事件到达系统处的时间。事件流中的两个事件有可能具有相同的时间信息。要注意的是，虽然时间戳排序要求特定于一个事件流，但是不同流的事件可以被任意交织。

[0035] 事件流具有相关联的模式“S”，该模式包括时间信息和一个或多个命名属性的集合。属于特定事件流的所有事件都符合与该特定事件流相关联的模式。因而，对于事件流(s，T)，事件流可以具有作为(<time_stamp>，<attribute(s)>)的模式“S”，其中<attributes>表示模式的数据部分并且可以包括一个或多个属性。例如，股票报价机事件流的模式可以包括属性<stock symbol(股票代号)>和<stock price(股票价格)>。经由这样的流接收的每个事件将具有时间戳和这两个属性。例如，股票报价机事件流可以接收以下事件和相关联的时间戳：

[0036] …

[0037] (<timestamp_N>，<NVDA，4>)

[0038] (<timestamp_N+1>，<ORCL，62>)

[0039] (<timestamp_N+2>，<PCAR，38>)

[0040] (<timestamp_N+3>，<SPOT，53>)

[0041] (<timestamp_N+4>，<PDCO，44>)

[0042] (<timestamp_N+5>，<PTEN，50>)

[0043] …

[0044] 在上面的流中，对于流元素(<timestamp_N+1>，<ORCL，62>)，事件是属性为“stock_symbol(股票_代号)”和“stock_value(股票_值)”的<ORCL，62>。与流元素相关联的时间戳是“timestamp_N+1”。因此，连续事件流是事件的流，每个事件具有相同的一系列属性。

[0045] 如所指出的，流可以是CQL查询可以对其起作用的数据的主要源。流S可以是元素(s,T)的包(也称为“多集”)，其中“s”处于S的模式中，并且“T”在时域中。此外，流元素可以是元组-时间戳对，其可以被表示为带时间戳的元组插入的序列。换句话说，流可以是带时间戳的元组的序列。在一些情况下，可以存在具有相同时间戳的多于一个元组。并且，可以请求输入流的元组以递增的时间戳次序到达系统。可替代地，关系(也被称为“时变关系”，并且不要与可以包括来自关系数据库的数据的“关系数据”混淆)可以是从时域到模式R的元组的无界包的映射。在一些示例中，关系可以是无序的、时变的元组包(即瞬时关系)。在一些情况下，在每个时间实例，关系可以是有界集合。它还可以被表示为带时间戳的元组的序列，带时间戳的元组可以包括插入、删除和/或更新以捕获关系的变化状态。类似于流，关系可以具有关系的每个元组可以符合的固定的模式。此外，如本文所使用的，连续查询可能一般能够处理流和/或关系的数据(即针对流和/或关系进行查询)。此外，关系可以引用流的数据。

[0046] 在一些方面，CQL引擎可以包括充分发展的(full blown)查询语言。照此，用户可以依据查询指定计算。此外，CQL引擎可以被设计为用于优化存储器、利用查询语言特征、运算符共享、丰富的模式匹配、丰富的语言构造等等。此外，在一些示例中，CQL引擎可以处理历史数据和流传输数据两者。例如，用户可以设置查询，以便在加利福尼亚销售高于某个目标时发送警报。因此，在一些示例中，警报可以至少部分地基于历史销售数据以及传入的实况(即实时)销售数据。

[0047] 在一些示例中，下面描述的概念的CQL引擎或其它特征可以被配置为以实时方式组合历史上下文(即仓库数据)与传入数据。因此，在一些情况下，本公开可以描述数据库存储的信息和飞行中(in-flight)的信息的边界。数据库存储的信息和飞行中的信息两者可以包括BI数据。照此，在一些示例中，数据库可以是BI服务器或者它可以是任何类型的数据库。此外，在一些示例中，本公开的特征可以使得能够在无需用户知道如何编程或以其它方式编写代码的情况下实现上述特征。换句话说，特征可以以允许非开发者实现历史数据与实时数据的组合的特征丰富的用户界面(UI)或其它方式被提供。

[0048] 在一些示例中，可以利用上述概念来充分利用与复杂事件处理相关联的丰富实时和连续事件处理能力。可以支持若干特征，诸如但不限于存档关系。照此，为了充分利用这些特征(例如，丰富、实时和连续事件处理)，系统可以被配置为透明地处理关系数据的启动状态和运行时状态。换句话说，系统可以被配置为管理在其创建的瞬间非空的查询(即存档关系)。

[0049] 在一些示例中，可以利用存档关系。照此，当CQL引擎看到指示它是基于存档关系的查询时，例如，那个存档关系还可以指示存在它可以调用以查询历史上下文的某些实体。在一些示例中，数据定义语言(DDL)可以指示关于存档关系的注释，诸如但不限于如何进行查询、表中的重要列是什么、和/或在何处发送其余的数据。在一些示例中，一旦在CQL引擎中构建了查询(例如，作为图)，系统就可以分析查询图。此外，在一些方面，存在某些有状态的运算符，如“distinct”、“group aggr”、“pattern”和/或“group by”。但是，无状态运算符可以只接受输入并将其发送给父母(例如，下游运算符)。因此，一种方法是将整个表存储在这。但是，利用存档关系，系统可以分析查询图并决定可以使用哪个最低有状态运算符来查询存档。在一些示例中，系统(或一个或多个计算机实现的方法)可以在遍历该图的同时检

索在到达的最低有状态运算符处的状态。例如,可以以来自源的拓扑次序分析查询图。至少部分地基于这第一有状态运算符,CQL引擎然后可以确定要提取的最佳数据量,以便初始化用于在存档关系上定义的查询的运算符的状态。

[0050] 在至少一个非限制性示例中,如关系和/或源的源运算符可以在拓扑遍历中首先出现,伴随查询输出和/或根最后出现。例如,如果CQL查询类似:select sum(c1)from R1where c2>c25,那么用于这个查询的计划可以类似:RelationSource→SELECT→GroupAggr。因此,遵循拓扑次序,并且由于RelationSource和SELECT两者都是无状态的,因此最低有状态运算符可以是GroupAggr。以这种方式,查询的有状态运算符(在这个示例中为GroupAggr)可以使得查询引擎能够在接收流传输数据之前用来自数据存储的历史数据填充查询引擎。这可以至少部分地基于查询正在分析存档关系并且存档关系已经被如此指示的事实来使能。

[0051] 在一些示例中,可以由用户指定给定存档关系的窗尺寸。在一些方面,与存档关系相关的窗可以包括分析或以其它方式评估传入的流传输内容的查询图中的节点。换句话说,窗可以定义由查询引擎分析和/或处理的流传输内容的量和/或将被包括在存档关系中的历史数据的量。

[0052] 在高级别,一旦窗被应用于流,它就变成了关系,然后可以应用常规关系逻辑,如同关系数据库一样。当元组到达和离开窗时,在考虑中的关系会随着针对它编译的查询而改变,同时发出结果。CQL可以支持RANGE(高达纳秒粒度)、ROWS、PARTITION BY和可扩展窗。这些窗是流到关系运算符的示例。另一方面,ISTREAM(即插入流)、DSTREAM(即删除流)和RSTREAM(即关系流)是关系到流运算符。在一些示例中,用户、开发者和/或管理者可以设置由查询引擎或者操作或托管查询引擎的一个或多个计算系统提供的窗尺寸(例如,经由UI)。在一些示例中,流上的窗可以是基于时间的范围窗。例如,可以使用窗尺寸和在其上计算窗的属性来指定存档关系上的可配置值窗。当存在在存档关系之上指定的可配置值窗时,可以计算快照查询并且可以输出窗限制内的快照元组。此外,在状态初始化之后,值窗可以应用于传入的活动数据。在一些示例中,仅将传入的活动数据插入窗,其窗属性的值与当前事件时间的差异小于窗尺寸。

[0053] 此外,在一些示例中,本公开的特征还可以充分利用CQL引擎和/或CEP引擎的连续查询处理能力来支持实时数据分析。在一些方面,CQL引擎和/或CEP引擎传统上可以是面向流的分析引擎;但是,它可以被增强以支持依靠持久存储(例如,上述存档关系)的面向流的数据。例如,本公开描述了可以支持数据对象(DO)的概念的特征,数据对象是持久存储(数据库和/或表)。对DO进行的修改可以使变化通知被广播给感兴趣的听众,从而实际上创建数据流。这个数据流可以被CQL引擎和/或CEP引擎消费,以支持任何正在运行的查询;但是,CQL引擎和/或CEP引擎可能未被设计为考虑DO后备存储中的现有数据。例如,CQL引擎和/或CEP引擎可以请求在CQL引擎和/或CEP引擎中运行的查询的初始状态反映DO的当前状态,包括当前在DO后备存储中的所有数据。一旦这样初始化了这个查询,CQL引擎和/或CEP引擎就只需要以传统面向流的样式使它自己关注从那一点开始的DO流的变化通知。

[0054] 在一些方面,CQL引擎和/或CEP引擎可以传统地处理流或非存档关系,因此可以没有初始状态。例如,可以加载查询,其中它可以开始运行并监听变化等。在一些情况下,如果用户通过状态、在条形图中询问销售,然后某人进行新的销售,那么表可以获得更新并且用

户可以预期看到推送给他们的图中的变化。但是，如果他们关闭仪表板并在一周后返回并带来一些销售，那么用户可以预期根据总计销售数据的表获得销售总额。换句话说，查询可能需要将查询带到存档的状态，然后监听活动的变化。

[0055] 在一些方面，例如，可以利用存档数据来预初始化CQL引擎。一旦被初始化，CQL引擎可以针对变化通知(例如，至少部分地基于用于插入、删除等来自存档的数据的API调用)监听Java消息传送服务(JMS)或其它信使。因此，服务可以监听，并且如果JMS在监听服务正在监听的相同主题上发布，那么它可以接收数据。这些服务不必知道谁在发布，也或者它们是否在发布。监听服务可以只是监听，并且如果发生了什么，监听服务可以听到它。在一些示例中，这是持续性如何从例如其消费者解耦。此外，在一些示例中，警报引擎可以基于警报引擎听到的内容发出警报，潜在地，并且进一步地，SQL引擎可能正在监听与监听器相关的处理中查询。

[0056] 在一些示例中，查询可以在CQL、SQL和/或CEP引擎中开始，并且指令可以被配置为获取存档数据(例如，以填装泵)，然后开始监听这些JMS消息。但是，对于大量插入、删除等，这可能包括大量信息。此外，在监听器听到消息之前可能存在滞后时间，并且在一些示例中，监听可能跳入、查询存档、返回并开始监听。因此，存在丢失和/或重复计数事件的可能性。

[0057] 此外，如果引擎仅运行查询，那么在它正在运行查询时，事物可以进入到JMS中并在引擎未监听的地方发布。因此，引擎可以被配置为首先设置监听器、运行存档查询，并且然后返回并实际开始从队列拉出，使得它不会丢失任何内容。因此，JMS可以将事物排队，并且如果事物备份，那么在引擎进行查询时是可以的，因为它可以在以后赶上并且不必担心它是否同步。如果在这里它不监听，那么它不会错过它，它只是排队，直到引擎回来，只要它已经建立了它的监听器即可。

[0058] 此外，在一些示例中，可以将系统列添加到用户的数据。这个系统列可以用于指示事务ID以尝试处置重复计数和/或丢失的操作问题。但是，在其它示例中，系统可以提供或以其它方式生成事务上下文表。此外，还可以有两个附加列TRANSACTION_CID和TRANSACTION_TID。上下文表可以始终由持续性服务来维护，以便知道最后提交的事务ID的线程(上下文)明智(wise)。事务ID可以被保证针对线程(上下文)以升序被提交。例如，当服务器启动时，它可以运行持续性服务。每个可以分派上下文ID和事务ID的集合，用于确定预先初始化的信息的数据是否包括已经通过JMS的所有数据。此外，在一些情况下，可以利用多个输出服务器(符合JTA和/或实现高可用性(HA)，其中每个服务器可以管理与由其它服务器管理的其它表完全分离的上下文/事务表的单个集合。

[0059] 在一些实施例中，当连续(例如，CQL)查询被创建或被注册时，它可以经历解析和语义分析，在其结束时逻辑查询计划被创建。当开始CQL查询时，例如，通过发出“alter query<queryname>start”DDL，可以将逻辑查询计划转换成物理查询计划。在一个示例中，物理查询计划可以被表示为物理运算符的有向无环图(DAG)。然后，可以将物理运算符转换成执行运算符，以到达用于那个CQL查询的最终查询计划。到CQL引擎的传入事件到达(一个或多个)源运算符，并最终向下游移动，其途中的运算符对那些事件执行它们的处理并产生适当的输出事件。

[0060] 事件处理应用

[0061] 在IT环境中，原始基础设施和业务事件的数量和速度都呈指数增长。无论是用于金融服务的流传输股票数据、用于军事的流传输卫星数据还是用于运输和物流业务的实时车辆位置数据，多个行业中的公司都必须实时处置大量复杂数据。此外，移动设备的爆炸式增长和无处不在的高速连接增加了移动数据的爆炸式增长。与此同时，对业务处理敏捷性和执行的需求也已在增长。这两个趋势给组织带来了增加其能力以支持事件驱动的架构实现模式的压力。实时事件处理要求基础架构和应用开发环境两者都在事件处理要求上执行。这些要求常常包括需要从日常用例扩展到极高速度的数据和事件吞吐量，可能还有以响应时间的微秒而不是秒为单位测得的时延。此外，事件处理应用必须常常检测这些事件的流中的复杂模式。

[0062] Oracle Stream Analytics平台面向众多行业和功能领域。以下是一些用例：

[0063] 电信：执行实时的呼叫细节(CDR)记录监视和分布式拒绝服务攻击检测的能力。

[0064] 金融服务：利用存在于毫秒或微秒窗中的套利机会的能力。执行金融证券交易的实时风险分析、监视和报告并计算外汇价格的能力。

[0065] 运输：在由于当地或目的地城市天气、地勤人员操作、机场安全等引起飞行差异的情况下创建乘客警报并检测行李位置的能力。

[0066] 公共部门/军事：检测分散的地理敌人信息、抽象它并解读敌人攻击的高概率的能力。警告最合适的资源以应对紧急情况的能力。

[0067] 保险：学习和检测潜在欺诈性索赔的能力。

[0068] IT系统：实时检测失败的应用或服务器并触发纠正措施的能力。

[0069] 供应链和物流：实时跟踪货物并检测和报告到达中的潜在延迟的能力。

[0070] 实时流传输和事件处理分析

[0071] 随着来自增加数量的连接设备的爆炸式增长的数据，大量动态变化的数据增加；不仅是在组织内移动的数据，而且还有防火墙外的数据。高速数据带来高价值，尤其是对于不稳定的业务处理。但是，这种数据中的一些在短时间内失去其操作价值。大数据允许处理中的充裕时间以供可操作的洞察力。另一方面，快数据要求从高度动态和战略性数据中提取最大价值。它要求更快地处理，并且促进尽可能接近生成的数据及时采取行动。Oracle Stream Analytics平台交付快数据和响应性。Oracle Edge Analytics将处理推送到网络边缘、实时地关联、过滤并分析数据，以供可操作的洞察力。

[0072] Oracle Stream Analytics平台提供将传入的流传输事件与持续数据连接的能力，从而交付上下文感知的过滤、关联、聚合和模式匹配。它为常见的事件源交付轻量级的开箱即用适配器。它还为自定义的适配器开发交付易于使用的适配器框架。利用这个平台，组织可以识别和预期机会以及由看似无关的事件所表示的威胁。其增量处理范例可以使用最少量资源来处理事件提供极低时延处理。它还允许它创建非常及时的警报，并立即检测丢失或延迟的事件，诸如以下：

[0073] 相关联的事件：如果事件A发生，那么事件B几乎总是在其2秒内跟随。

[0074] 丢失或失序的事件：事件A、B、C应当按次序发生。在A之后立即看到C，没有B。

[0075] 因果事件：所制造物品的重量缓慢地趋于变低或读数落在可接受规范之外。这标志着潜在的问题或未来的维护需求。

[0076] 除了实时事件源之外，Oracle Stream Analytics平台设计环境和运行时执行还

支持跨事件流和持续数据存储(如数据库和高性能数据网格)的基于标准的连续查询执行。这使得平台能够充当需要在几微秒或几分钟内回答以辨别否则将被忽视的模式和趋势的系统的智能的核心。事件处理用例要求内存(in-memory)处理的速度以及标准数据库SQL的数学准确性和可靠性。这个平台利用先进的自动化算法用于查询优化,查询监听传入的事件流,并对每个事件内存连续执行已注册的查询。但是,当基于内存执行模型,这个平台充分利用标准ANSI SQL语法用于查询开发,从而确保查询构造的准确性和可扩展性。这个平台完全符合ANSI SQL'99标准,并且是业界中支持ANSI SQL审查的标准SQL的扩展、用于实时连续查询模式匹配的可用的首批产品之一。CQL引擎优化处理器内查询的执行,从而使开发人员更多地关注业务逻辑而不是优化。

[0077] Oracle Stream Analytics平台允许组合SQL和Java代码两者以交付鲁棒的事件处理应用。充分利用标准行业术语来描述事件源、处理器和事件输出或接收点(sink),这个平台提供元数据驱动的方法,以在应用内定义和操纵事件。它的开发人员使用可视化的有向图画布和调色板用于应用设计,以快速概述事件流以及跨事件源和数据源两者的处理。通过拖放建模和配置向导来开发流,开发人员然后可以输入适当的元数据定义以将设计与实现连接。在必要或优选时,单击一下,开发人员就可以进入自定义Java代码开发或直接使用**Spring®** 框架将高级概念编码到其应用中。

[0078] 事件驱动的应用的特征常常在于在处置极高速率的流传输输入数据时提供低和确定性时延的需要。Oracle Stream Analytics平台的基础是基于**OSGi®** 背板(backplane)的轻量级Java容器。它包含来自WebLogic JEE应用服务器的成熟部件,诸如安全性、日志记录和工作管理算法,但在实时事件处理环境中充分利用那些服务。集成的实时内核提供独特的服务来优化由JMX框架支持的线程和存储器管理,从而为了性能和配置而使能与容器的交互。Web 2.0丰富互联网应用可以使用HTTP发布和订阅服务与平台通信,这使他们能够订阅应用频道并将事件推送到客户端。这个平台占用小(small footprint),是基于Java的轻量级容器,这可以交付更快的生产时间和更低的总体拥有成本。

[0079] Oracle Stream Analytics平台具有在标准的商用硬件上以微秒级的处理时延处置每秒数百万个事件或者最佳地使用Oracle Exalogic及其其它工程系统组合(portfolio)的能力。这是通过完整的"自上而下"分层解决方案实现的,不仅将设计聚焦于高性能事件处理用例上,而且还与企业级实时处理基础设施部件紧密集成。面向性能的服务器聚类的平台架构通过与Oracle Coherence技术的紧密集成而聚集于可靠性、容错和极高的灵活性,并使得企业能够可预测地跨数据网格扩展关键任务应用,从而确保持续的数据可用性和事务完整性。

[0080] 此外,这个平台允许确定性处理,这意味着可以以不同的速率将相同的事件馈送到多个服务器或相同的服务器,从而每次都实现相同的结果。与仅依赖正在运行的服务器的系统时钟的系统相比,这具有令人难以置信的优势。

[0081] 以上和以下描述的技术可以以多种方式并在许多上下文中实现。参考以下附图提供了若干示例实现和上下文,如下面更详细地描述的。但是,以下实现和上下文只是其中的一小部分。

[0082] 基于地理栅栏的应用的自动并行性

[0083] 在诸如汽车交通监视的应用中，数据是连续数据流的形式。连续数据流是没有明确结束范围的、到达流处理服务器的数据的流。通过处理连续数据流，应用可以检测复杂模式、事件关联以及事件之间的相互关系(relationship)。例如，连续数据流可能具有关于通过高速公路上特定区段的汽车的信息。汽车可以连续发送坐标。基于该数据流，可以解决诸如在其中解决方案需要处置车辆的数量和使用其它车辆的伙伴(buddy)之间“m”到“n”的关系以及当车辆的数量在数百万的范围内的情况下，检测接近汽车位置的伙伴的问题。

[0084] 传统数据库系统和数据处理算法中的空间数据支持通常被设计为处理被存储为有限存储数据集的空间数据。传统数据库系统将数据存储在数据库表中，其中可以使用诸如SQL的数据管理语言来查询和操纵数据。但是，数据库管理系统和算法无法处置几何形状的连续数据流，因为它们是基于系统存储大量但有限的数据集合的假设而设计的。

[0085] 由于存储器资源的限制，单个应用仅可以支持有限数量的移动对象。为了解决可伸缩性问题，可以使用聚类。但是，简单的聚类不能解决某些用例的问题，诸如上述的伙伴检测问题，因为在处理节点之间通常需要所支持区域的一些重叠。

[0086] 先前的方法使用地理分区器来解决上述问题。地理分区器通常基于其中范围被划分为预定义的距离的基于网格的方法，或者其中几何形状指定对输入几何形状进行分区的区域的基于区域的方法。这种方法是静态方法，其中分区是静态的并且不能在分区中包含移动对象的变化。例如，这种方法可能无法适应网格中移动对象的数量突然增加的场景，例如，当体育场处的球赛结束时出租车数量的增加。

[0087] 在某些实施例中，动态网格是基于诸如K-均值(K-Means)或DBSPAN的空间聚类算法而生成的。图1描绘了其中可以实现使用空间聚类算法动态确定分区尺寸的技术的示例动态网格系统或架构100。在各种实施例中，动态网格系统或架构100包括以下部件：空间聚类生成器(SCG)105、基于聚类的分区器(CBP)110和空间查询处理器(SQP)115。空间聚类生成器105部件可以被配置为递增地连续运行诸如K-Means或DBSCAN的空间聚类算法并且从输入几何形状创建几何形状聚类。可以将输出(聚类几何形状和聚类中的几何形状的数量)发送到基于聚类的分区器110。基于聚类的分区器110部件可以被配置为对输入几何形状进行分区并确定哪个几何形状去到哪个分区。在一些实施例中，基于聚类的分区器110具有动态改变分区尺寸的角色。空间查询处理器115部件处置空间查询。一个示例是带有空间数据暗盒(Spatial Cartridge)的CQLEngine。每个空间查询处理器115部件可以被配置为处置一个或多个分区。图1中描绘的部件中的一个或多个部件可以用软件、硬件或其组合来实现。在一些实施例中，软件可以存储在存储器(例如，非瞬态计算机可读介质)中、存储器设备或一些其它物理存储器上，并且可以由一个或多个处理单元(例如，一个或多个处理器、一个或多个处理器核、一个或多个GPU等)执行。

[0088] 图2-图4图示了根据一些实施例的使用空间聚类算法动态确定分区的尺寸的技术。各个实施例可以被描述为被描绘为流程图、数据流程图、结构图或框图的处理。虽然流程图可以将操作描述为顺序处理，但是许多操作可以并行或并发执行。另外，可以重新安排操作的次序。处理在其操作完成时终止，但可能有未包含在图中的附加步骤。处理可以对应于方法、函数、过程、子例程、子程序等。当处理对应于函数时，其终止可以对应于该函数到调用函数或主函数的返回。

[0089] 图2-图4中描绘的处理和/或操作可以用由一个或多个处理单元(例如，处理器核)

执行的软件(例如,代码、指令、程序)、硬件或其组合来实现。软件可以存储在存储器中(例如,存储在存储设备上、存储在非瞬态计算机可读存储介质上)。图2-图4中的处理步骤的特定系列不意图是限制性的。根据替代实施例,也可以执行其它步骤序列。例如,在替代实施例中,可以以不同次序执行上述步骤。此外,在图2-图4中所示的各个步骤可以包括多个子步骤,这些子步骤可以以适合于个别步骤的各种顺序执行。此外,取决于特定应用,可以添加或移除附加步骤。本领域普通技术人员将认识到许多变化、修改和替代。

[0090] 图2示出了流程图200,其图示了由本公开的实施例实现的总体工作流程。在一些实施例中,流程图200中描绘的处理可以由图1的动态网格系统或架构100实现。在步骤205处,可以接收与应用相关的连续数据流。在步骤210处,可以识别与连续数据流相关联的输入几何形状。输入几何形状可以指定要被分区的连续数据流中的区域。在步骤215处,可以将输入几何形状发送到SCG。在步骤220处,SCG可以在输入几何形状上执行增量空间聚类算法,以至少部分地基于输入几何形状来生成几何形状聚类。在步骤225处,SCG可以至少部分地基于几何形状聚类和几何形状聚类中的每个聚类中的几何形状的数量来生成输出几何形状。在步骤230处,SCG可以将输出几何形状发送到CBP。在步骤235处,CBP可以使用来自SCG的几何形状聚类和几何形状的数量来确定输出几何形状的一个或多个分区。在一些实施例中,CBP的分区算法可以包括以下处理:(i)聚类变化检测,(ii)聚类移除,(iii)分区的改变,以及(iv)分区标识符到输入几何形状的分配。下面更详细地描述这些处理。在步骤240处,可以动态地改变输出几何形状的一个或多个分区的尺寸。在步骤245处,可以在一个或多个分区的尺寸被动态改变的情况下传送与连续数据流相关联的输出几何形状。

[0091] 图3示出了流程图300,其图示了由本公开的实施例实现的CBP的聚类变化检测处理。在一些实施例中,流程图300中描绘的处理可以由图1的动态网格系统或架构100实现。在某些实施例中,可以使用索引(诸如RTree索引)来检测聚类的变化来进行聚类中是否存在变化的确定。RTree索引或R树是用于空间访问方法的树数据结构,即用于对诸如地理坐标、矩形或多边形的多维信息进行索引。在步骤305处,可以将聚类的边界框加载到RTree。在步骤310处,可以执行RTree索引中的输出聚类的查找以比较尺寸。例如,可以将几何形状聚类的边界框加载到索引,并且可以将几何形状聚类与RTree索引内的其他几何形状聚类进行比较。可以基于包括聚类的尺寸的一个或多个参数来执行比较。在步骤315处,如果未从索引中找到聚类,则可以进行聚类是新聚类的确定。在步骤320处,当聚类是新聚类时,可以在没有旧聚类尺寸的情况下执行分区的改变处理。在步骤325处,如果从索引中找到聚类,则可以进行聚类不是新聚类(例如,是现有聚类)的确定。在步骤330处,当聚类不是新聚类时,可以识别尺寸阈值以确定分区的改变。例如,如果尺寸变化为X百分比,则可以利用X百分比执行分区的改变处理。

[0092] 图4示出了流程图400,其图示了由本公开的实施例实现的聚类移除处理、分区的改变处理以及分区标识符的分配处理。在一些实施例中,流程图400中描绘的处理可以由图1的动态网格系统或架构100实现。在步骤405处,如果利用定时器在预定时间内没有从SCG中找到聚类,则可以移除RTree中的聚类。在步骤410处,可以改变分区,其可能涉及决定网格范围,创建到分区索引表的网格索引,以及将到分区索引表的网格索引设置为存储在RTree中的聚类对象,使得聚类对象可以用在分配分区标识符中。网格范围的决定可以通过配置参数网格尺寸(GridSize)、尺寸影响因子(SizeEffectFactor)和从SCG给出的几何形

状的数量来实现。在示例中，可以使用几何形状的数量通过应用等式(GridSize/(几何形状的数量*SizeEffectFactor))来调整GridSize。可以通过调整后的GridSize来划分聚类的边界框，这创建了新的分区或添加了附加的分区。在示例中，可以使用一致的散列技术来创建分区标识符以维持系统负载平衡。由于网格索引不直接映射到分区id，因此在一些实施例中可以使用查找表将网格索引映射到分区id。在某些实施例中，聚类移除算法还可以包括向输入几何形状分配分区Id。可以通过在RTree中搜索聚类、通过范围比较找到网格并且执行对到分区索引表的网格索引的查找，来执行分区标识符的分配。因此，本公开的实施例提供了用于使用结合空间聚类和网格分区的空间聚类算法动态确定分区的尺寸的技术。此外，公开了从几何形状的流中动态地改变分区和网格尺寸的技术。

[0093] 图5是根据本公开的实施例的至少部分地基于输入几何形状505、诸如Rtree的其中可以加载输入几何形状505的索引510，以及包括网格范围和到分区索引表的网格索引(其被设置为存储在索引510中的聚类对象)的网格515生成的几何形状聚类的示例性图示。

[0094] 实时流传输(SparkStreaming)中的空间变化检测器以及检查和设置操作

[0095] 本公开的实施例提供了执行流中移动对象的接近度检测和检查的技术。在各种实施例中，公开了一种空间变化检测器，其可以跟踪移动对象并且可以在两个或更多个移动对象在彼此的接近度内时产生警报。例如，空间变化检测器可以跟踪机场中的移动对象，并确保它们不会进入给定范围的接近度内。

[0096] 在一些实施例中，可以使用空间“在距离内(withindistance)”操作来实现对移动对象的跟踪。但是，如果移动对象是流的一部分，那么必须计算距移动对象流的自连接的n!组合的距离。计算两个地点与GPS的距离可能是相对繁重的点操作。由本公开的实施例实现的一个解决方案是利用如图6中所示的将流605的几何形状(例如，关于车辆的全球位置的数据)转换为关系610并执行流605和关系610之间的连接操作615的空间变化检测器600。可以是FROM子句中可能的TableExpressions当中的连接操作在两个数据集合或表之间执行连接。这使得能够充分利用关系中的空间索引并避免计算n!组合的距离。空间索引是一类扩展索引，其允许空间列被索引。空间列是表列，其包含空间数据类型(诸如几何形状或地形)的数据。当充分利用空间索引时，可以应用具有范围的过滤，并且可以从过滤的结果计算距离。这导致n平方运算。为了将流605转换为关系610，可以检测相同键的内容的变化。在实施例中，支持插入、删除和更新操作的内存高速缓存620可以用于检测是否存在内容的变化，发出“更新”事件，并将流605转换为关系610。诸如散列图(HashMap)之类的内存高速缓存620可以被设计为通过将经常被请求的数据保持在存储器中，从而减少对数据库查询以得到该数据的需要来提高应用性能。此后，流605和关系610可以用在空间“withindistance”操作625中，以跟踪移动对象彼此的关系并执行接近度检测。诸如“withindistance”之类的空间操作使用几何形状函数将空间数据作为输入、分析数据，然后产生输出数据，该输出数据是对输入数据执行的分析的衍生物(derivative)。

[0097] 在一些实施例中，可以实现检查和设置操作。例如，当存在针对对象的流的请求和匹配操作时(例如，在根据乘客请求调度出租车时)，需要来自匹配的仲裁。对于上面的示例，根据乘客请求流，可以确定特定范围内的出租车。一旦找到候选出租车，就需要选取其中一辆出租车并将其标记为“已预订(booked)”，使得其它乘客可以避免对相同出租车的双重预订。由于CQL是查询语言而不是过程语言，因此不能同时执行检查和设置操作。

[0098] 支持这种检查和设置操作的一种方式是使用内存高速缓存(或HashMap)。类似于用作并发原子操作的测试和设置(Test-and-set)或比较和交换(Compare-and-swap),由内存高速缓存705支持的如图7所示的设置(CheckAndSet)检测器700可以被配置为将第一流710的几何形状(例如,关于车辆的全球位置的数据)转换为关系715并且在关系715和第二流725(例如,关于请求的全局位置的数据)之间执行检查和设置操作720。这使得能够充分利用来自关系的空间索引并避免计算n!组合的距离。当充分利用空间索引时,可以应用具有范围的过滤,并且可以从过滤的结果计算距离。这导致n平方运算。为了将第一流710转换为关系715,可以检测相同键的内容的变化。在实施例中,支持插入、删除和更新操作的内存高速缓存705可以用于检测是否存在内容的变化,发出“更新”事件,并将第一流710转换为关系715。此后,关系715和第二流725可以用在检查和设置操作720中,使得来自第一流710的数据针对来自第二流725的数据(例如,靠近乘客请求的地点的出租车可以被选取)被检查,并且条目对象的属性被设置(例如,出租车被标记为“已预订”,使得其它乘客可以避免双重预订)。在某些实施例中,诸如'select passengerId,taxiId from taxiRelation, requestStream,where taxiCache.checkAndSet('booked',true)'的select语句可以用于检查条目对象的“已预订”属性在出租车高速缓存(taxiCache)中是否为“假(false)”并将“已预订”属性设置为真(true),并返回true,并且在产生改变条目的副作用时对结果应用过滤。

[0099] 在各种实施例中,检查和设置操作可以涉及使用内存高速缓存(或HashMap),并且可以如下实现:

添加 CacheDStream 和 CacheRDD

CQLEngine 创建单例 Cache。

CQLEngine 通过委托添加 Cache 方法的 RPC 操作。

CacheRDD 应该与 CQLEngine 共同分区。

CacheRDD 通过 Cache RPC 操作将高速缓存操作委托给 CQLEngine

Cache - 变化检测的正常操作

```
if (get(key) == null) {
        put
        add(new TupleValue(PLUS, ...))
} else {
        put
        add(new TupleValue(UPDATE, ...)
}
if (expiredTupleQueue.size > 0)  {
        remove all tuples from the queue and remove
        add(new TupleValue(MINUS, ...)
}
```

Cache 高速缓存启动定时器以使元组自我过期并将过期的元组添加到队列中，

expiredTupleQueue

Cache 高速缓存还具有 CheckAndSet 操作

```
value = get(key)
if (value.getProperty(property) != newValue)
        value.setProperty(property, newValue)
        return true
else return false
```

[0102] 因此，使用上述技术，所公开的空间变化检测器使得能够更快地针对几何数据流执行空间“withindistance”操作，并且所公开的检查和设置操作使得仲裁能够与CQL一起被提供。使用上述技术，可以实现诸如移动对象的接近度检查和请求调度系统的地理流传输用例。

[0103] 图8描绘了其中可以实现用于执行流中移动对象的接近度检测和检查以及检查和设置操作的技术的示例系统或架构。在一些实施例中，被配置为提供用于处理事件流的环境的事件处理服务或系统800包括CQL处理器805、JAVA暗盒810、定时器815、内存高速缓存820和CacheRDD 825。CQL处理器805可以与对由输入通道提供的事件进行操作的一个或多个CQL查询相关联。CQL处理器连接到对其写入查询结果的输出通道。定时器815可以用于组织和从内存高速缓存820中移除过期数据。CacheRDD 825可以接收数据流830并利用内存高速缓存820将数据流转换为关系835，如本文关于至少图6和图7所讨论的。

[0104] 图9描绘了可以结合本公开实施例的事件处理系统200的简化高级图。在一些实施例中，事件处理系统900可以包括一个或多个事件源(904、906、908)、被配置为提供用于处理事件流的环境的事件处理服务(EPS)902(也称为CQ服务902)以及一个或多个事件接收点(910、912)。事件源生成由EPS 902接收的事件流。EPS 902可以从一个或多个事件源接收一个或多个事件流。例如，如图9中所示，EPS 902从事件源904接收第一输入事件流914，从事件源906接收第二输入事件流916，以及从事件源908接收第三事件流918。一个或多个事件处理应用(920、922和924)可以部署在EPS 902上并由EPS 902执行。由EPS 902执行的事件处理应用可以被配置为监听一个或多个输入事件流，基于从输入事件流中选择一个或多个事件作为值得注意的事件的处理逻辑来处理经由一个或多个事件流接收的事件。然后可以以一个或多个输出事件流的形式将值得注意的事件发送到一个或多个事件接收点(910、912)。例如，在图9中，EPS 902将第一输出事件流926输出到事件接收点910，并将第二输出事件流928输出到事件接收点912。在某些实施例中，事件源、事件处理应用和事件接收点彼此解耦，使得可以添加或移除任何这些部件而不会导致对其它部件的改变。

[0105] 在一个实施例中，EPS 902可以被实现为包括轻量级Java应用容器的Java服务器，诸如基于Equinox OSGi、具有共享服务的容器。在一些实施例中，EPS 902可以支持用于处理事件的超高吞吐量和微秒时延，例如，通过使用JRockit Real Time。EPS 902还可以提供开发平台(例如，完整的实时端到端Java事件驱动架构(EDA)开发平台)，其包括用于开发事件处理应用的工具(例如，Oracle CEP Visualizer和Oracle CEP IDE)。

[0106] 事件处理应用被配置为监听一个或多个输入事件流，执行从一个或多个输入事件流中选择一个或多个值得注意的事件的逻辑(例如，查询)，并经由一个或多个输出事件流将所选择的值得注意的事件输出到一个或多个事件源。图9提供了用于一个这样的事件处理应用920的深入分析。如图9中所示，事件处理应用920被配置为监听输入事件流918，执行包括从输入事件流918中选择一个或多个值得注意的事件的逻辑的连续查询930，并经由输出事件流928将所选择的值得注意的事件输出到事件接收点912。事件源的示例包括但不限于适配器(例如，JMS、HTTP和文件)、信道、处理器、表、高速缓存等。事件接收点的示例包括但不限于适配器(例如，JMS、HTTP和文件)、信道、处理器、高速缓存等。

[0107] 虽然图9中的事件处理应用020被示为监听一个输入流并经由一个输出流输出所选择的事件，但是这不意图是限制性的。在替代实施例中，事件处理应用可以被配置为监听

从一个或多个事件源接收的多个输入流，从被监视的流中选择事件，并经由一个或多个输出事件流将所选择的事件输出到一个或多个事件接收点。相同的查询可以与多于一个事件接收点和与不同类型的事件接收点相关联。

[0108] 由于其无界性质，经由事件流接收的数据量一般非常大。因此，为了查询目的而存储或存档所有数据一般是不切实际且不合需要的。事件流的处理要求在事件被EPS 902接收时对事件的实时处理，而不必存储所有接收到的事件数据。因而，EPS 902提供特殊的查询机制，该机制使得能够在事件被EPS 902接收时执行事件的处理，而不必存储所有接收到的事件。

[0109] 事件驱动的应用是规则驱动的，并且这些规则可以以用于处理输入流的连续查询的形式表达。连续查询可以包括指令(例如，业务逻辑)，该指令识别要对接收到的事件执行的处理，包括将要选择什么事件作为值得注意的事件并作为查询处理的结果输出。连续查询可以持续留存到数据存储并且用于处理事件的输入流并生成事件的输出流。连续查询通常执行过滤和聚合功能，以从输入事件流中发现和提取值得注意的事件。因此，输出事件流中的外出(outbound)事件的数量一般远低于从中选择事件的输入事件流中的事件的数量。

[0110] 与在有限数据集上运行一次的SQL查询不同，每次在特定事件流中接收到事件时，可以执行已经由具有EPS 902的应用为该特定事件流注册的连续查询。作为连续查询执行的一部分，EPS 902基于由连续查询指定的指令来评估接收到的事件，以确定是否要选择一个或多个事件作为值得注意的事件并且作为连续查询执行的结果输出。

[0111] 可以使用不同语言对连续查询进行编程。在某些实施例中，可以使用由Oracle公司提供并且由Oracle的复杂事件处理(CEP)产品供应使用的CQL来配置连续查询。Oracle的CQL是一种声明性语言，其可以用于对可以针对事件流执行的查询(称为CQL查询)进行编程。在某些实施例中，CQL基于SQL，其中添加了支持流事件数据处理的构造。

[0112] 在一个实施例中，事件处理应用可以由以下部件类型组成：

[0113] (1)一个或多个适配器，它们直接接口到输入和输出流以及关系源和接收点。适配器被配置为理解输入和输出流协议，并负责将事件数据转换成可由应用处理器查询的规范化形式。适配器可以将规范化的事件数据转发到信道或输出流和关系接收点中。可以为各种数据源和接收点定义事件适配器。

[0114] (2)充当事件处理端点的一个或多个信道。除其它之外，信道尤其负责对事件数据进行排队，直到事件处理代理可以对其起作用。

[0115] (3)一个或多个应用处理器(或事件处理代理)被配置为消费来自信道的规范化事件数据、使用查询来处理它以选择值得注意的事件，并将所选择的值得注意的事件转发(或复制)到输出信道。

[0116] (4)一个或多个bean被配置为监听输出信道，并且通过将新事件插入输出信道而被触发。在一些实施例中，这个用户代码是简单的旧Java对象(POJO)。用户应用可以使用外部服务的集合(诸如JMS、Web服务和文件编写器)以将生成的事件转发到外部事件接收点。

[0117] (5)事件bean可以被注册以监听输出信道，并且通过将新事件插入输出信道被触发。在一些实施例中，这个用户代码可以使用Oracle CEP事件bean API，使得可以由Oracle CEP管理bean。

[0118] 在一个实施例中，事件适配器将事件数据提供给输入信道。输入信道被连接到与

一个或多个CQL查询相关联的CQL处理器，所述CQL查询对由输入信道提供的事件进行操作。CQL处理器连接到写入查询结果的输出信道。

[0119] 在一些实施例中，可以为事件处理应用提供组装文件，该文件描述事件处理应用的各种部件、部件如何连接在一起、由应用处理的事件类型。可以提供分离的文件，用于指定用于事件选择的连续查询或业务逻辑。

[0120] 应当认识到的是，图9中描绘的系统900可以具有除图9中描绘的那些之外的其它部件。另外，图9中示出的实施例仅仅是可以结合本公开实施例的系统的一个示例。在一些其它实施例中，系统900可以具有比图9中所示更多或更少的部件，可以组合两个或更多个部件，或者可以具有不同的部件配置或布置。系统900可以是各种类型，包括服务提供商计算机、个人计算机、便携式设备(例如，移动电话或设备)、工作站、网络计算机、大型机、信息亭、服务器或任何其它数据处理系统。在一些其它实施例中，系统900可以被配置为分布式系统，其中系统900的一个或多个部件跨云中的一个或多个网络而分布。

[0121] 图9中描绘的部件中的一个或多个可以用软件、硬件或其组合来实现。在一些实施例中，软件可以存储在存储器(例如，非瞬态计算机可读介质)中、存储器设备或某种其它物理存储器上，并且可以由一个或多个处理单元(例如，一个或多个处理器、一个或多个处理器核、一个或多个GPU等)执行。

[0122] 图10图示了根据一些实施例的用于空间变化检测以及检查和设置操作的技术。各个实施例可以被描述为被描绘为流程图、数据流程图、结构图或框图的处理。虽然流程图可以将操作描述为顺序处理，但是许多操作可以并行或并发执行。此外，可以重新安排操作的次序。处理在其操作完成时终止，但可以有未包含在图中的附加步骤。处理可以与方法、函数、过程、子例程、子程序等对应。当处理与函数对应时，其终止可以与该函数到调用函数或主函数的返回对应。

[0123] 图10中描绘的处理和/或操作可以由一个或多个处理单元(例如，处理器核)执行的软件(例如，代码、指令、程序)、硬件或其组合来实现。软件可以存储在存储器中(例如，存储在存储设备上、存储在非瞬态计算机可读存储介质上)。图10中的处理步骤的特定系列不意图是限制性的。根据替代实施例，还可以执行其它步骤序列。例如，在替代实施例中，可以以不同次序执行上面概述的步骤。而且，图10中所示的各个步骤可以包括多个子步骤，这些子步骤可以以适合于各个步骤的各种顺序执行。此外，取决于特定应用，可以添加或移除附加步骤。本领域普通技术人员将认识到许多变化、修改和替代。

[0124] 图10示出了流程图1000，其图示了由本公开的实施例实现的空间变化检测和/或检查和设置操作。在一些实施例中，流程图1000中描绘的处理可以由图8和图9的事件处理系统实现。在步骤1005处，接收与应用相关的第一连续数据流。在步骤1010处，将第一连续数据流的几何形状转换为关系。关系可以包括第一连续数据流的几何形状的空间索引。在某些实施例中，向空间索引应用具有范围的过滤，以获得第一连续数据流的几何形状的过滤的结果。在一些实施例中，转换包括确定第一连续数据流的几何形状中是否存在变化，以及当第一连续数据流的几何形状中存在变化时发出更新事件。更新事件在支持插入、删除和更新操作的内存高速缓存上发出。

[0125] 在步骤1015处，跟踪第一连续数据流中的多个移动对象。可以使用空间操作执行跟踪来跟踪至少第一移动对象与第二移动对象的相互关系。在步骤1020处，基于第一连续

数据流的几何形状和关系中的至少一个来确定或检查多个移动对象中的至少第一移动对象和第二移动对象之间的相互关系。在某些实施例中，第一连续数据流的几何形状和关系在跟踪之前使用连接操作进行连接。

[0126] 可选地，接收与应用相关的第二连续数据流，并且至少基于关系和第二连续数据流的几何形状来确定或检查多个移动对象中的至少一个移动对象与第二连续数据流中的对象之间的相互关系。在一些实施例中，确定相互关系包括至少基于第一连续数据流的几何形状和关系来确定多个移动对象中的至少第一移动对象和第二移动对象之间的接近度。在其它实施例中，确定相互关系包括至少基于关系和第二连续数据流的几何形状来确定多个移动对象中的至少移动对象与第二连续数据流中的对象之间的接近度。确定接近度可以包括使用过滤的结果计算第一移动对象和第二移动对象之间的距离，或者计算多个移动对象中的移动对象与第二连续数据流中的对象之间的距离。

[0127] 在步骤1025处，基于所确定的相互关系采取动作。在各种实施例中，通过应用于以下中的一个或多个的操作来执行动作：关系、第一连续数据流的几何形状和第二连续数据流的几何形状。在一些实施例中，动作包括当至少第一移动对象和第二移动对象之间的接近度超过预定阈值时产生警报。在其它实施例中，动作包括当相互关系满足预定标准(例如，超过预定阈值)时设置移动对象的属性。

[0128] 说明性系统

[0129] 图11-图13图示了用于实现根据各种实施例的本公开各方面的示例环境的各方面。图11描绘了用于实现本公开实施例的分布式系统1100的简化图。在所示实施例中，分布式系统1100包括一个或多个客户端计算设备1102、1104、1106和1108，这些客户端计算设备被配置为通过(一个或多个)网络1110执行和操作客户端应用，诸如web浏览器、专有客户端(例如，Oracle Forms)等等。服务器1112可以经由网络1110与远程客户端计算设备1102、1104、1106和1108通信地耦合。

[0130] 在各种实施例中，服务器1112可以适于运行一个或多个服务或软件应用，诸如提供身份管理服务的服务和应用。在某些实施例中，服务器1112还可以提供其它服务或者软件应用可以包括非虚拟和虚拟环境。在一些实施例中，这些服务可以作为基于web的或云服务或者在软件即服务(SaaS)模型下提供给客户端计算设备1102、1104、1106和/或1108的用户。操作客户端计算设备1102、1104、1106和/或1108的用户转而可以利用一个或多个客户端应用与服务器1112交互以利用由这些部件提供的服务。

[0131] 在图11描绘的配置中，系统1100的软件部件1118、1120和1122被示出为在服务器1112上实现。在其它实施例中，系统1100的一个或多个部件和/或由这些部件提供的服务可以也可以由客户端计算设备1102、1104、1106和/或1108中的一个或多个来实现。然后，操作客户端计算设备的用户可以利用一个或多个客户端应用来使用由这些部件提供的服务。这些部件可以用硬件、固件、软件或其组合来实现。应该理解的是，各种不同的系统配置是可能的，其可以与分布式系统1100不同。因此，图11中所示的实施例是用于实现实施例系统的分布式系统的一个示例，而不意图是限制性的。

[0132] 客户端计算设备1102、1104、1106和/或1108可以包括各种类型的计算系统。例如，客户端设备可以包括便携式手持设备(例如，**iPhone®**、蜂窝电话、**iPad®**、计算平板、

个人数字助理(PDA))或可穿戴设备(例如,Google Glass®头戴式显示器),其运行诸如Microsoft Windows Mobile®之类的软件和/或诸如iOS、Windows Phone、Android、BlackBerry 10,Palm OS等之类的各种移动操作系统。设备可以支持各种应用,诸如各种互联网相关的应用、电子邮件、短消息服务(SMS)应用,并且可以使用各种其它通信协议。客户端计算设备还可以包括通用个人计算机,作为示例,包括运行各种版本的Microsoft Windows®、Apple Macintosh®和/或Linux操作系统的个人计算机和/或膝上型计算机。客户端计算设备可以是运行任何各种商用的UNIX®或类UNIX操作系统(包括但不限于诸如像Google Chrome OS的各种GNU/Linux操作系统)的工作站计算机。客户端计算设备还可以包括能够在(一个或多个)网络1110上通信的电子设备,诸如瘦客户端计算机、启用互联网的游戏系统(例如,具有或不具有Kinect®姿势输入设备的Microsoft Xbox®游戏控制台)和/或个人消息传送设备。

[0133] 虽然图11中的分布式系统1100被示出具有四个客户端计算设备,但是可以支持任何数量的客户端计算设备。其它设备,诸如具有传感器的设备等,可以与服务器1112交互。

[0134] 分布式系统1100中的(一个或多个)网络1110可以是对本领域技术人员来说熟悉的、可以使用任何各种可用协议支持数据通信的任何类型的网络,其中各种协议包括但不限于TCP/IP(传输控制协议/互联网协议)、SNA(系统网络架构)、IPX(互联网分组交换)、AppleTalk等。仅仅作为示例,(一个或多个)网络1110可以是局域网(LAN)、基于以太网的网络、令牌环、广域网、因特网、虚拟网络、虚拟专用网络(VPN)、内联网、外联网、公共交换电话网络(PSTN)、红外网络、无线网络(例如,在任何电气和电子协会(IEEE)1002.11协议套件、Bluetooth®、和/或任何其它无线协议下操作的网络)和/或这些和/或其它网络的任意组合。

[0135] 服务器1112可以由一个或多个通用计算机、专用服务器计算机(作为示例,包括PC(个人计算机)服务器、UNIX®服务器、中型服务器、大型计算机、机架安装的服务器等)、服务器场、服务器集群或任何其它适当的布置和/或组合组成。服务器1112可以包括运行虚拟操作系统的一个或多个虚拟机,或涉及虚拟化的其它计算架构。一个或多个灵活的逻辑存储设备池可以被虚拟化,以维护用于服务器的虚拟存储设备。虚拟网络可以由服务器1112利用软件定义网络来控制。在各种实施例中,服务器1112可以适于运行在前述公开内容中描述的一个或多个服务或软件应用。例如,服务器1112可以与根据本公开的实施例的用于如上所述执行处理的服务器对应。

[0136] 服务器1112可以运行包括以上讨论那些中的任何操作系统的操作系统,以及任何商业上可用的服务器操作系统。服务器1112还可以运行任何各种附加的服务器应用和/或中间层应用,包括HTTP(超文本传输协议)服务器、FTP(文件传输协议)服务器、CGI(公共网关接口)服务器、JAVA®服务器、数据库服务器等。示例性数据库服务器包括但不限于可从Oracle、Microsoft、Sybase、IBM(国际商业机器)等商业上可用的数据库服务器。

[0137] 在一些实现中,服务器1112可以包括一个或多个应用,以分析和整合从客户端计算设备1102、1104、1106和1108的用户接收到的数据馈送和/或事件更新。作为示例,数据馈

送和/或事件更新可以包括但不限于从一个或多个第三方信息源和持续数据流接收到的**Twitter®**馈送、**Facebook®**更新或实时更新，其可以包括与传感器数据应用、金融报价机、网络性能测量工具(例如，网络监视和流量管理应用)、点击流分析工具、汽车流量监视等相关的实时事件。服务器1112还可以包括经由客户端计算设备1102、1104、1106和1108的一个或多个显示设备显示数据馈送和/或实时事件的一个或多个应用。

[0138] 分布式系统1100还可以包括一个或多个数据库1114和1116。这些数据库可以提供用于存储信息(诸如用户身份信息和由本公开实施例使用的其它信息)的机制。数据库1114和1116可以驻留在各种地点中。举例来说，数据库1114和1116中的一个或多个可以驻留在服务器1112本地(和/或驻留在服务器1112中)的非瞬态存储介质上。可替代地，数据库1114和1116可以远离服务器1112并经由基于网络的或专用的连接与服务器1112通信。在一组实施例中，数据库1114和1116可以驻留在存储区域网络(SAN)中。类似地，用于执行归属于服务器1112的功能的任何必要文件可以适当地本地存储在服务器1112上和/或远程存储。在一组实施例中，数据库1114和1116可以包括适于响应于SQL格式的命令来存储、更新和检索数据的关系数据库，诸如由Oracle提供的数据库。

[0139] 图12图示了可以用于实现本公开实施例的示例性计算机系统1200。在一些实施例中，计算机系统1200可以用于实现上述各种服务器和计算机系统中的任何一种。如图12中所示，计算机系统1200包括各种子系统，包括处理子系统1204，处理子系统1204经由总线子系统1202与多个外围子系统通信。这些外围子系统可以包括处理加速单元1206、I/O子系统1208、存储子系统1218和通信子系统1224。存储子系统1218可以包括有形计算机可读存储介质1222和系统存储器1210。

[0140] 总线子系统1202提供用于使计算机系统1200的各种部件和子系统按预期彼此通信的机制。虽然总线子系统1202被示意性地示为单个总线，但总线子系统的替代实施例可以使用多个总线。总线子系统1202可以是若干类型的总线结构中的任何一种，包括使用各种总线架构中的任何总线架构的存储器总线或存储器控制器、外围总线和本地总线。例如，此类架构可以包括工业标准架构(ISA)总线、微信道架构(MCA)总线、增强型ISA(EISA)总线、视频电子标准协会(VESA)本地总线和外围部件互连(PCI)总线，其可以实现为根据IEEE P1386.1标准制造的夹层(Mezzanine)总线等等。

[0141] 处理子系统1204控制计算机系统1200的操作，并且可以包括一个或多个处理单元1232、1234等。处理单元可以包括一个或多个处理器(包括单核或多核处理器)、处理器的一个或多个核，或其组合。在一些实施例中，处理子系统1204可以包括一个或多个专用协处理器，诸如图形处理器、数字信号处理器(DSP)等。在一些实施例中，处理子系统1204的一些或全部处理单元可以使用自定义电路来实现，诸如专用集成电路(ASIC)或现场可编程门阵列(FPGA)。

[0142] 在一些实施例中，处理子系统1204中的处理单元可以执行存储在系统存储器1210中或计算机可读存储介质1222上的指令。在各种实施例中，处理单元可以执行各种程序或代码指令并且可以维护多个并发执行的程序或进程。在任何给定时间，要执行的程序代码中的一些或全部可以驻留在系统存储器1210中和/或驻留在计算机可读存储介质1210上，计算机可读存储介质1210可能包括在一个或多个存储设备。通过适当的编程，处理子系统1204可以提供上述各种功能，用于响应于使用模式而动态地修改文档(例如，网页)。

[0143] 在某些实施例中，可以提供处理加速单元1206，用于执行自定义处理或用于卸载由处理子系统1204执行的一些处理，以便加速由计算机系统1200执行的整体处理。

[0144] I/O子系统1208可以包括用于向计算机系统1200输入信息和/或用于从计算机系统1200或经由计算机系统1200输出信息的设备和机制。一般而言，术语“输入设备”的使用旨在包括所有可能的用于向计算机系统1200输入信息的设备和机制的类型。用户接口输入设备可以包括例如键盘、诸如鼠标或轨迹球之类的定点设备、结合到显示器中的触摸板或触摸屏、滚轮、点击轮、拨号盘、按钮、开关、小键盘、具有语音命令识别系统的音频输入设备、麦克风和其它类型的输入设备。用户接口输入设备也可以包括使用户能够控制输入设备并与其交互的诸如Microsoft **Kinect®**运动传感器的运动感测和/或姿势识别设备、Microsoft **Xbox®** 360游戏控制器、提供用于接收利用姿势和口语命令的输入的接口的设备。用户接口输入设备也可以包括眼睛姿势识别设备，诸如从用户检测眼睛活动(例如，当拍摄图片和/或进行菜单选择时的“眨眼”)并将眼睛姿势转换为到输入设备(例如，Google **Glass®**)中的输入的Google **Glass®**眨眼检测器。此外，用户接口输入设备可以包括使用户能够通过语音命令与语音识别系统(例如，**Siri®**导航器)交互的语音识别感测设备。

[0145] 用户接口输入设备的其它示例包括但不限于三维(3D)鼠标、操纵杆或指点杆、游戏手柄和图形输入板，以及诸如扬声器、数码相机、数码摄像机、便携式媒体播放器、网络摄像头、图像扫描仪、指纹扫描仪、条形码阅读器3D扫描仪、3D打印机、激光测距仪和眼睛注视跟踪设备之类的音频/视觉设备。此外，用户接口输入设备可以包括，例如医疗成像输入设备，诸如计算机断层摄影、磁共振成像、位置发射断层摄影、医疗超声检查设备。用户接口输入设备也可以包括，例如音频输入设备，诸如MIDI键盘、数字乐器等。

[0146] 用户接口输出设备可以包括显示子系统、指示器灯或诸如音频输出设备之类的非可视显示器等。显示子系统可以是阴极射线管(CRT)、诸如使用液晶显示器(LCD)或等离子体显示器的平板设备、投影设备、触摸屏等。一般而言，术语“输出设备”的使用旨在包括用于从计算机系统1200向用户或其它计算机输出信息的所有可能类型的设备和机制。例如，用户接口输出设备可以包括但不限于可视地传达文本、图形和音频/视频信息的各种显示设备，诸如监视器、打印机、扬声器、耳机、汽车导航系统、绘图仪、语音输出设备和调制解调器。

[0147] 存储子系统1218提供用于存储由计算机系统1200使用的信息的储存库或数据存储。存储子系统1218提供有形非瞬态计算机可读存储介质，用于存储提供一些实施例的功能的基本编程和数据结构。当由处理子系统1204执行时提供上述功能的软件(程序、代码模块、指令)可以存储在存储子系统1218中。软件可以由处理子系统1204的一个或多个处理单元执行。存储子系统1218也可以提供用于存储根据本公开使用的数据的储存库。

[0148] 存储子系统1218可以包括一个或多个非瞬态存储器设备，包括易失性和非易失性存储器设备。如图12中所示，存储子系统1218包括系统存储器1210和计算机可读存储介质1222。系统存储器1210可以包括多个存储器，包括用于在程序执行期间存储指令和数据的易失性主随机存取存储器(RAM)和其中存储固定指令的非易失性只读存储器(ROM)或闪存存储器。在一些实现中，包含帮助在诸如启动期间在计算机系统1200内的元件之间传递信

息的基本例程的基本输入/输出系统(BIOS)通常可以存储在ROM中。RAM通常包含当前由处理子系统1204操作和执行的数据和/或程序模块。在一些实现中,系统存储器1210可以包括多个不同类型的存储器,诸如静态随机存取存储器(SRAM)或动态随机存取存储器(DRAM)。

[0149] 作为示例而非限制,如在图12中所描绘的,系统存储器1210可以存储应用程序1212,其可以包括客户端应用、Web浏览器、中间层应用、关系数据库管理系统(RDBMS)等,程序数据1214和操作系统1216。作为示例,操作系统1216可以包括各种版本的Microsoft **Windows®**、Apple **Macintosh®**和/或Linux操作系统、各种商业上可用的**UNIX®**或类UNIX操作系统(包括但不限于各种GNU/Linux操作系统、Google **Chrome®** OS等)和/或诸如iOS、**Windows®** Phone、**Android®** OS、**BlackBerry®** 10OS和**Palm®** OS操作系统的移动操作系统。

[0150] 计算机可读存储介质1222可以存储提供一些实施例的功能的编程和数据构造。当由处理子系统1204执行时使处理器提供上述功能的软件(程序、代码模块、指令)可以存储在存储子系统1218中。作为示例,计算机可读存储介质1222可以包括非易失性存储器,诸如硬盘驱动器、磁盘驱动器、诸如CD ROM、DVD、**Blu-Ray®** (蓝光)盘或其它光学介质的光盘驱动器。计算机可读存储介质1222可以包括但不限于**Zip®**驱动器、闪存存储器卡、通用串行总线(USB)闪存驱动器、安全数字(SD)卡、DVD盘、数字视频带等。计算机可读存储介质1222也可以包括基于非易失性存储器的固态驱动器(SSD)(诸如基于闪存存储器的SSD、企业闪存驱动器、固态ROM等)、基于易失性存储器的SSD(诸如基于固态RAM、动态RAM、静态RAM、DRAM的SSD、磁阻RAM(MRAM)SSD),以及使用基于DRAM和基于闪存存储器的SSD的组合的混合SSD。计算机可读介质1222可以为计算机系统1200提供计算机可读指令、数据结构、程序模块和其它数据的存储。

[0151] 在某些实施例中,存储子系统1200也可以包括计算机可读存储介质读取器1220,其可以进一步连接到计算机可读存储介质1222。一起并且可选地,与系统存储器1210组合,计算机可读存储介质1222可以全面地表示远程、本地、固定和/或可移动存储设备加上用于存储计算机可读信息的存储介质。

[0152] 在某些实施例中,计算机系统1200可以提供对执行一个或多个虚拟机的支持。计算机系统1200可以执行诸如管理程序之类的程序,以便促进虚拟机的配置和管理。每个虚拟机可以被分派存储器、计算(例如,处理器、内核)、I/O和联网资源。每个虚拟机通常运行其自己的操作系统,其可以与由计算机系统1200执行的其它虚拟机执行的操作系统相同或不同。相应地,多个操作系统可以潜在地由计算机系统1200并发地运行。每个虚拟机一般独立于其它虚拟机运行。

[0153] 通信子系统1224提供到其它计算机系统和网络的接口。通信子系统1224用作用于从计算机系统1200的其它系统接收数据和向其传送数据的接口。例如,通信子系统1224可以使计算机系统1200能够经由因特网建立到一个或多个客户端设备的通信信道,用于从客户端设备接收信息和发送信息到客户端设备。此外,通信子系统1224可以用于从特权账户管理器向发出请求的用户传递成功登录的通知或重新输入密码的通知。

[0154] 通信子系统1224可以支持有线和/或无线通信协议两者。例如,在某些实施例中,

通信子系统1224可以包括用于(例如,使用蜂窝电话技术、高级数据网络技术(诸如3G、4G或EDGE(全球演进的增强数据速率)、WiFi(IEEE 802.11族标准)、或其它移动通信技术、或其任意组合)接入无线语音和/或数据网络的射频(RF)收发器部件、全球定位系统(GPS)接收器部件和/或其它部件。在一些实施例中,作为无线接口的附加或替代,通信子系统1224可以提供有线网络连接(例如,以太网)。

[0155] 通信子系统1224可以以各种形式接收和传送数据。例如,在一些实施例中,通信子系统1224可以以结构化和/或非结构化的数据馈送1226、事件流1228、事件更新1230等形式接收输入通信。例如,通信子系统1224可以被配置为实时地从社交媒体网络的用户和/或其它通信服务(诸如**Twitter®**馈送、**Facebook®**更新、诸如丰富站点摘要(RSS)馈送的web馈送、和/或来自一个或多个第三方信息源的实时更新)接收(或发送)数据馈送1226。

[0156] 在某些实施例中,通信子系统1224可以被配置为以连续数据流的形式接收本质上可以是连续的或无界的没有明确结束的数据,其中连续数据流可以包括实时事件的事件流1228和/或事件更新1230。生成连续数据的应用的示例可以包括例如传感器数据应用、金融报价机、网络性能测量工具(例如网络监视和流量管理应用)、点击流分析工具、汽车流量监视等。

[0157] 通信子系统1224也可以被配置为向一个或多个数据库输出结构化和/或非结构化的数据馈送1226、事件流1228、事件更新1230等,其中所述一个或多个数据库可以与耦合到计算机系统1200的一个或多个流传输数据源计算机通信。

[0158] 计算机系统1200可以是各种类型中的一种,包括手持便携式设备(例如,**iPhone®**蜂窝电话、**iPad®**计算平板、PDA)、可穿戴设备(例如,Google **Glass®**头戴式显示器)、个人计算机、工作站、大型机、信息站、服务器机架或任何其它数据处理系统。

[0159] 由于计算机和网络不断变化的性质,对图12中描绘的计算机系统1200的描述旨在仅仅作为具体示例。具有比图12中所描绘的系统更多或更少部件的许多其它配置是可能的。基于本文所提供的公开内容和教导,本领域普通技术人员将理解实现各种实施例的其它方式和/或方法。

[0160] 一些附图中描绘的系统可以以各种配置提供。在一些实施例中,系统可以被配置为分布式系统,其中系统的一个或多个部件跨一个或多个云基础设施系统中的一个或多个网络而分布。

[0161] 云基础设施系统是一个或多个服务器计算设备、网络设备和/或存储设备的集合。这些资源可以由云服务提供商划分,并以某种方式分配给其客户。例如,云服务提供商(诸如加利福尼亚州Redwood Shores的Oracle Corporation)可以提供各种类型的云服务,包括但不限于在软件即服务(SaaS)类别下提供的一个或多个服务、在平台即服务(PaaS)类别下提供的服务、在基础设施即服务(IaaS)类别下提供的服务、或包括混合服务的其它类别的服务。SaaS服务的示例包括但不限于构建和递送一套按需应用(诸如Oracle Fusion应用)的能力。SaaS服务使客户能够利用在云基础设施系统上执行的应用,而无需客户为应用购买软件。PaaS服务的示例包括但不限于使组织(诸如Oracle)能够在共享的公共架构上整合现有应用的服务,以及构建充分利用由平台提供的共享服务(诸如Oracle Java云服务(JCS)、Oracle数据库云服务(DBCS)等)的新应用的能力。IaaS服务通常有助于管理和控制

底层计算资源，诸如存储、网络和其它基本计算资源，用于让客户利用由SaaS平台和PaaS平台提供的服务。

[0162] 图13是根据本公开实施例的系统环境1300的一个或多个部件的简化框图，通过该系统环境1300，由实施例系统的一个或多个部件提供的服务可以被提供为云服务。在所示实施例中，系统环境1300包括一个或多个客户端计算设备1304、1306和1308，其可以由用户用于与提供云服务的云基础设施系统1302交互。客户端计算设备可以被配置为操作可以由客户端计算设备的用户用来与云基础设施系统1302交互以使用由云基础设施系统1302提供的服务的客户端应用，诸如web浏览器、专有客户端应用(例如，Oracle Forms)或某个其它应用。

[0163] 应当认识到的是，图中描绘的云基础设施系统1302可以具有除所描绘的那些之外的其它部件。另外，图中所示的实施例仅仅是可以结合本公开实施例的云基础设施系统的一个示例。在一些其它实施例中，云基础设施系统1302可以具有比图中所示更多或更少的部件，可以组合两个或更多个部件，或者可以具有不同的部件配置或布置。

[0164] 客户端计算设备1304、1306和1308可以是与上面针对1102、1104、1106和1108描述的设备类似的设备。

[0165] 虽然示出了具有三个客户端计算设备的示例性系统环境1300，但是可以支持任何数量的客户端计算设备。诸如具有传感器的设备等之类的其它设备可以与云基础设施系统1302交互。

[0166] (一个或多个)网络1310可以促进客户端1304、1306和1308与云基础设施系统1302之间的数据通信和交换。每个网络可以是对本领域技术人员来说熟悉的、可以使用各种商业上可用的协议(包括上面针对(一个或多个)网络1610描述的那些协议)中的任何一种支持数据通信的任何类型的网络。

[0167] 云基础设施系统1302可以包括一个或多个计算机和/或服务器，其可以包括上面针对服务器1112描述的那些。

[0168] 在某些实施例中，由云基础设施系统提供的服务可以包括按需对云基础设施系统的用户可用的大量服务，诸如在线数据存储和备份解决方案、基于Web的电子邮件服务、托管的办公套件和文档协作服务、数据库处理、受管理的技术支持服务等。由云基础架构系统提供的服务可以动态扩展以满足其用户的需求。由云基础设施系统提供的服务的具体实例化在本文中被称为“服务实例”。一般而言，经由通信网络(诸如因特网)来自云服务提供者的系统使得对用户可用的任何服务被称为“云服务”。通常，在公共云环境中，构成云服务提供者的系统的服务器和系统与消费者自己的本地服务器和系统不同。例如，云服务提供者的系统可以托管应用，而用户可以经由诸如因特网的通信网络按需订购和使用该应用。

[0169] 在一些示例中，计算机网络云基础设施中的服务可以包括对存储装置、托管的数据库、托管的web服务器、软件应用或者由云供应商向用户提供的其它服务的受保护的计算机网络访问，或者如本领域中另外已知的。例如，服务可以包括通过因特网对云上的远程存储的受密码保护的访问。作为另一个示例，服务可以包括基于web服务的托管的关系数据库和脚本语言中间件引擎，用于由联网的开发者私人使用。作为另一个示例，服务可以包括对在云供应商的网站上托管的电子邮件软件应用的访问。

[0170] 在某些实施例中，云基础设施系统1302可以包括以自助服务、基于订阅、弹性可扩

展、可靠、高度可用和安全的方式交付给消费者的应用套件、中间件和数据库服务提供品。这种云基础设施系统的示例是由本受让人提供的Oracle Public Cloud(Oracle公共云)。

[0171] 在各种实施例中，云基础设施系统1302可以适于自动地供应、管理和跟踪消费者对由云基础设施系统1302提供的服务的订阅。云基础设施系统1302可以经由不同的部署模型提供云服务。例如，服务可以在公共云模型下提供，其中云基础设施系统1302由销售云服务的组织拥有(例如，由Oracle拥有)并且使服务对一般公众或不同的工业企业可用。作为另一个示例，服务可以在私有云模型下提供，其中云基础设施系统1302仅针对单个组织操作，并且可以为组织内的一个或多个实体提供服务。云服务还可以在社区云模型下提供，其中云基础设施系统1302和由云基础设施系统1302提供的服务由相关社区中的若干个组织共享。云服务还可以在混合云模型下提供，混合云模型是两个或更多个不同模型的组合。

[0172] 在一些实施例中，由云基础设施系统1302提供的服务可以包括在软件即服务(SaaS)类别、平台即服务(PaaS)类别、基础设施即服务(IaaS)类别、或包括混合服务的服务的其它类别下提供的一个或多个服务。消费者经由订阅订单可以订购由云基础设施系统1302提供的一个或多个服务。云基础设施系统1302然后执行处理，以提供消费者的订阅订单中的服务。

[0173] 在一些实施例中，由云基础设施系统1302提供的服务可以包括但不限于应用服务、平台服务和基础设施服务。在一些示例中，应用服务可以由云基础设施系统经由SaaS平台提供。SaaS平台可以被配置为提供属于SaaS类别的云服务。例如，SaaS平台可以提供在集成的开发和部署平台上构建和交付按需应用套件的能力。SaaS平台可以管理和控制用于提供SaaS服务的底层软件和基础设施。通过利用由SaaS平台提供的服务，消费者可以利用在云基础设施系统上执行的应用。消费者可以获取应用服务，而无需消费者购买单独的许可证和支持。可以提供各种不同的SaaS服务。示例包括但不限于为大型组织提供用于销售绩效管理、企业集成和业务灵活性的解决方案的服务。

[0174] 在一些实施例中，平台服务可以由云基础设施系统经由PaaS平台提供。PaaS平台可以被配置为提供属于PaaS类别的云服务。平台服务的示例可以包括但不限于使组织(诸如Oracle)能够在共享的共同架构上整合现有应用的服务，以及充分利用由平台提供的共享服务来构建新应用的能力。PaaS平台可以管理和控制用于提供PaaS服务的底层软件和基础设施。消费者可以获取由云基础设施系统提供的PaaS服务，而无需消费者购买单独的许可证和支持。平台服务的示例包括但不限于Oracle Java云服务(JCS)、Oracle数据库云服务(DBCS)以及其它。

[0175] 通过利用由PaaS平台提供的服务，消费者可以采用由云基础设施系统支持的编程语言和工具，并且还控制所部署的服务。在一些实施例中，由云基础设施系统提供的平台服务可以包括数据库云服务、中间件云服务(例如，Oracle Fusion Middleware服务)和Java云服务。在一个实施例中，数据库云服务可以支持共享服务部署模型，其使得组织能够汇集数据库资源并且以数据库云的形式向消费者提供数据库即服务。中间件云服务可以为消费者提供开发和部署各种业务应用的平台，以及Java云服务可以为消费者提供在云基础设施系统中部署Java应用的平台。

[0176] 可以由云基础设施系统中的IaaS平台提供各种不同的基础设施服务。基础设施服务促进管理和控制底层计算资源，诸如存储装置、网络和其它基本计算资源，以便消费者利

用由SaaS平台和PaaS平台提供的服务。

[0177] 在某些实施例中，云基础设施系统1302还可以包括基础设施资源1330，用于提供用来向云基础设施系统的消费者提供各种服务的资源。在一个实施例中，基础设施资源1330可以包括用以执行由PaaS平台和SaaS平台提供的服务的硬件(诸如服务器、存储装置和联网资源)的预先集成和优化的组合。

[0178] 在一些实施例中，云基础设施系统1302中的资源可以由多个用户共享并且按需动态地重新分派。此外，资源可以分派给在不同时区中的用户。例如，云基础设施系统1330可以使第一时区内的第一用户集合能够利用云基础设施系统的资源指定的小时数，然后使得能够将相同资源重新分派给位于不同时区中的另一用户集合，从而最大化资源的利用。

[0179] 在某些实施例中，可以提供由云基础设施系统1302的不同部件或模块以及由云基础设施系统1302提供的服务共享的多个内部共享服务1332。这些内部共享服务可以包括，但不限于，安全和身份服务、集成服务、企业储存库服务、企业管理器服务、病毒扫描和白名单服务、高可用性、备份和恢复服务、用于启用云支持的服务、电子邮件服务、通知服务、文件传输服务等。

[0180] 在某些实施例中，云基础设施系统1302可以在云基础设施系统中提供云服务(例如，SaaS、PaaS和IaaS服务)的综合管理。在一个实施例中，云管理功能可以包括用于供应、管理和跟踪由云基础设施系统1302等接收到的消费者的订阅的能力。

[0181] 在一个实施例中，如图中所描绘的，云管理功能可以由诸如订单管理模块1320、订单编排模块1322、订单供应模块1324、订单管理和监视模块1326以及身份管理模块1328的一个或多个模块提供。这些模块可以包括一个或多个计算机和/或服务器或可以使用一个或多个计算机和/或服务器来提供，该一个或多个计算机和/或服务器可以是通用计算机、专用服务器计算机、服务器场，服务器集群或任何其它适当的布置和/或组合。

[0182] 在示例性操作1334中，使用客户端设备(诸如客户端设备1304、1306或1308)的客户可以通过请求由云基础设施系统1302提供的一个或多个服务并且对由云基础设施系统1302提供的一个或多个服务的订阅下订单来与云基础设施系统1302交互。在某些实施例中，消费者可以访问云用户界面(UI)，云UI 1312、云UI 1314和/或云UI 1316，并经由这些UI下订阅订单。响应于消费者下订单而由云基础设施系统1302接收到的订单信息可以包括识别消费者和消费者打算订阅的由云基础设施系统1302提供的一个或多个服务的信息。

[0183] 在客户下订单之后，经由云UI，1312、1314和/或1316，接收订单信息。

[0184] 在操作1336处，订单存储在订单数据库1318中。订单数据库1318可以是由云基础设施系统1318操作并且与其它系统元件一起操作的若干数据库之一。

[0185] 在操作1338处，订单信息被转发到订单管理模块1320。在一些实例中，订单管理模块1320可以被配置为执行与订单相关的计费和记帐功能，诸如验证订单，并且一经验证，预订订单。

[0186] 在操作1340处，将关于订单的信息传送到订单编排模块1322。订单编排模块1322可以利用订单信息来编排用于由客户下的订单的服务和资源的供应。在一些实例中，订单编排模块1322可以使用订单供应模块1324的服务来编排资源的供应以支持订阅的服务。

[0187] 在某些实施例中，订单编排模块1322使得能够管理与每个订单相关联的业务处理并应用业务逻辑，以确定订单是否应当前进到供应。在操作1342处，一经接收到对新订阅的

订单，订单编排模块1322向订单供应模块1324发送请求以分派资源并配置履行订阅的订单所需的那些资源。订单供应模块1324使得能够为客户订阅的服务分派资源。订单供应模块1324提供由云基础设施系统1300提供的云服务与用于供应用于提供所请求服务的资源的物理实现层之间的抽象级别。因此，订单编排模块1322可以与实现细节隔离，诸如服务和资源实际上是实时供应的还是预先供应并且仅在请求时被分派/分配。

[0188] 在操作1344处，一旦供应了服务和资源，就可以通过云基础设施系统1302的订单供应模块1324向客户端设备1304、1306和/或1308上的客户发送所提供服务的通知。在操作1346处，客户的订阅订单可以由订单管理和监视模块1326管理和跟踪。在一些情况下，订单管理和监视模块1326可以被配置为收集订阅订单中的服务的使用统计数据，诸如所使用的存储的量、传递的数据量、用户的数量、系统运行时间和系统停机时间的量。

[0189] 在某些实施例中，云基础设施系统1300可以包括身份管理模块1328。身份管理模块1328可以被配置为在云基础设施系统1300中提供身份服务，诸如访问管理和授权服务。在一些实施例中，身份管理模块1328可以控制关于希望利用由云基础设施系统1302提供的服务的消费者的信息。这种信息可以包括认证这些消费者的身份的信息和描述那些消费者被授权相对于各种系统资源(例如，文件、目录、应用、通信端口、存储器段等)执行的动作的信息。身份管理模块1328还可以包括关于每个消费者的描述性信息以及关于描述性信息可以如何和由谁来访问和修改的管理。

[0190] 虽然已经描述了本公开的具体实施例，但是各种修改、变更、替代构造和等同物也包含在本公开的范围内。本公开的实施例不限于在某些特定数据处理环境内的操作，而是可以在多个数据处理环境内自由操作。此外，虽然已使用特定系列的事务和步骤描述了本公开的实施例，但是，对本领域技术人员应当显而易见的是，本公开的范围不限于所描述系列的事务和步骤。上述实施例的各种特征和方面可以被单独或结合使用。

[0191] 另外，虽然已经使用硬件和软件的特定组合描述了本公开的实施例，但是应该认识到的是，硬件和软件的其它组合也在本公开的范围内。本公开的实施例可以只用硬件、或只用软件、或利用其组合来实现。本文描述的各种进程可以在同一处理器或以任何组合的不同处理器上实现。相应地，在部件或模块被描述为被配置为执行某些操作的情况下，这种配置可以例如通过设计电子电路来执行操作、通过对可编程电子电路(诸如微处理器)进行编程来执行操作、或其任意组合来实现。进程可以利用各种技术来通信，包括但不限于用于进程间通信的常规技术，并且不同的进程对可以使用不同的技术，或者同一对进程可以在不同时间使用不同的技术。

[0192] 因而，说明书和附图应当在说明性而不是限制性的意义上考虑。但是，将明显的是，在不背离权利要求中阐述的更广泛精神和范围的情况下，可以对其进行添加、减少、删除和其它修改和改变。因此，虽然已描述了具体的公开实施例，但是这些实施例不旨在进行限制。各种修改和等同物都在以下权利要求的范围之内。

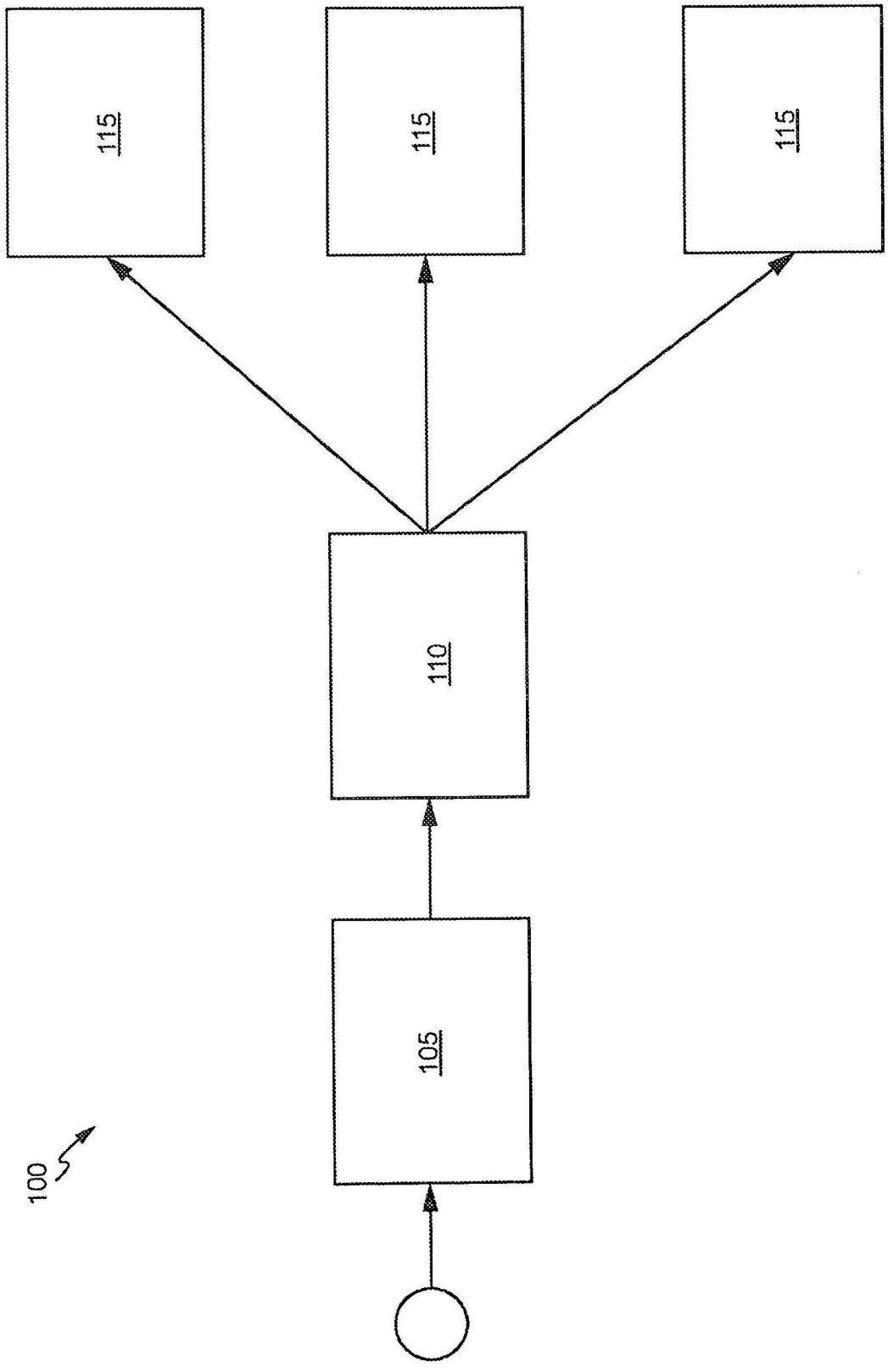


图1

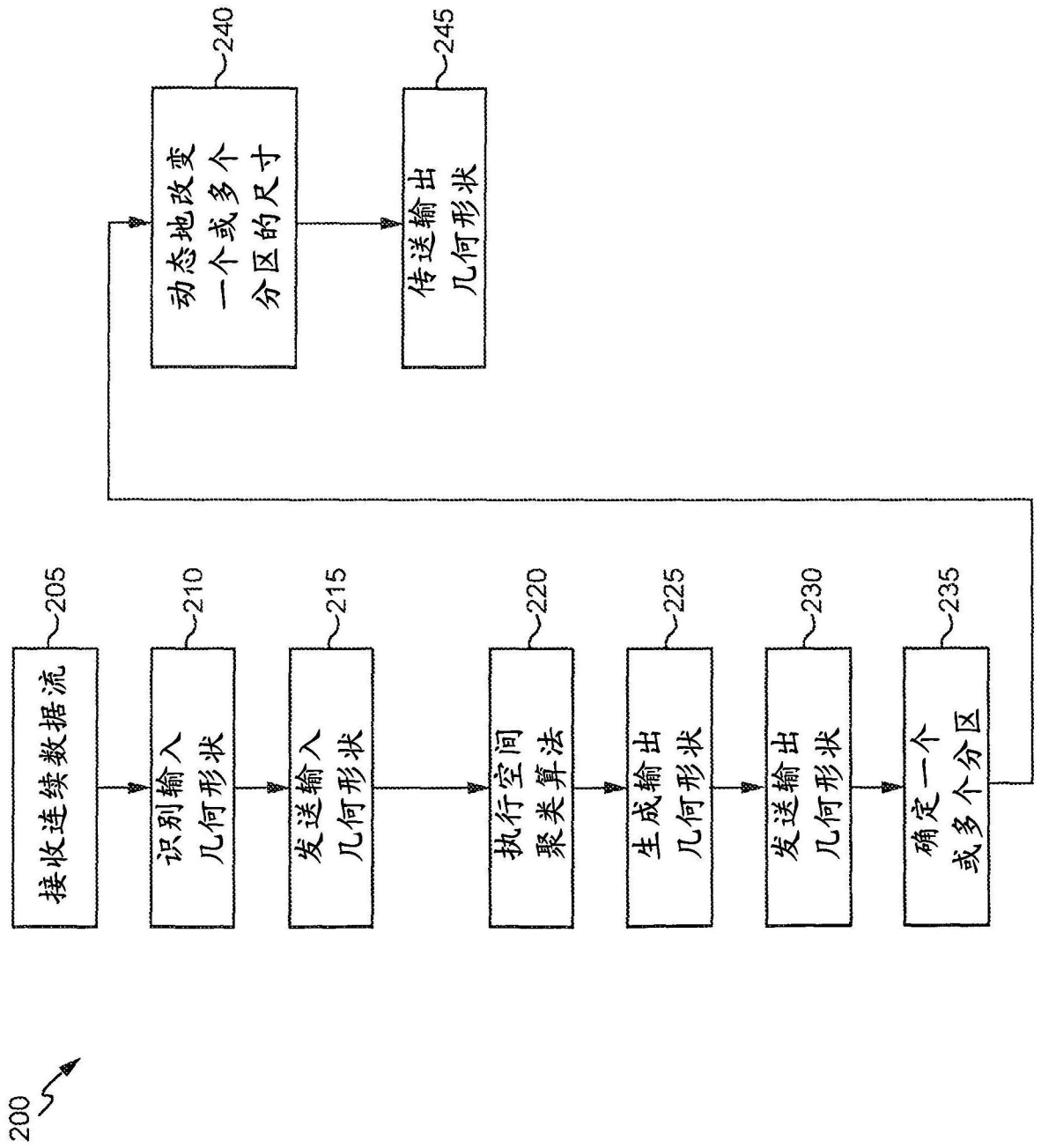

200
205
接收连续数据流
210
识别输入几何形状
215
发送输入几何形状
220
执行空间聚类算法
225
生成输出几何形状
230
发送输出几何形状
235
确定一个或多个分区
240
动态地改变一个或多个分区的尺寸
245
传送输出几何形状

图2

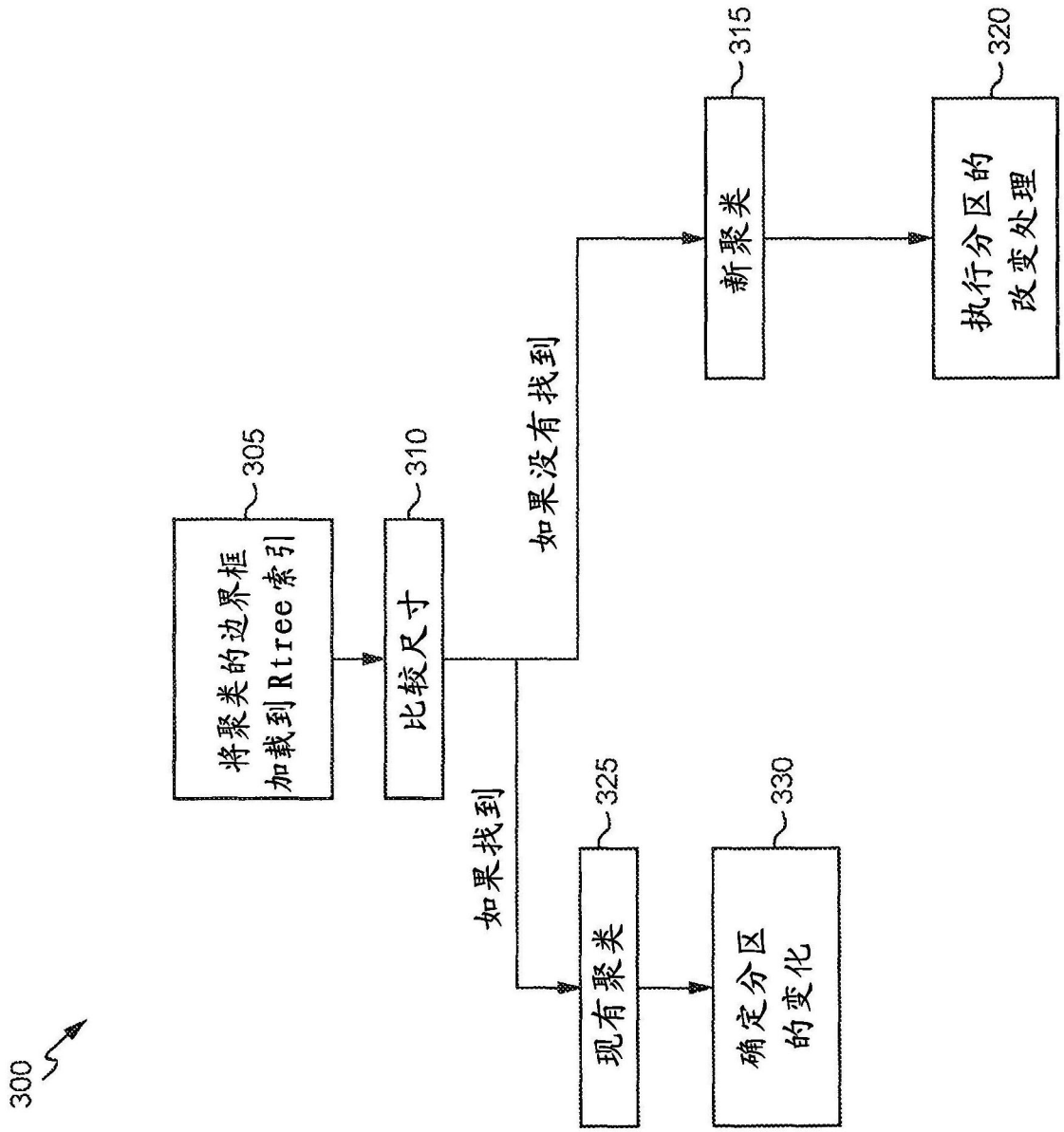

300
将聚类的边界框加载到Rtree索引
305
比较尺寸
310
如果找到
如果没有找到
现有聚类
325
确定分区的变化
330
新聚类
315
执行分区的改变处理
320

图3

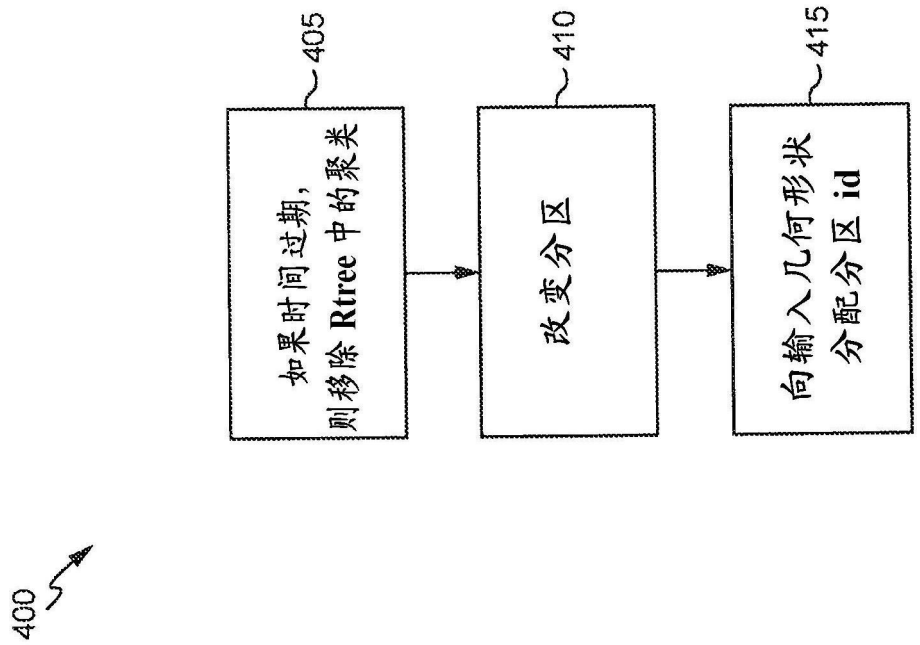


图4

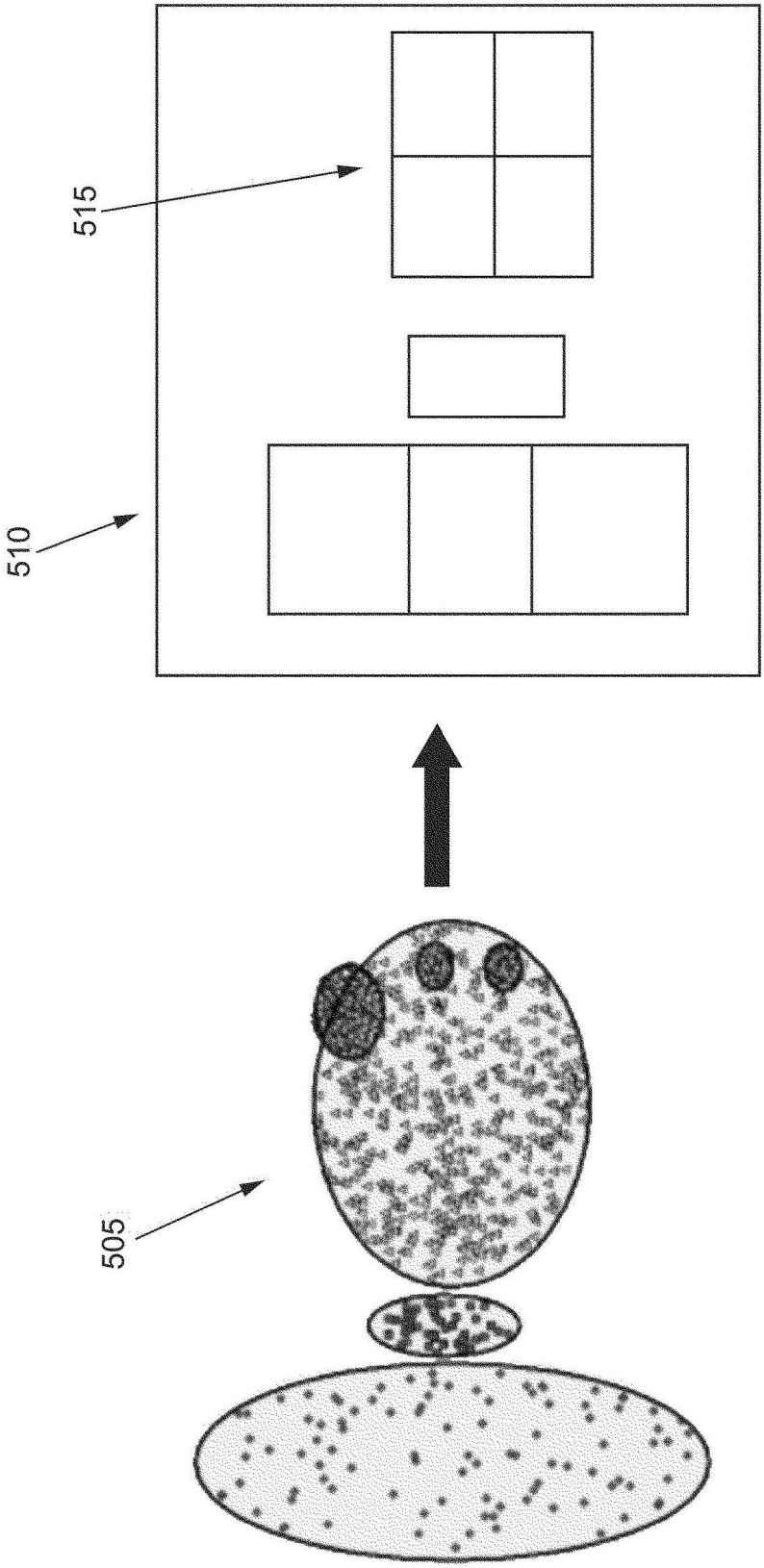


图5

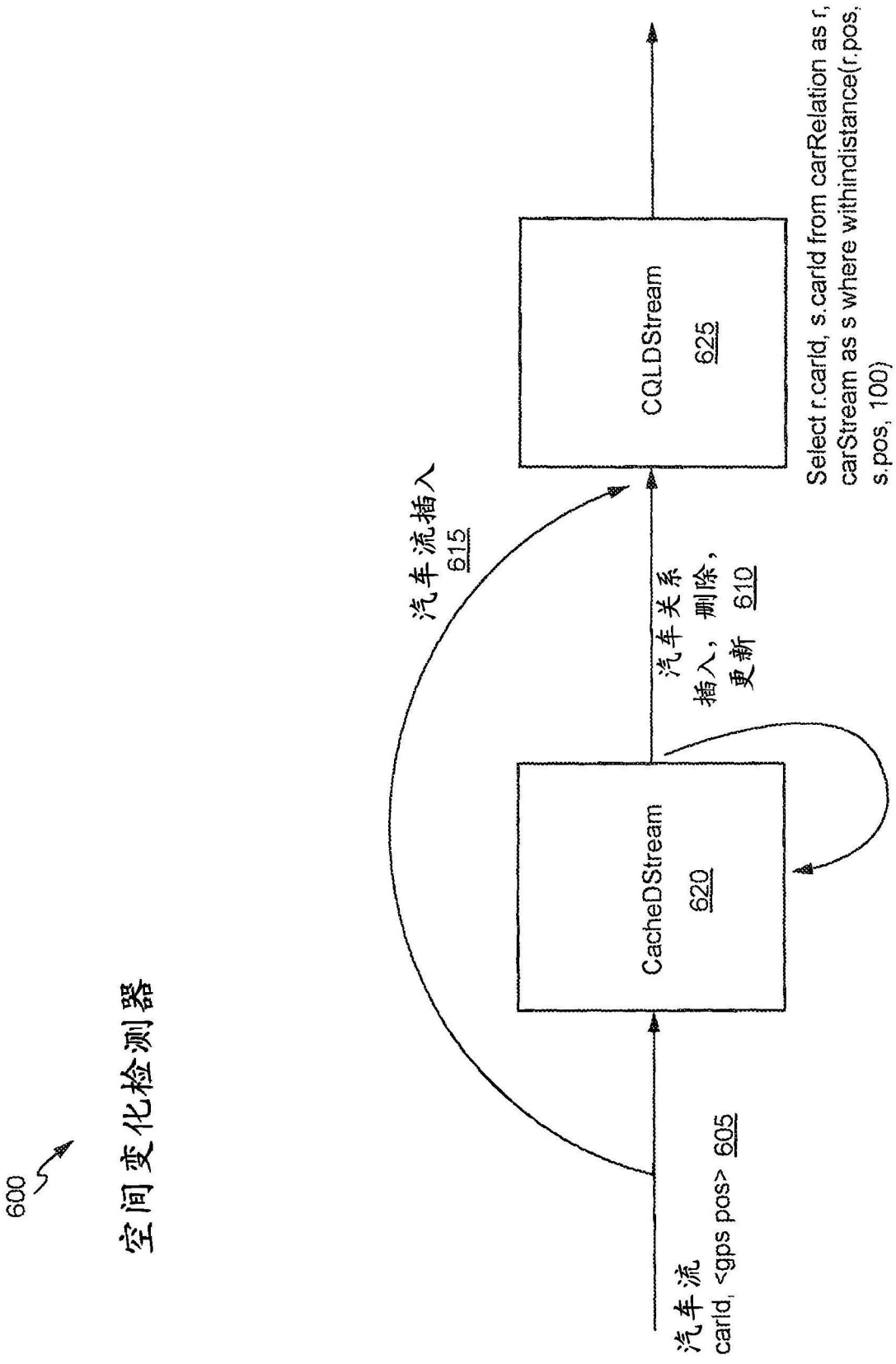

600
空间变化检测器
汽车流
carId, <gps pos> 605
CacheDStream
620
汽车关系
插入，删除，
更新 610
汽车流插入
615
CQLDStream
625
Select r.carId, s.carId from carRelation as r,
carStream as s where withindistance(r.pos,
s.pos, 100)

图6

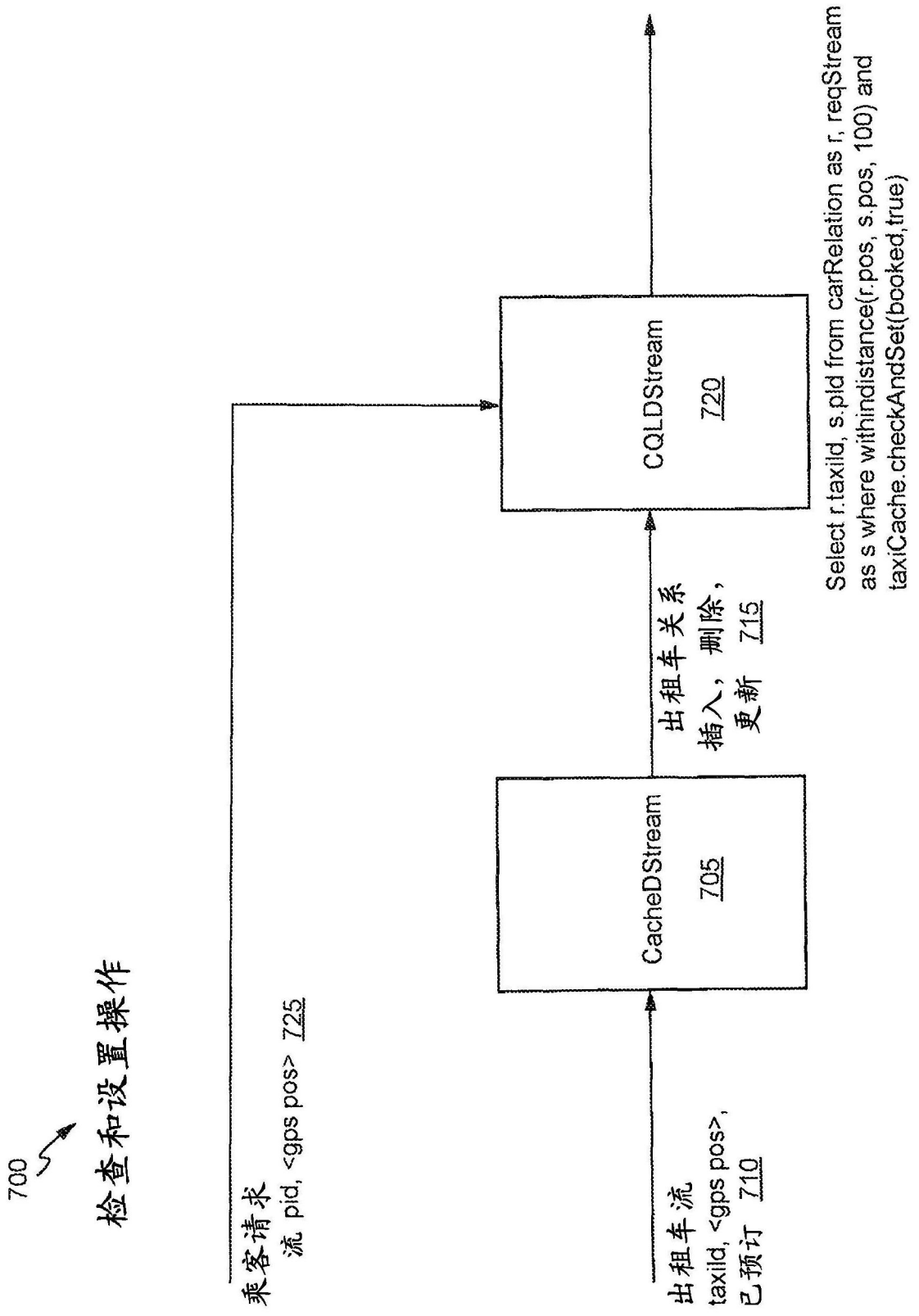

700
检查和设置操作
乘客请求
流 pid, <gps pos> 725
出租车流
taxiId, <gps pos>,
已预订 710
CacheDStream
705
出租车关系
插入，删除，
更新 715
CQLDStream
720
Select r.taxiId, s.pId from carRelation as r, reqStream as s where withindistance(r.pos, s.pos, 100) and taxiCache.checkAndSet(booked,true)

图7

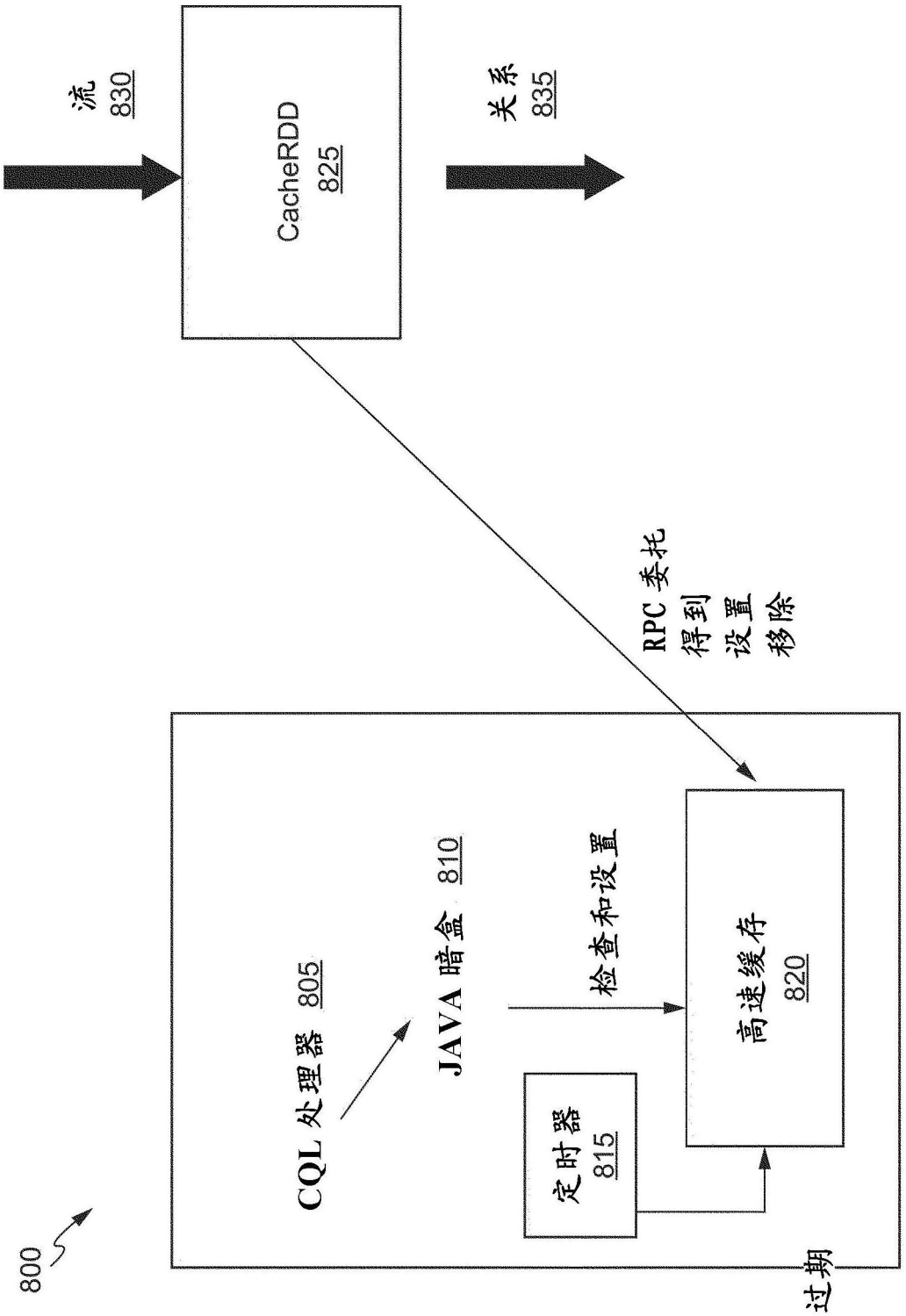

流
830
CacheRDD
825
关系
835
RPC 委托
得到
设置
移除
CQL 处理器 805
JAVA 暗盒 810
检查和设置
定时器
815
高速缓存
820
800
过期

图8

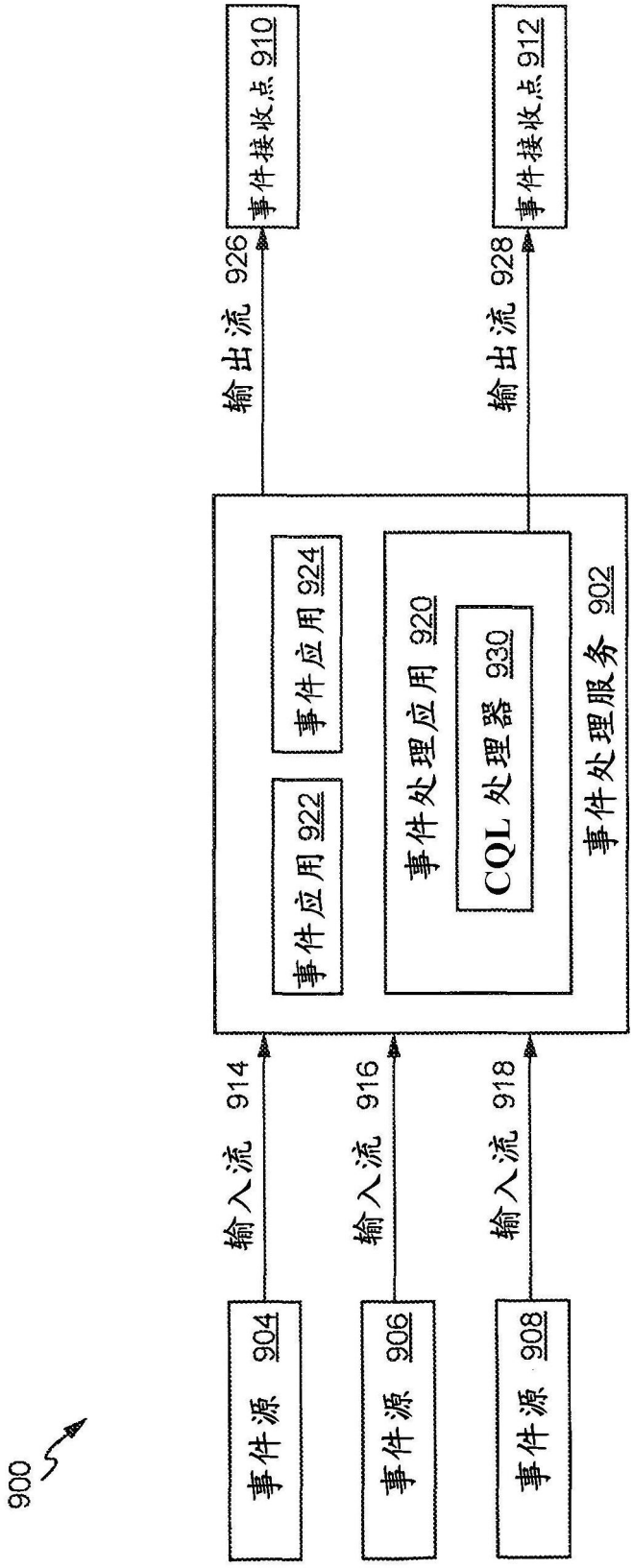

900
事件源 904
事件源 906
事件源 908
输入流 914
输入流 916
输入流 918
事件处理服务 902
事件应用 922
事件应用 924
事件处理应用 920
CQL 处理器 930
输出流 926
输出流 928
事件接收点 910
事件接收点 912

图9

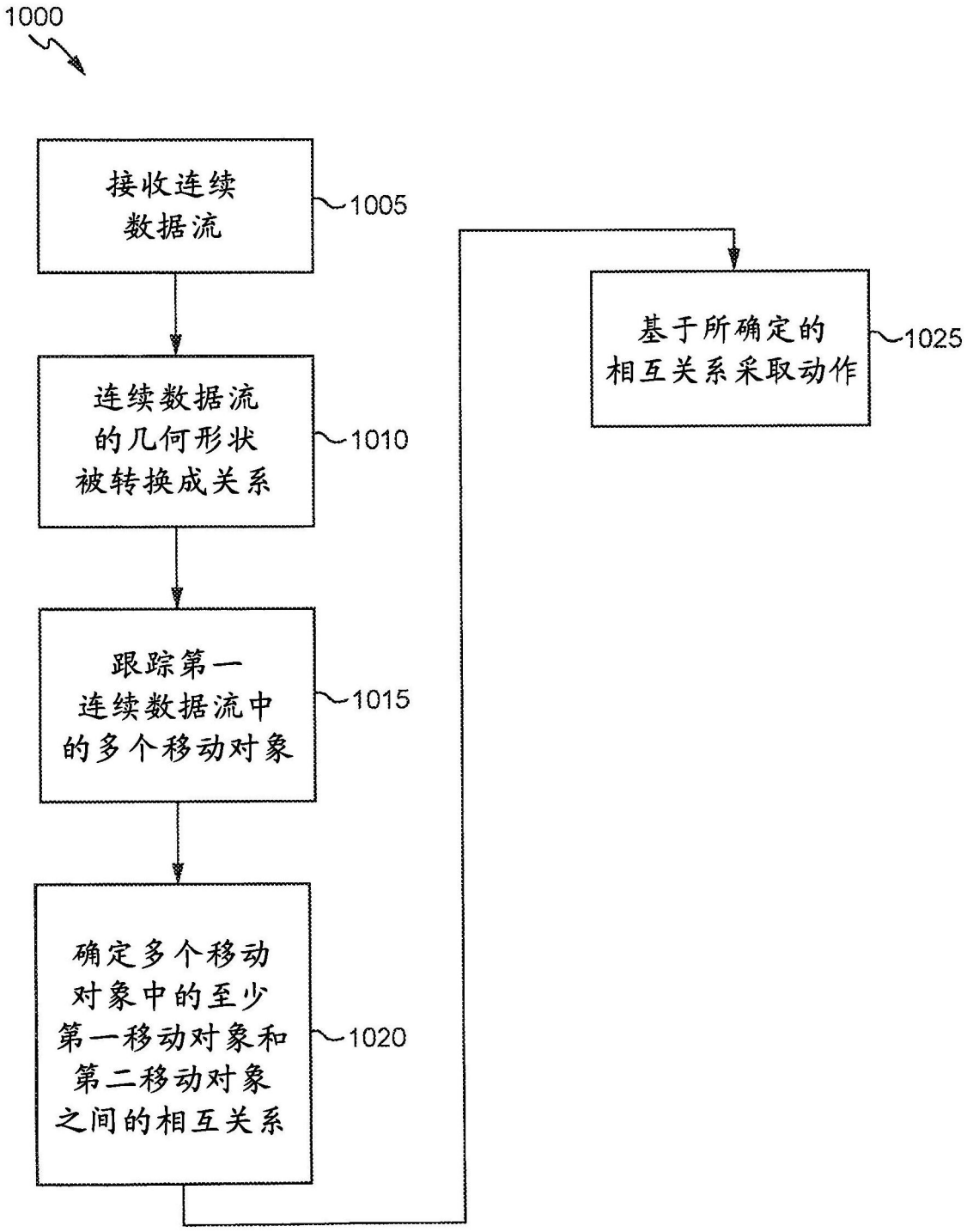

1000
接收连续数据流
1005
连续数据流的几何形状被转换成关系
1010
跟踪第一连续数据流中的多个移动对象
1015
确定多个移动对象中的至少第一移动对象和第二移动对象之间的相互关系
1020
基于所确定的相互关系采取动作
1025

图10

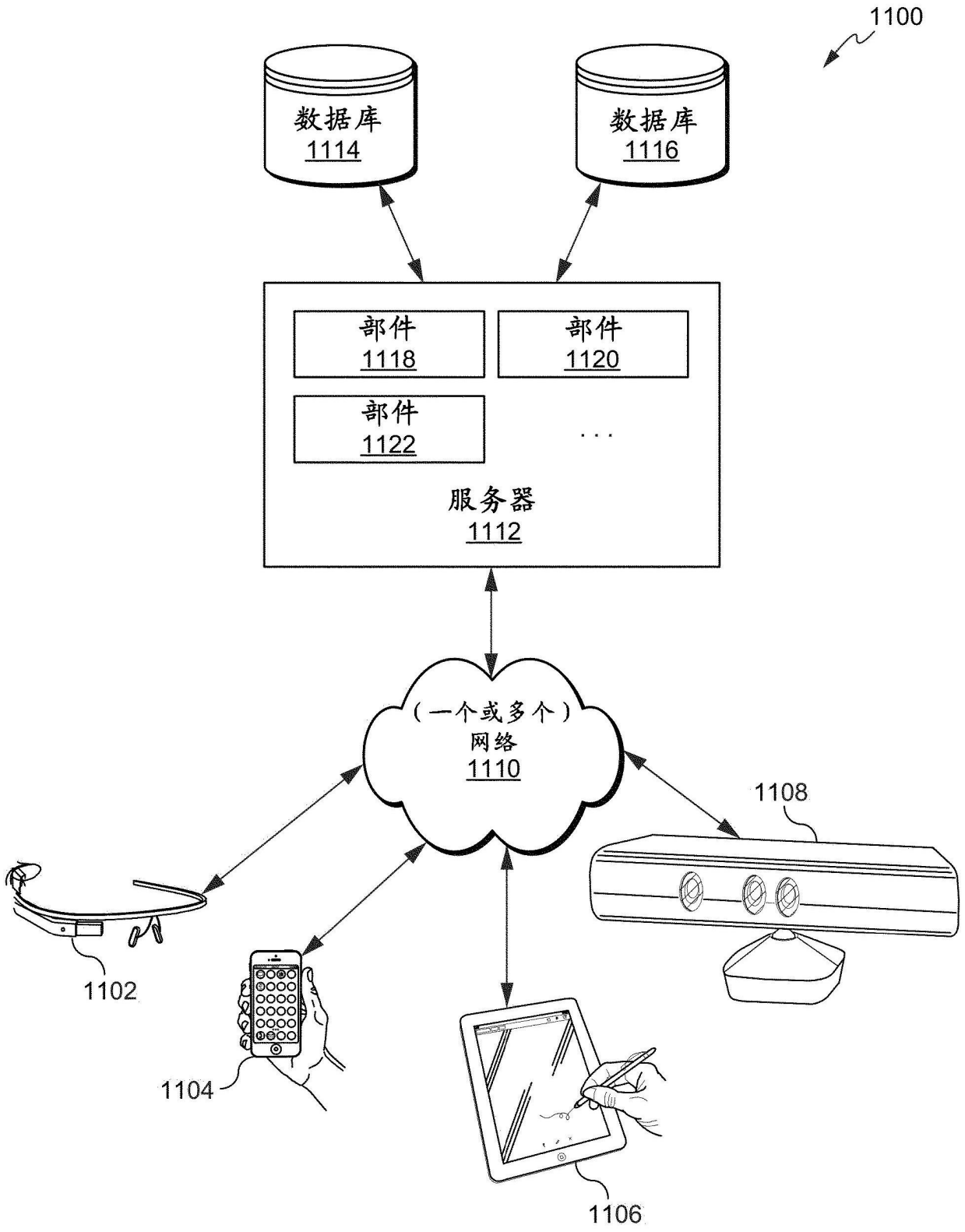

1100
数据库
1114
数据库
1116
部件
1118
部件
1120
部件
1122
. . .
服务器
1112
（一个或多个）
网络
1110
1108
1102
1104
1106

图11

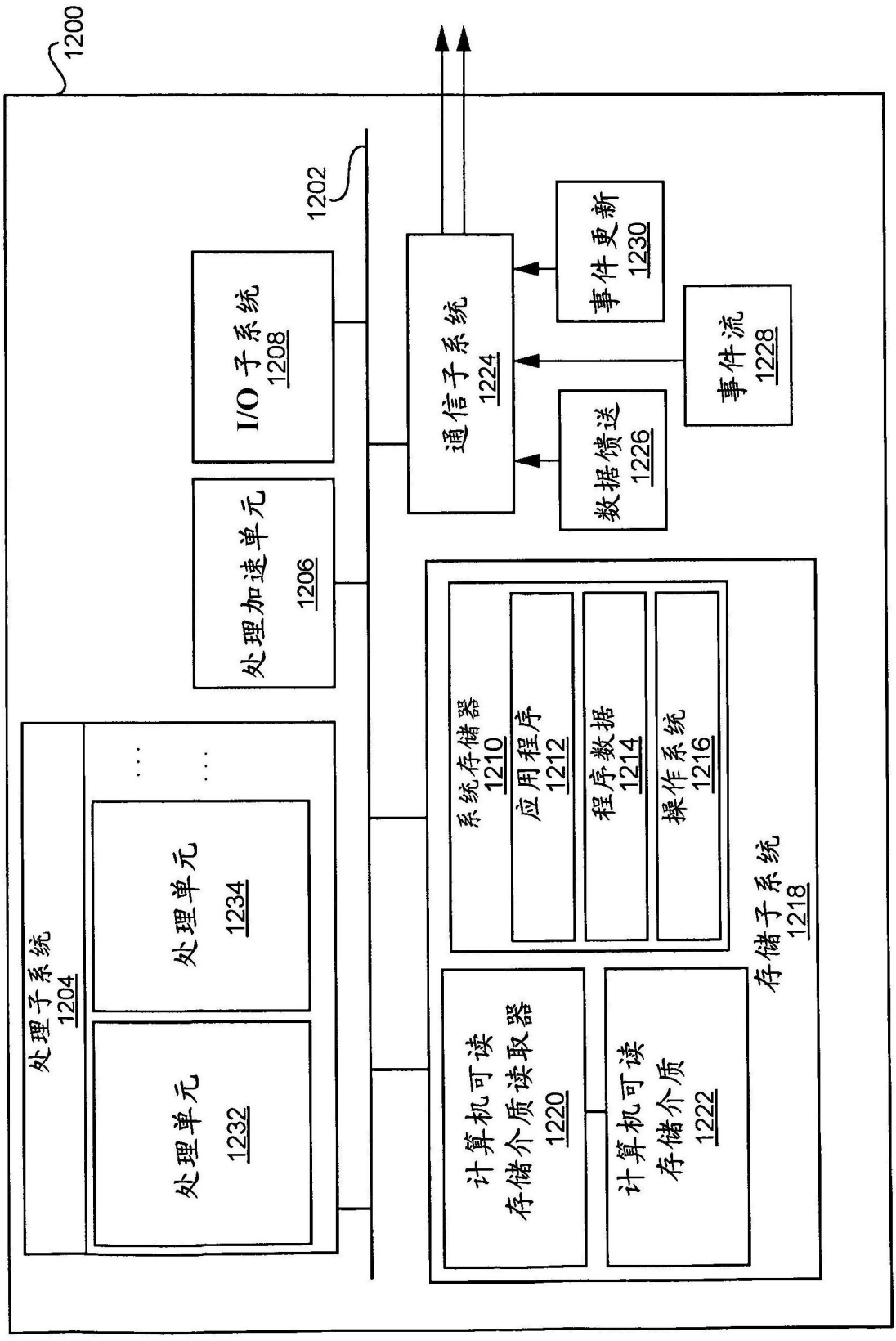

1200
1202
处理子系统
1204
处理单元
1232
处理单元
1234
处理加速单元
1206
I/O 子系统
1208
通信子系统
1224
事件更新
1230
数据馈送
1226
事件流
1228
存储子系统
1218
系统存储器
1210
应用程序
1212
程序数据
1214
操作系统
1216
计算机可读
存储介质读取器
1220
计算机可读
存储介质
1222

图12

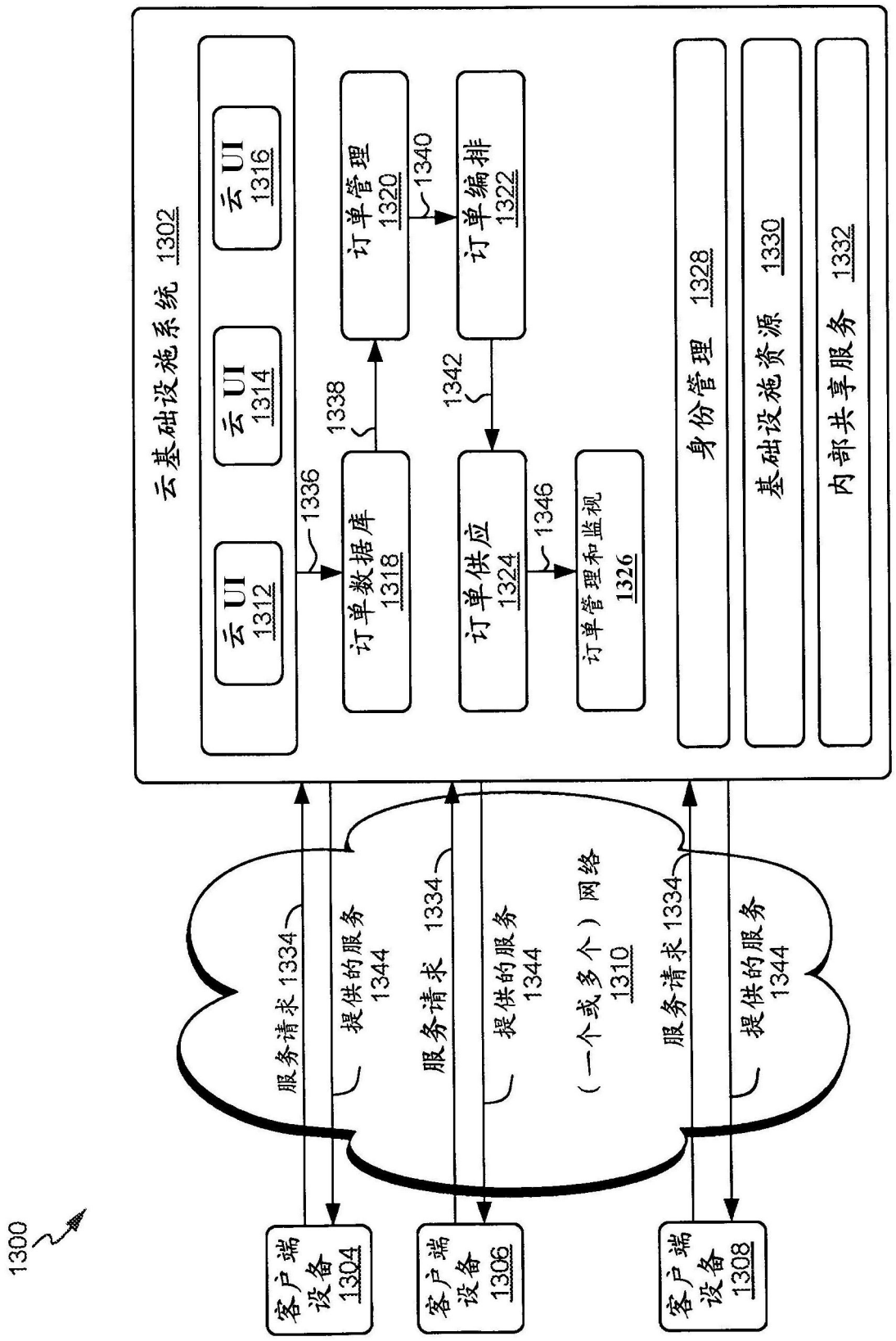

1300
云基础设施系统 1302
云 UI 1312
云 UI 1314
云 UI 1316
订单数据库 1318
订单管理 1320
订单编排 1322
订单供应 1324
订单管理和监视 1326
身份管理 1328
基础设施资源 1330
内部共享服务 1332
1336
1338
1340
1342
1346
服务请求 1334
提供的服务 1344
（一个或多个）网络 1310
客户端设备 1304
客户端设备 1306
客户端设备 1308

图13