(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 104903894 B
(45)授权公告日 2018.12.28

(21)申请号 201380069759.9
(22)申请日 2013.12.20
(65)同一申请的已公布的文献号
申请公布号 CN 104903894 A
(43)申请公布日 2015.09.09
(30)优先权数据
13198563.2 2013.12.19 EP
13/735,820 2013.01.07 US
(85)PCT国际申请进入国家阶段日
2015.07.07
(86)PCT国际申请的申请数据
PCT/US2013/077240 2013.12.20
(87)PCT国际申请的公布数据
WO2014/107359 EN 2014.07.10
(73)专利权人 脸谱公司
地址 美国加利福尼亚州
(72)发明人 拉戈特姆·穆尔蒂 拉贾特·格尔
(74)专利代理机构 北京康信知识产权代理有限责任公司 11240
代理人 梁丽超 王红艳
(51)Int.Cl.
G06F 17/30(2006.01)
(56)对比文件
US 2003074352 A1,2003.04.17,
US 5987449 A,1999.11.16,
US 2004103087 A1,2004.05.27,
CN 101067823 A,2007.11.07,
US 2011082859 A1,2011.04.07,
US 2012054182 A1,2012.03.01,
WO 2005076160 A1,2005.08.18,
CN 101093501 A,2007.12.26,
US 7984043 B1,2011.07.19,
US 2011228668 A1,2011.09.22,
T Condie等.MapReduce online.《NSDI"10 Proceedings of the 7th USENIX conference on Networked systems design and implementation》.2010,第1-15页.
审查员 曾伟

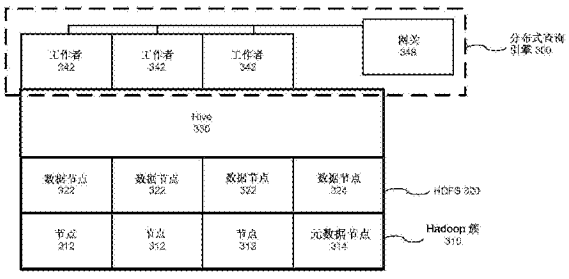
权利要求书3页 说明书12页 附图7页

(54)发明名称

用于分布式数据库查询引擎的系统和方法

(57)摘要

在本文中公开了用于能够执行低延迟数据库查询处理的技术。所述系统包括网关服务器和多个工作者节点。所述网关服务器被配置为将包含存储在具有多个数据节点的分布式存储簇内的数据的数据库的数据库查询分成多个局部查询，并且根据多个中间结果构造查询结果。多个工作者节点中的每个工作者节点被配置为通过扫描与存储在分布式存储簇的至少一个数据节点上的相应局部查询相关的数据，来处理多个局部查询的相应局部查询，并且生成存储在工作者节点的存储器内的多个中间结果的一个中间结果。



CN 104903894 B

1.一种用于分布式数据库查询引擎的系统,包括:

网关服务器,被配置为从包含存储在具有多个数据节点的分布式存储簇内的数据的数据库的数据库查询中生成多个局部查询,并且根据多个中间结果构建查询结果;以及

多个工作者节点,其中,所述多个工作者节点中的每个工作者节点被配置为通过扫描与相应局部查询相关并且存储在分布式存储簇的至少一个数据节点上的数据,来处理所述多个局部查询中的相应局部查询,并且其中,所述多个工作者节点中的每个工作者节点进一步被配置为生成所述多个中间结果中的存储在所述工作者节点的存储器内的一个中间结果,

其中,所述网关服务器进一步被配置为识别离散的工作者节点,进一步将分配给所述离散的工作者节点的局部查询分成多个从属局部查询,并且将所述多个从属局部查询分配给所述多个工作者节点中的一些工作者节点,其中,所述离散的工作者节点是未向所述网关服务器报告进度或者在预定的时间段之后向网关服务器报告低于预定值的进度的工作者节点。

2.根据权利要求1所述的系统,其中,所述多个工作者节点中的每个工作者节点进一步被配置为通过扫描与存储在分布式存储簇的至少一个数据节点上的相应局部查询相关的一部分数据,来处理所述多个局部查询的相应局部查询,并且生成存储在所述工作者节点的存储器内的近似中间结果;

其中,所述网关服务器进一步被配置为根据至少一个近似中间结果,构建近似查询结果。

3.根据权利要求1或2所述的系统,其中,所述网关服务器进一步被配置为根据所述多个中间结果的一部分,构建近似查询结果。

4.根据权利要求1到2中任一项所述的系统,其中,所述多个工作者节点中的每个工作者节点是运行分布式存储簇内的相应数据节点的服务。

5.根据权利要求1到2中任一项所述的系统,进一步包括:

元数据缓存,被配置为缓存数据库的表格级元数据以及分布式存储簇的文件级元数据;

其中,所述元数据缓存被配置为保持来自用于所述数据库查询的前一数据库查询的缓存的元数据。

6.根据权利要求1到2中任一项所述的系统,其中,所述多个工作者节点中的每个工作者节点将心跳消息定期发送给网关服务器,以通过所述工作者节点报告局部查询处理的状态。

7.根据权利要求1到2中任一项所述的系统,其中,所述网关服务器进一步被配置为从客户端装置接收指令,以返回近似查询结果或终止数据库查询的处理。

8.根据权利要求1到2中任一项所述的系统,其中,所述网关服务器进一步被配置为指示工作者节点立即返回近似中间结果,并且根据近似中间结果,将近似查询结果返回至客户端装置。

9.根据权利要求1到2中任一项所述的系统,其中,所述数据库查询包括近似查询结果的请求。

10.根据权利要求1到2中任一项所述的系统,其中,所述查询结果伴有存储在数据节点

内的已经为所述查询结果扫描的一部分相关数据的指示。

11.根据权利要求1到2中任一项所述的系统，其中，所述数据库是Hive数据仓库系统，并且所述分布式存储簇是Hadoop簇。

12.一种用于分布式数据库查询引擎的方法，其中，使用根据权利要求1到11中任一项所述的系统。

13.一种用于分布式数据库查询引擎的方法，包括：

从客户端装置中接收包含存储在具有多个数据节点的分布式存储簇内的数据的数据库的数据库查询；

将所述数据库查询分成多个局部查询；

将每个局部查询发送给多个工作者节点中的相应工作者节点，其中，每个工作者节点是在所述分布式存储簇的数据节点上运行的服务；

从工作者节点中检索局部查询的多个中间结果，其中，通过扫描存储在运行相应工作者节点的数据节点内的相关数据，每个中间结果由工作者节点的相应工作者节点处理；以及

根据所述多个中间结果，生成查询结果，

所述方法进一步包括：

对于每个局部查询，检索关于哪个数据节点存储与局部查询相关的数据的元数据；

并且所述发送步骤包括：

根据所述元数据，将每个局部查询发送给多个工作者节点中的相应工作者节点。

14.根据权利要求13所述的方法，进一步包括：

将查询结果以及部分指示符返回至客户端装置，其中，所述部分指示符指示存储在数据节点内的已经为所述查询结果扫描的一部分相关数据。

15.根据权利要求13所述的方法，进一步包括：

指示工作者节点立即返回近似查询结果；

并且其中，所述检索步骤包括：

从工作者节点检索所述局部查询的多个近似中间结果，其中，通过扫描存储在运行相应工作者节点的数据节点内的一部分相关数据，每个近似中间结果由工作者节点的相应工作者节点处理。

16.一种用于分布式数据库查询引擎的方法，包括：

从客户端装置接收包含存储在具有多个数据节点的分布式存储簇内的数据的数据库的数据库查询；

将数据库查询分成多个局部查询；

将每个局部查询发送给多个工作者节点中的相应工作者节点，其中，每个工作者节点是在所述分布式存储簇的数据节点上运行的服务；

识别离散的工作者节点，将分配给所述离散的工作者节点的局部查询分成多个从属局部查询，并且将所述多个从属局部查询分配给所述多个工作者节点中的一些工作者节点；

从工作者节点检索局部查询的多个中间结果，其中，通过扫描存储在运行相应工作者节点的数据节点内的相关数据，每个中间结果由工作者节点中的相应工作者节点处理；以及

根据多个中间结果，生成查询结果。

17.根据权利要求16所述的方法，其中，所述识别步骤包括：

通过监控工作者节点定期发送的心跳消息，识别离散的工作者节点，其中，在预定的时间段内没有接收到来自离散的工作者节点的心跳消息时，或者在接收到来自离散的工作者节点的心跳消息并且所述心跳消息包括低于阈值的表示离散的工作者节点的局部查询处理的状态的数量时，识别所述离散的工作者节点。

用于分布式数据库查询引擎的系统和方法

[0001] 交叉引用相关申请

[0002] 本申请要求于2013年1月7日提交的美国专利申请号13/735,820的优先权，该申请之全文并入本文中，以作参考。

[0003] 本申请要求于2013年12月19日提交的欧洲专利申请号13198563.2的优先权，该申请之全文并入本文中，以作参考。

技术领域

[0004] 本发明总体上涉及数据库，并且尤其涉及用于低查询延迟数据库分析的分布式数据库查询引擎。

背景技术

[0005] 计算机和网络计算的发展引起了需要大量数据存储的应用程序。例如，数千万用户可以创建网页并且将图像和文本上传到社会媒体网站中。因此，社会媒体网站每天可以累积大量数据，因此，需要一种用于存储和处理数据的高度可扩展的系统(scalable system)。存在促进这样的大量数据存储的多种工具。

[0006] 存在框架，通过使得应用程序能够与成千台计算机的簇(也称为节点)以及千兆字节的数据交互，这些框架支持大规模数据密集型分布式应用程序。例如，称为Hadoop的框架使用分布式、可扩展的、便携式文件系统，称为Hadoop分布式文件系统(HDFS)，用于在Hadoop簇中在数据节点(也称为子节点)之中分布大量数据。为了减少数据节点电力故障或网络故障(包括开关故障)的不利影响，通常在不同的数据节点上复制HDFS内的数据。

[0007] 开发了Hive(一种开放源数据仓库系统)，用于在Hadoop簇的顶部上运行。Hive支持以脚本查询语言(SQL)(像称为HiveQL的声明式语言)表示的数据查询。然后，Hive系统将以HiveQL表示的查询编译成可以在Hadoop簇上执行的映射-归约(map-reduce)工作，具有有向非循环图的数学形式。HiveQL语言包括支持包含原始类型、集合(例如，阵列和映射)以及嵌套布局类型(nested compositions of types)的表格的类型系统。此外，Hive系统包括包含方案和统计的称为Hive-元数据存储的系统目录，可用于数据探索(data exploration)和查询优化中。

[0008] 与Hadoop簇耦合的Hive系统可以为社会网络系统存储和分析大量数据。例如，Hive系统可以分析在用户之间的连接程度，以对用户在社会网络系统上的历史进行分类。Hive系统可以分析活动日志，以了解社会网络系统的服务如何被用来帮助应用程序开发员、网页管理员以及广告人员做出开发和业务决定。Hive系统可以运行复杂的数据挖掘程序，以优化向社会网络系统的用户示出的广告。Hive系统可以进一步分析使用日志，以识别社会网络系统的垃圾邮件和滥用。

[0009] Hive系统包括供没有制作和执行Hive查询的编程能力的人使用的网络工具，用于制作、调试以及调度复杂的数据管线(data pipeline)，并且用于根据存储在Hive系统和其他关系数据库(例如，MySQL和Oracle)内的数据，生成报告。

[0010] 然而，Hive系统的查询延迟通常较高。由于大量数据以及Hadoop簇的map-reduce方案，甚至最简单的查询可能需要花费几秒到几分钟来完成。这对于在操作人员需要当前查询的结果来决定一系列查询中的下一个查询时的交互式分析尤其是个问题。由于在等待当前查询的结果时，分析人员不能确定下一个查询，所以延迟问题明显影响分析人员的生产力。

[0011] 一种可能的变通方案(workaround solution)是创建数据管线，这些数据管线将聚集数据从Hive载入其他类型的关系数据库管理系统(RDBMS)，例如，MySQL和Oracle。然后，操作人员执行交互式分析，并且使用这些RDBMS建立报告。然而，每个RDBMS需要单独的数据管线。数据管线也需要时间来将聚集数据从Hive传输给其他RDBMS。因此，这种变通方案处理依然麻烦并且不方便。

发明内容

[0012] 在本文中介绍的技术在存储在大规模存储簇(例如，Hadoop簇)内的非常大量的数据上提供了低延迟查询的优点，该簇在系统目录(例如，Hive元数据存储)内存储元数据。尤其地，在本文中介绍的技术包括基于服务树计算框架的分布式查询引擎。分布式查询引擎包括网关服务器和多个工作者节点。所述网关服务器将查询任务分成局部任务。引擎的每个工作者节点处理一个局部任务，以在存储器中生成中间查询结果。中间查询结果可以是通过扫描一部分相关数据所生成的近似中间结果。网关服务器接收中间查询结果，并且根据中间查询结果，构建查询任务的查询结果。

[0013] 因此，根据在本文中介绍的技术，提供了用于处理数据库的数据库查询的系统。该系统包括网关服务器和多个工作者节点。所述网关服务器被配置为将包含存储在具有多个数据节点的分布式存储簇内的数据的数据库的数据库查询分成多个局部查询，并且根据多个中间结果构建查询结果。所述多个工作者节点中的每个工作者节点被配置为通过扫描与存储在分布式存储簇中的至少一个数据节点上的相应局部查询相关的数据，来处理多个局部查询的相应局部查询，并且生成多个中间结果中的存储在所述工作者节点的存储器内的一个中间结果。

[0014] 在本文中介绍的技术能够对存储在大规模存储簇(例如，Hadoop簇)内的大数据集合体执行低延迟查询处理。由于操作人员不需要等待完成当前查询来确定下一个查询，所以这对于交互式分析特别有利。通过扫描一部分相关数据，这个分布式查询系统可以进一步生成近似结果。在处理对整组相关数据的一系列查询之前，系统的操作人员可以接收一系列查询的快速原型，以测试这一系列查询的有效性。

[0015] 尤其在所附权利要求中公开了根据本发明的实施方式，涉及一种系统、一种存储介质以及一种方法，其中，也可以在另一个权利要求目录(例如，方法)中要求在一个权利要求目录(例如，系统)中提及的任何特征。

[0016] 在本发明的一个实施方式中，一种系统包括：

[0017] 网关服务器，被配置为从包含存储在具有多个数据节点的分布式存储簇内的数据的数据库的数据库查询中生成多个局部查询，并且根据多个中间结果构建查询结果；以及

[0018] 多个工作者节点，其中，所述多个工作者节点中的每个工作者节点被配置为通过扫描与相应局部查询相关的并且存储在分布式存储簇的至少一个数据节点上的数据，来处

理多个局部查询的相应局部查询，并且其中，所述多个工作者节点中的每个工作者节点进一步被配置为生成多个中间结果的存储在所述工作者节点的存储器内的一个中间结果。

[0019] 所述多个工作者节点中的每个工作者节点进一步被配置为通过扫描与存储在分布式存储簇的至少一个数据节点上的相应局部查询相关的数据的一部分，来处理多个局部查询中的相应局部查询，并且生成存储在所述工作者节点的存储器内的近似中间结果。

[0020] 所述网关服务器可以进一步被配置为根据至少一个近似中间结果构建近似查询结果。

[0021] 所述网关服务器还可以进一步被配置为根据多个中间结果的一部分构建近似查询结果。

[0022] 所述网关服务器可以甚至进一步被配置为识别离散的工作者节点，进一步将分配给离散的工作者节点的局部查询分成多个从属局部查询，并且将所述多个从属局部查询分配给所述多个工作者节点中的一些工作者节点，其中，所述离散的工作者节点是未向网关服务器报告进度或者在预定的时间段之后向网关服务器报告低于预定值的进度的工作者节点。

[0023] 所述多个工作者节点中的每个工作者节点可以是在分布式存储簇内运行相应数据节点的服务。

[0024] 所述系统可以进一步包括：

[0025] 元数据缓存，被配置为缓存数据库的表格级元数据以及分布式存储簇的文件级元数据。

[0026] 所述元数据缓存可以被配置为保持来自用于所述数据库查询的前一数据库查询的缓存的元数据。

[0027] 所述多个工作者节点中的每个工作者节点可以将心跳消息定期发送给网关服务器，以通过所述工作者节点报告局部查询处理的状态。

[0028] 所述网关服务器可以进一步被配置为从客户端装置接收指令，以返回近似查询结果或终止数据库查询的处理。

[0029] 所述网关服务器还可以进一步被配置为指示工作者节点立即返回近似中间结果，并且根据近似中间结果，将近似查询结果返回至客户端装置。

[0030] 所述数据库查询可以包括近似查询结果的请求。

[0031] 所述查询结果可以伴有存储在数据节点内的已经为所述查询结果扫描的一部分相关数据的指示。

[0032] 所述数据库可以是Hive数据仓库系统，并且所述分布式存储簇是Hadoop簇。

[0033] 在本发明的进一步实施方式中，一种方法使用根据本发明或上述实施方式中的任一个所述的系统。

[0034] 在本发明的进一步实施方式中，一种方法包括：

[0035] 从客户端装置中接收包含存储在具有多个数据节点的分布式存储簇内的数据的数据库的数据库查询；

[0036] 将数据库查询分成多个局部查询；

[0037] 将每个局部查询发送给多个工作者节点中的相应工作者节点，其中，每个工作者节点是在分布式存储簇的数据节点上运行的服务；

[0038] 从工作者节点中检索局部查询的多个中间结果，其中，通过扫描存储在运行相应工作者节点的数据节点内的相关数据，每个中间结果由工作者节点的相应工作者节点处理；以及

[0039] 根据多个中间结果，生成查询结果。

[0040] 所述方法可以进一步包括：

[0041] 将查询结果以及部分指示符返回至客户端装置，其中，所述部分指示符指示存储在数据节点内的已经为所述查询结果扫描的一部分相关数据。

[0042] 所述方法可以进一步包括：

[0043] 指示工作者节点立即返回近似查询结果；

[0044] 并且其中，所述检索步骤包括：

[0045] 从工作者节点中检索局部查询的多个近似中间结果，其中，通过扫描存储在运行相应工作者节点的数据节点内的一部分相关数据，每个近似中间结果由工作者节点的相应工作者节点处理。

[0046] 所述方法可以进一步包括：

[0047] 对于每个局部查询，检索关于哪个数据节点存储与局部查询相关的数据的元数据；

[0048] 并且所述发送步骤包括：

[0049] 根据所述元数据，将每个局部查询发送给多个工作者节点的相应工作者节点。

[0050] 在还可以要求的本发明的进一步实施方式中，一种方法包括：

[0051] 从客户端装置中接收包含存储在具有多个数据节点的分布式存储簇内的数据的数据库的数据库查询；

[0052] 将数据库查询分成多个局部查询；

[0053] 将每个局部查询发送给多个工作者节点中的相应工作者节点，其中，每个工作者节点是在分布式存储簇的数据节点上运行的服务；

[0054] 识别离散的工作者节点，将分配给离散的工作者节点的局部查询分成多个从属局部查询，并且将所述多个从属局部查询分配给所述多个工作者节点中的一些；

[0055] 从工作者节点中检索局部查询的多个中间结果，其中，通过扫描存储在运行相应工作者节点的数据节点内的相关数据，每个中间结果由工作者节点的相应工作者节点处理；以及

[0056] 根据多个中间结果，生成查询结果。

[0057] 所述识别步骤可以包括：

[0058] 通过监控工作者节点定期发送的心跳消息，识别离散的工作者节点，其中，在预定的时间段内没有接收到离散的工作者节点的心跳消息时，或者在接收来自离散的工作这节点的心跳消息并且所述心跳消息包括低于阈值的表示离散的工作者节点的局部查询处理的状态的数量时，识别所述离散的工作者节点。

[0059] 在还可以要求的本发明的进一步实施方式中，一个或多个计算机可读永久性存储介质实现为软件，在执行时，该软件可操作，以在根据本发明或上述实施方式中的任一个所述的系统中运行。

[0060] 通过附图并且通过以下详细描述，在本文中介绍的技术的其他方面显而易见。

附图说明

[0061] 通过结合所附权利要求和附图的以下具体描述(它们均构成本说明书的一部分)的研究,本发明的这些和其他目标、特征以及特性对于本领域的技术人员更加显而易见。在图中:

[0062] 图1示出了可以在其上建立分布式查询引擎的Hadoop簇的一个实例;

[0063] 图2示出了具有管理MapReduce任务的工作跟踪器(JobTracker)的Hadoop簇的一个实例;

[0064] 图3示出了分布式查询引擎、Hadoop分布式文件系统(HDFS)、Hive数据仓库以及存储簇之间的关系;

[0065] 图4示出了一个示例性分布式查询引擎的高级框图;

[0066] 图5示出了用于识别离散的工作者节点并且进一步分割局部查询的示例性处理;

[0067] 图6示出了数据库查询的近似处理的示例性处理;

[0068] 图7是示出可以表示在本文中描述的任何簇节点的计算机节点的架构的一个实例的高级框图。

具体实施方式

[0069] 在本说明书中参考“实施方式”、“一个实施方式”等,表示在本发明的至少一个实施方式中包括所描述的特定特征、功能或特性。这样的短语出现在本说明书中,不必指相同的实施方式,也不必互相排斥。

[0070] 现代社会网络系统每天可以累积大量数据,因此,需要一种用于存储和分析数据的高度可扩展的系统。尤其地,对大量数据的有效的交互分析需要一种处理数据查询的低延迟快速响应的方式。本发明公开了一种分布式查询引擎,通过使基于内存内服务树的计算框架与近似查询处理相结合,来启用该引擎。分布式查询引擎将查询任务分成多个局部任务,并且将局部任务分配给工作者节点,用于进一步内存内处理。分布式查询引擎能够通过根据数据的扫描部分从工作者节点中请求近似中间结果,来在查询处理期间随时生成近似结果。与传统的Hadoop簇的map-reduce方案不同,工作者节点处理局部任务并且在存储器中存储整个中间结果,以减少处理时间并且改善总体延迟。仅仅传输中间结果,而非下面的日期,用于构建结果,大幅减少了传输数据的量和传输时间。

[0071] 在一个实施方式中,分布式查询引擎可以在运行Hadoop分布式文件系统(HDFS)、Hive数据仓库以及Hive-Metastore的Hadoop簇的顶部上建立。分布式查询引擎可以与Hive的数据格式和元数据兼容,并且支持HiveQL语言的子集。使用分布式查询引擎的操作人员可以在由Hive数据仓库管理的数据内有效地发现统计模式。分布式查询引擎可以通过生成近似结果,来进行一系列查询的快速分析和快速原型化。此外,分布式查询引擎可以通过扫描整个相关数据聚合来运行全分析。

[0072] 图1示出了可以在其上建立分布式查询引擎的Hadoop簇的一个实例。在图1中,Hadoop簇100包括元数据节点110A和多个数据节点110B、110C以及110D。这些节点可以通过互连120来彼此通信。例如,互连120可以是局域网(LAN)、广域网(WAN)、城域网(MAN)、全球区域网络(例如,互联网)、光纤通道架构或这样的互连的任何组合。在一些实施方式中,互

连120可以包括网络开关，用于在网络协议(包括TCP/IP)之下在节点之间处理和路由数据。客户端130A和130B可以通过网络140与Hadoop簇100通信，该网络可以是(例如)互联网、LAN或任何其他类型的网络或网络的组合。例如，每个客户端可以是传统的个人电脑(PC)、服务器类计算机、工作站、手持式计算/通信装置等。在一些实施方式中，使用商品等级服务器(commodity-class server)的一个或多个机架(rack)，实现Hadoop簇。

[0073] 通过分布的方式，在Hadoop分布式文件系统(HDFS)内的Hadoop簇100中的节点上存储文件和数据。对于包括客户端130A和130B的簇100的客户端，HDFS提供传统分层文件系统的功能。可以在HDFS内创建、删除或移动文件以及文件的数据块。名称节点服务150在元数据节点110A上运行，以在HDFS内提供元数据服务，包括由外部客户端保持文件系统名称空间并且控制访问。名称节点服务可以在元数据节点内的称为Fslmage的文件160内存储文件系统索引(包括将块映射到文件中)以及文件系统性能。在一些实施方式中，可以存在在次级名称节点服务上运行的次级元数据节点。如果元数据节点失效，那么次级元数据节点用作备份。

[0074] 每个数据节点110负责存储HDFS的文件。存储在HDFS内的文件被分成子集，在本文中称为“块”。在一个实施方式中，块的尺寸是64MB。块通常被复制到多个数据节点。因此，在Hadoop簇100内的HDFS可以但是不必使用传统的RAID架构来获得数据可靠性。文件操作由在元数据节点110A上运行的名称节点服务150控制。在一些实施方式中，将数据节点110B、110C、110D组织到rack内，其中，所有节点通过网络开关连接。在rack内的节点之间的网络速度可以比在不同的rack内的节点之间的网络速度更快。Hadoop簇在分配任务时可以考虑这个事实。数据节点服务170在每个数据节点上运行，用于响应块的读取和写入请求。数据节点服务170还对来自用于创建、删除以及复制块的元数据节点的请求作出响应。

[0075] 在一些实施方式中，数据节点110B、110C、110D将包括块报告的周期性心跳消息发送给元数据节点110A。元数据节点110A使用周期性块报告验证其块映射以及其他文件系统元数据。

[0076] 在客户端130A或130B试图将文件写入Hadoop簇100内时，客户端将文件创建请求发送给元数据节点110A。元数据节点110A通过一个或多个分配的数据节点的身份以及文件的块的目的地位置对客户端作出响应。客户端将文件的数据块发送给分配的数据节点；并且簇可以在一个或多个数据节点上复制数据块。一旦发送了所有块，元数据节点就在其元数据内记录文件创建，包括Fslmage文件。

[0077] Hadoop簇根据称为MapReduce的框架用作平行数据处理引擎。Hadoop簇包括JobTracker，用于实现MapReduce功能。如在图2中所示，JobTracker可以被实施为Hadoop簇200内的专用服务器(JobTracker节点210A)。Hadoop簇200包括通过开关226互连的两个rack 242和244。rack 242包括JobTracker节点210A、元数据节点210B、数据节点210C-210D以及用于互连rack 242内的节点的开关222。rack 244包括数据节点210E-210H和用于互连rack 242内的节点的开关224。在一些其他实施方式中，JobTracker可以被实施为与名称节点服务共享相同的元数据节点的服务。元数据节点210B(也称为名称节点)运行名称节点服务，以跟踪在簇之上保持数据的地方。专用于控制MapReduce工作的JobTracker节点210A从客户端230接收请求以开始MapReduce工作。一旦将MapReduce工作(也称为MapReduce应用程序或MapReduce任务)提交给JobTracker 210A，JobTracker 210A就在HDFS内为该工作识

别输入和输出文件和/或目录。MapReduce任务的输入文件可以包括包含用于MapReduce任务的输入数据的多个输入文件块。JobTracker 210A使用输入文件块(包括块的物理数量以及块的位置)的知识决定将创建多少从属任务。将MapReduce应用程序复制到存在输入文件块的每个处理节点。对于每个分配的处理节点,JobTracker 210A创建至少一个从属任务。在每个分配的处理节点上,任务跟踪器(TaskTracker)服务监控在该节点上从属任务的状态,并且将该状态和中间输出报告至JobTracker。Hadoop簇200根据文件块的知识,分配从属任务。因此,并未将存储移动到处理位置,Hadoop簇将处理任务移动到存储位置。

[0078] 虽然节点210A-210H在图2中被示出为单个单元,但是每个节点可以具有分布式架构。例如,节点可以被设计为多个计算机的组合,这些计算就可以彼此在物理上分开并且可以通过物理互连彼此通信。这样的架构允许方便的缩放,例如,通过部署能够通过互连彼此通信的计算机。

[0079] 在一个实施方式中,在运行Hadoop分布式文件系统(HDFS)和Hive数据仓库的Hadoop簇的顶部上,建立分布式查询引擎。图3示出了分布式查询引擎300、Hive数据仓库、HDFS以及存储簇之间的关系。分布式查询引擎300建立在Hive数据仓库和HDFS的顶部上,Hive数据仓库和HDFS由此依赖于存储簇来操作。Hadoop簇310包括负责存储大量数据的多个节点312。Hadoop簇310进一步包括元数据节点314。Hadoop分布式文件系统(HDFS)320在Hadoop簇310上运行,以分配和管理节点312中的数据。数据节点服务322在节点312上运行,以管理节点312内的本地数据存储。将数据和文件分成存储在Hadoop簇310的节点312内的块。名称节点服务324在元数据节点314上运行,以在Hadoop簇内提供元数据服务,包括由保持文件系统名称空间并且控制外部客户端的访问。

[0080] Hive数据仓库系统330建立在Hadoop簇310和HDFS 320的顶部上。Hive数据仓库系统330用作数据库接口。Hive数据仓库系统330支持以类似于SQL的声明式语言HiveQL表示的数据查询。并非依赖于Hadoop簇的map-reduce方案来处理Hive系统的数据库查询,分布式查询引擎包括多个工作者节点342,用于以平行的方式处理数据库查询。分布式查询引擎300进一步包括网关348。在一个实施方式中,工作者节点342被实施为在Hadoop簇310的节点312上运行的服务。在另一个实施方式中,工作者节点342被实施为与Hadoop簇310的节点312互连的专用服务器。

[0081] 在一些实施方式中,工作者节点342负责将以HiveQL表示的局部任务编译成HDFS 320的数据节点服务322可以执行的指令。

[0082] 分布式查询引擎从客户端接收查询任务,并且将查询任务分成多个局部任务。图4示出了示例性分布式查询引擎400的高级框图。分布式查询引擎400的操作人员可以通过客户端480的输入界面486提供查询任务。在一个实施方式中,输入接口486包括命令行界面482和图形界面484。使用命令行界面482,操作人员可以提供作为直接以数据库查询语言(例如,SQL或HiveQL)表示的程序的查询任务。通过使用图形界面484,操作人员可以通过使用图形界面元素484来提供查询任务。在一个实施方式中,图形界面484被实施为输入网页。操作人员可以通过与在输入网页上的元素交互、选择选项以及输入输入数据来提供查询任务。图形界面484可以将操作人员的选择和输入转化成以数据库查询语言表示的相应程序。输入接口486将从命令行界面482或图形界面484中接收的程序作为查询任务传输给分布式查询引擎400的网关410。

[0083] 网关410从客户端480中接收查询任务，并且解析查询任务。网关410根据查询任务，向Hive元数据存储440发送询问。Hive元数据存储440将表格元数据和HDFS文件识别返回网关410，用于需要运行查询任务的数据。然后，网关410根据HDFS文件识别，从HDFS名称节点460检索相应的HDFS块的位置。

[0084] 在一个实施方式中，网关410根据对应的HDFS块将查询任务分成多个局部查询。网关410分配每个单独的局部查询，以在对应的HDFS块内在一个HDFS块上执行。在其他实施方式中，网关410可以其他方式将查询任务分成局部任务，如本领域的技术人员可以预计的。

[0085] 网关410将每个局部查询发送给工作者412，用于本地处理。在一个实施方式中，工作者412覆盖在存储Hive表格数据的Hadoop簇上。每个工作者412作为服务在Hadoop簇节点432上运行。生成局部查询，以便每个工作者412负责局部查询，以在运行这个特定的工作者412的节点432上处理数据存储。工作者412与DataNode服务422直接接触，该服务在与工作者412相同的簇节点432上运行。通过请求在单个簇节点432内的数据，工作者412能够实现局部查询的低延迟数据读取。

[0086] 簇节点432可以使用远程程序调用(RPC)框架来促进服务的实施。例如，在一个实施方式中，簇节点432使用RPC框架(例如，Apache Thrift框架)来限定和创建工作者服务412，作为高度可扩展的以及高性能的服务器服务。

[0087] 在一个实施方式中，工作者节点342被实施为在Hadoop簇310的节点312上运行的服务。在另一个实施方式中，工作者节点342被实施为与Hadoop簇310的节点312互连的专用服务器。

[0088] 工作者412将状态更新(称为“心跳”)定期返回给网关410，指示局部查询处理的过程。在一个实施方式中，如果存在停止返回心跳或者不显示进度的分配的工作者，那么网关410确定该工作者失效并且向另一个工作者重新安排局部查询。每个工作者412扫描与存储在一个或多个簇节点432上的相应局部查询相关的数据，并且生成局部查询的中间结果。在一个实施方式中，工作者412在运行工作者412的簇节点的存储器内整体处理局部查询。工作者412在其存储器内存储中间结果。在局部查询的处理结束时，工作者412将中间结果发送给网关410。在一些实施方式中，工作者412通过RPC调用(例如，Apache Thrift调用)发送中间结果。

[0089] 网关410从工作者412中接收所有中间结果，并且将中间结果组合成查询结果，作为对查询任务的回应。然后，网关410将查询结果返回给客户端480。在一个实施方式中，客户端480可选地在显示元件上显示查询结果。

[0090] 元数据缓存(meta cache)414在网关410上运行，以高速缓存Hive表格级和HDFS文件级的元数据，以减少查询延迟。在一些实施方式中，元数据缓存414可以被实施为与网关410互连的独立式服务器。元数据缓存414可以保持先前查询的缓存数据。例如，当操作人员对Hive表格的数据进行交互式分析时，操作人员在相同的Hive表格上运行多个连续的查询。通过保存来自先前查询的缓存数据，元数据缓存414可以重新使用缓存的元数据，而非从Hive元数据存储440和HDFS名称节点460重复提取元数据。

[0091] 元数据缓存414具有高的缓存命中率，这是因为在典型的Hive表格中的数据写入一次，并且读取多次，而不进一步改变。在一个实施方式中，元数据缓存414可以检索Hive系统的审计记录的实时馈送，以使在缓存数据内的条目无效，用于在Hive查询或其他操作可

以改变的Hive系统内划分。在另一个实施方式中，元数据缓存414自动清除缓存数据中在预定的时间段(例如，1个小时)内未被查询的条目。这样，元数据缓存414防止了内存使用的任意增长，并且尽可能减少缓存错误。

[0092] 工作者412在其上运行的每个簇节点432的工作量可以不同。簇节点432和工作者服务412还可以由于各种原因而失效。虽然网关410可以在合理的时间段内从多数工作者412接收中间结果，但是由于节点或服务故障或延迟，所以存在不能传输中间结果的工作者412。这些工作者在从局部查询分布以来的预定时间段之后报告低于预定百分比的报告进度，或者仅仅不对网关410做出进度回应。这些工作者被认定为离散工作者(straggling worker)。一旦网关410识别离散工作者，网关410就将消息发送给离散工作者，以取消局部查询的分配。对于离散工作者不能传输中间结果的每个未完成的局部查询，网关410进一步将局部查询分成多个从属局部查询，并且将从属局部查询分配给工作者412中的一些。在一个实施方式中，网关410根据工作者412的当前工作量确定从属局部查询的分配。在其他实施方式中，网关410可以通过其他方式确定分配，如在本领域的技术人员所预计的。这个额外的并行化处理加速了未完成的局部查询的重试，从而减少由离散工作者造成的查询延迟。

[0093] 图5示出了用于识别离散的工作者节点并且进一步分割局部查询的示例性处理。在步骤502中，分布式查询引擎的网关从客户端装置中接收包含存储在具有多个数据节点的分布式存储簇内的数据的数据库的数据库查询。在步骤504中，网关将数据库查询分成多个局部查询。然后，在步骤506中，网关将每个局部查询发送给多个工作者节点中的相应工作者节点。每个工作者节点可以是在分布式存储簇的数据节点上运行的服务。

[0094] 在步骤508中，网关服务器识别离散的工作者节点。网关进一步将分配给离散的工作者节点的局部查询分成多个从属局部查询，并且将多个从属局部查询分配给多个工作者节点中的一些工作者节点。在一个实施方式中，网关通过监控工作者节点定期发送的心跳消息，识别离散的工作者节点。在预定的时间段内没有接收到离散的工作者节点的心跳消息时，识别离散的工作者节点。在另一个实施方式中，在接收离散的工作者节点的心跳消息，其中，所述心跳消息包括低于阈值的表示离散的工作者节点的局部查询处理的状态的数量时，识别所述离散的工作者节点。

[0095] 在步骤510中，网关从工作者节点中检索局部查询的多个中间结果。通过扫描存储在运行相应工作者节点的数据节点内的相关数据，每个中间结果由工作者节点的相应工作者节点处理。在步骤512中，网关根据多个中间结果，生成查询结果。

[0096] 在一个实施方式中，分布式查询引擎可以与Hive系统的数据格式和元数据兼容，并且可以支持HiveQL语言的子集或整个集合。而且，HiveQL是与SQL相似的声明式语言。HiveQL不需要严格地遵循标准SQL，并且提供最初在SQL中未规定的扩展部分。例如，分布式查询引擎可以支持滤波器、集合体、top-k、百分位数、在FROM从句内的子查询、UNION ALL以及用户定义的功能。

[0097] 在一个实施方式中，分布式查询引擎支持TABLESAMPLE从句可以用于明确地限制扫描的输入数据的量。在另一个实施方式中，通过在相同的查询内给多次使用的复杂表达式声明变量，分布式查询引擎支持WITH从句允许操作人员写入更多可读的查询。WITH从句还可以给操作人员提供一种向优化程序规定提示的方法，以便在运行时间仅仅评估一次公

共子表达式。

[0098] 在一些实施方式中，分布式查询引擎可以在完成整个查询处理之前提供近似查询结果。分布式查询引擎可以在自初始查询输入预定时间段之后或者在查询的处理满足预定的条件(例如，一定数量的工作者失效)时自动提供近似查询结果。分布式查询引擎还可以响应于操作人员指令提供近似查询结果。例如，等待查询结果的操作人员可以输入Ctrl-C，以指示分布式查询引擎停止查询处理。在接收指令时，分布式查询引擎停止查询处理并且返回近似查询结果。在一个实施方式中，分布式查询引擎进一步提供百分比指示符，用于指示近似查询结果的扫描的数据的百分比。在一个实施方式中，分布式查询引擎返回近似查询结果，并且继续查询处理，用于精确的查询结果(即，具有100％的百分比指示符)。

[0099] 尤其对于探索数据(而非写入或改变数据)的查询，近似查询结果对于操作人员的分析可以是足够的。节点故障、错误的输入数据、或者甚至用户终止查询等运行时间错误可以被视为所有输入数据未被扫描的情况。当存在故障时，迄今为止，分布式查询引擎可以根据局部查询的到目前为止的处理立即返回近似查询结果，而非仅仅返回错误消息。在一个实施方式中，分布式查询引擎返回近似查询结果以及百分比指示符。

[0100] 在另一个实施方式中，操作人员还可以在其查询任务中指定需要精确的查询结果。在这些情况下，如果查询处理失败，那么可以返回运行时间错误。

[0101] 分布式查询引擎使用一遍聚合算法(one-pass algorithms for aggregations)，并且在存储器中存储所有中间结果。中间结果和最终查询结果的尺寸可以相对小。分布式查询引擎的返回近似查询结果的能力可以进一步减小该尺寸。例如，如果查询是通过规定的列使数据记录排序(例如，ORDER BY从句)，那么通过允许工作者仅仅扫描一部分相关数据记录，分布式查询引擎可以生成近似回答。同样，分布式查询引擎还可以生成查询类型的近似回答，例如，计算不同的条目并且计算百分位数。

[0102] 例如，在一个实施方式中，分布式查询引擎的操作人员可以输入指示从特定一组数据记录中对country列的不同值的数量进行技术的查询任务。分布式查询引擎接收查询，将查询分成局部查询，并且分配工作者，以处理这些局部查询。在从分布式查询引擎开始该任务的时间开始20秒之后，操作人员通过在客户端装置的键盘上按压CTRL-C来终止任务。在接收终止指令时，分布式查询引擎立即指示分配的工作者返回近似中间结果，并且从而通过组合这些近似中间结果来返回近似结果。近似结果被返回操作人员的客户端装置。客户端装置可以进一步接收关于总处理时间、所使用的工作者的数量、扫描的数据记录的行数、扫描的数据量、精确结果的扫描的数据量、扫描的数据的百分比和/或故障的数量的信息。

[0103] 在另一个实施方式中，分布式查询引擎可以在故障的数量超过预定的阈值之后，自动返回近似查询结果。在又一个实施方式中，操作人员在查询任务中指定需要精确结果。分布式查询引擎将保持处理，直到所有相关的数据记录被分配的工作者扫描。精确的查询结果可以被返回至操作人员的用户装置。客户端装置可以进一步接收关于总处理时间、所使用的工作者的数量、扫描的数据记录的行数、扫描的数据量、扫描的数据的百分比(即，100％)和/或故障的数量的信息。

[0104] 图6示出了数据库查询的近似处理的示例性处理。在步骤602中，分布式查询引擎的网关从客户端装置接收包含存储在具有多个数据节点的分布式存储簇内的数据的数据

库的数据库查询。在步骤604中，网关将数据库查询分成多个局部查询。然后，在步骤606中，网关将每个局部查询发送给多个工作者节点中的相应工作者节点。每个工作者节点可以是在分布式存储簇的数据节点上运行的服务。在步骤608中，在发送局部查询之后，网关可以指示工作者节点立即返回近似查询结果。在一个实施方式中，立即返回近似查询结果表示在非常短的时间段(例如，1秒)内返回结果。该指令可以由各种事件触发。例如，网关可以从客户端装置接收终止数据库查询的处理的指令；或者在预定的时间段之后，精确的查询结果不可用时，网关可以决定自动返回近似查询结果。因此，近似结果可以由客户端装置手动请求，或者由分布式查询引擎自动触发，无需用户干预。

[0105] 在步骤610中，在近似中间结果的指令之后，网关从工作者节点检索局部查询的多个近似中间结果。通过扫描存储在运行相应工作者节点的数据节点内的一部分相关数据，每个近似中间结果由工作者节点的相应工作者节点处理。在接收近似中间结果时，在步骤612中，网关根据多个近似中间结果，生成近似查询结果。然后，在步骤614中，分布式查询引擎的网关返回近似查询结果。在一个实施方式中，将近似查询结果以及百分比指示符返回给客户端装置。百分比指示符提示存储在数据节点内的已经为查询结果扫描的相关数据的百分比。

[0106] 除了以上显示的优点以外，如下面所讨论的，在本文中提出的技术还显示了额外的优点。

[0107] 与仅仅使用Hive系统的查询延迟相比，分布式查询引擎大幅减少了对存储在数据存储簇(例如，Hadoop簇)内的数据的查询的延迟。分布式查询引擎的操作人员或用户可以少的等待时间进行ad hoc查询。可以在不同的情况下使用分布式查询引擎。例如，在没有分布式查询引擎的情况下，操作人员或分析员需要将数据从Hive中明确地载入数据库(例如，MySQL或Oracle)内，然后，从数据库中提取数据，以驱动基于网络的数据分析报告。使用分布式查询引擎，操作人员可以从Hive系统中直接提取数据，以生成基于网络的数据分析报告。

[0108] 在一个实施方式中，在操作人员使用(例如)在图4中示出的图形界面484制作查询任务时，分布式查询引擎可以提取数据样本，以在图形界面484上向操作人员显示数据的预览。在另一个实施方式中，分布式查询引擎可以在存储器内索引并且锁定(pin)常用的数据集，以进一步提高查询延迟。

[0109] 在本文中提出的技术提供了一种低延迟的分布式查询引擎，其可以建立在数据存储簇的顶部上。分布式查询引擎可以与Hive系统的现有数据和元数据兼容。分布式查询引擎可以被用于驱动数据分析报告，无需将数据载入其他数据库(例如，MySQL或Oracle)内以用于ad-hoc分析的管线。

[0110] 图7是示出可以表示在本文中描述的任何簇节点的计算机节点的架构的一个实例的高级框图。节点700包括与互连730耦接的一个或多个处理器710和存储器720。在图7中示出的互连730是表示由合适的桥接器、适配器或控制器连接的任何一个或多个单独的物理总线、点对点连接或这两者的抽象化表示。因此，互连730可以包括(例如)系统总线、外部设备互连(PCI)总线或PCI快线、超传输或工业标准架构(ISA)总线、小型计算机系统接口(SCSI)总线、通用串行总线(USB)、IIC(I2C)总线或电气与电子工程师协会(IEEE)标准1394总线，也称为“火线”。

[0111] 处理器710是存储控制器700的中央处理单元(CPU),从而控制节点700的总体操作。在某些实施方式中,处理器710通过执行存储在存储器720内的软件或固件来完成这个。处理器710可以是或者可以包括一个或多个可编程通用或专用微处理器、数字信号处理器(DSP)、可编程控制器、专用集成电路(ASIC)、可编程逻辑装置(PLD)、可信任平台模块(TPM)等或这样的装置的组合。

[0112] 存储器720是或者包括节点700的主存储器。存储器720表示任何形式的随机存取存储器(RAM)、只读存储器(ROM)、闪速存储器等或这样的装置的组合。在使用时,存储器720可以包含代码770,该代码包含根据在本文中公开的技术的指令。

[0113] 同样通过互连730连接至处理器710的是网络适配器740和存储适配器750。网络适配器740给节点700提供与远程装置通过网络通信的能力,并且例如可以是以太网适配器或光纤通道适配器。网络适配器740还可以给节点700提供与在簇内的其他节点通信的能力。在一些实施方式中,节点可以使用不止一个网络适配器来单独处理在簇内以及簇外的通信。存储适配器750允许节点700访问永久存储器,并且例如可以是光纤通道适配器或SCSI适配器。

[0114] 存储在存储器720内的代码770可以实施为软件和/或固件,以对处理器710进行编程,以执行上述动作。在某些实施方式中,可以通过节点700(即,通过网络适配器740)从远程系统中下载这种软件或固件,为将这种软件或固件首先提供给节点700。

[0115] 在本文中介绍的技术例如可以由通过软件和/或固件或者完全在专用硬接线电路内或者在这种形式的组合内编程的可编程电路(例如,一个或多个微处理器)实现。例如,专用硬接线电路可以具有一个或多个专用集成电路(ASIC)、可编程逻辑装置(PLD)、现场可编程门阵列(FPGA)等的形式。

[0116] 用于实现在本文中介绍的技术的软件或固件可以存储在机器可读存储介质上,并且可以由一个或多个通用或专用可编程微处理器执行。在本文中使用的术语“机器可读存储介质”包括可以通过机器(例如,机器可以是计算机、网络装置、蜂窝电话、个人数字助理(PDA)、制造工具、具有一个或多个处理器的任何装置等)可访问的形式存储信息的任何机制。例如,机器可访问存储介质包括可记录的/不可记录的介质(例如,只读存储器(ROM);随机存取存储器(RAM);磁盘存储介质;光学存储介质;闪速存储器装置等)等。

[0117] 例如,在本文中使用的术语“逻辑”可以包括通过特定的软件和/或固件编程的可编程电路、专用硬接线电路或其组合。

[0118] 除了上述实例以外,在不背离本发明的情况下,本发明还可以做出各种其他修改和变更。因此,以上公开内容不被视为具有限制性,并且所附权利要求被解释为包括本发明的真实精神和整个范围。

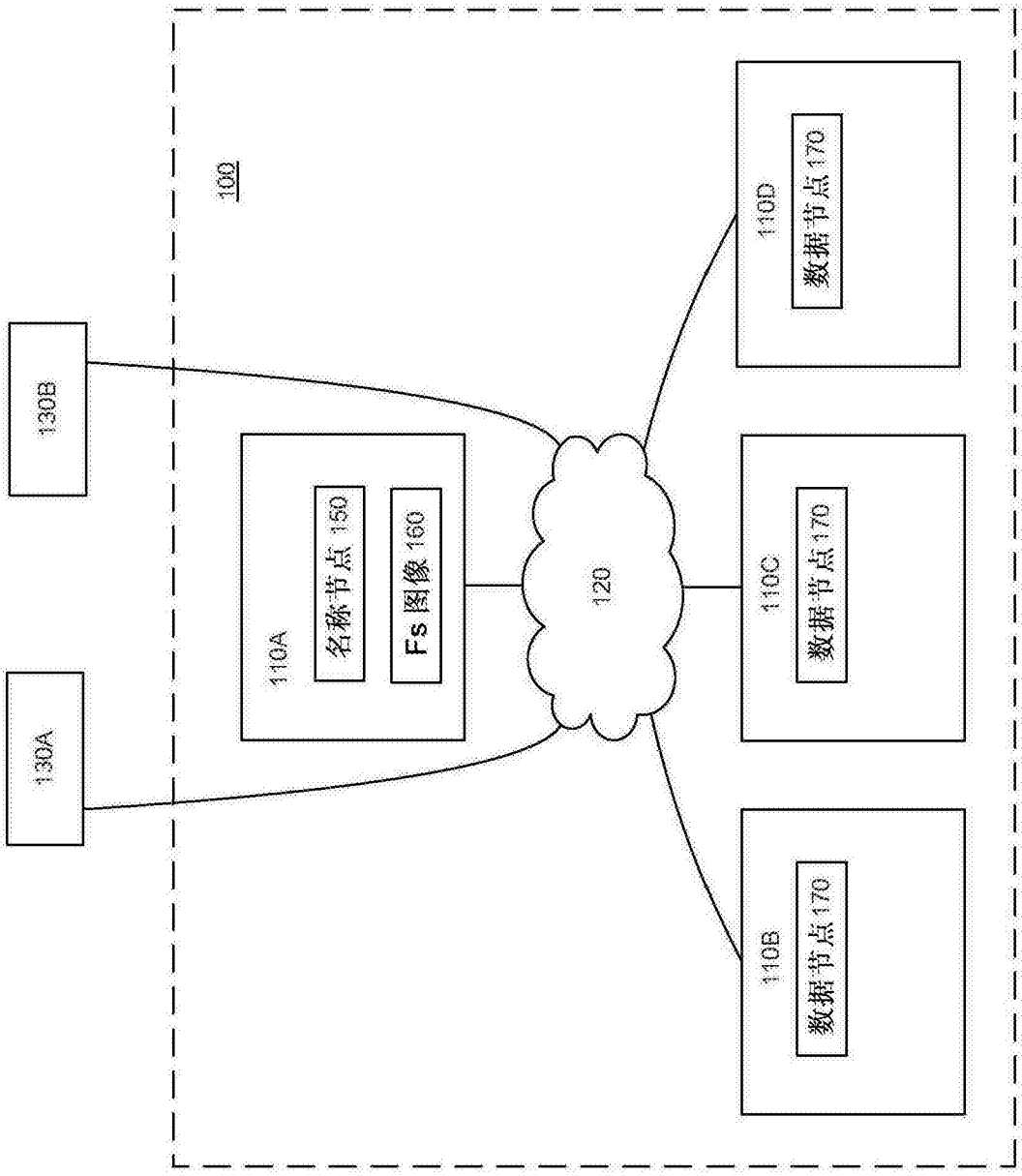

100
130B
130A
110A
名称节点 150
Fs 图像 160
120
110D
数据节点 170
110C
数据节点 170
110B
数据节点 170

图1

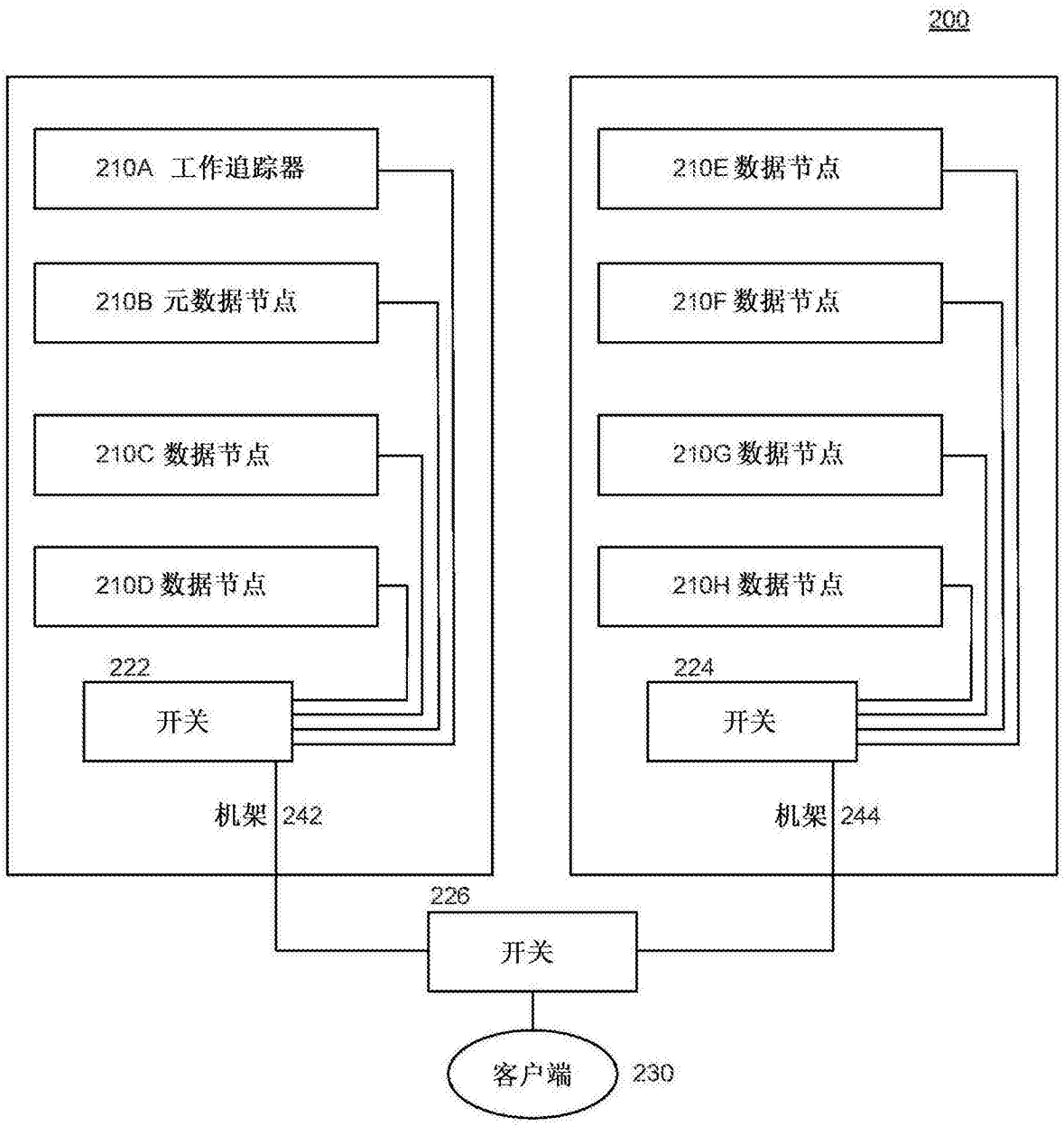

200
210A 工作追踪器
210B 元数据节点
210C 数据节点
210D 数据节点
222
开关
机架 242
210E 数据节点
210F 数据节点
210G 数据节点
210H 数据节点
224
开关
机架 244
226
开关
客户端
230

图2

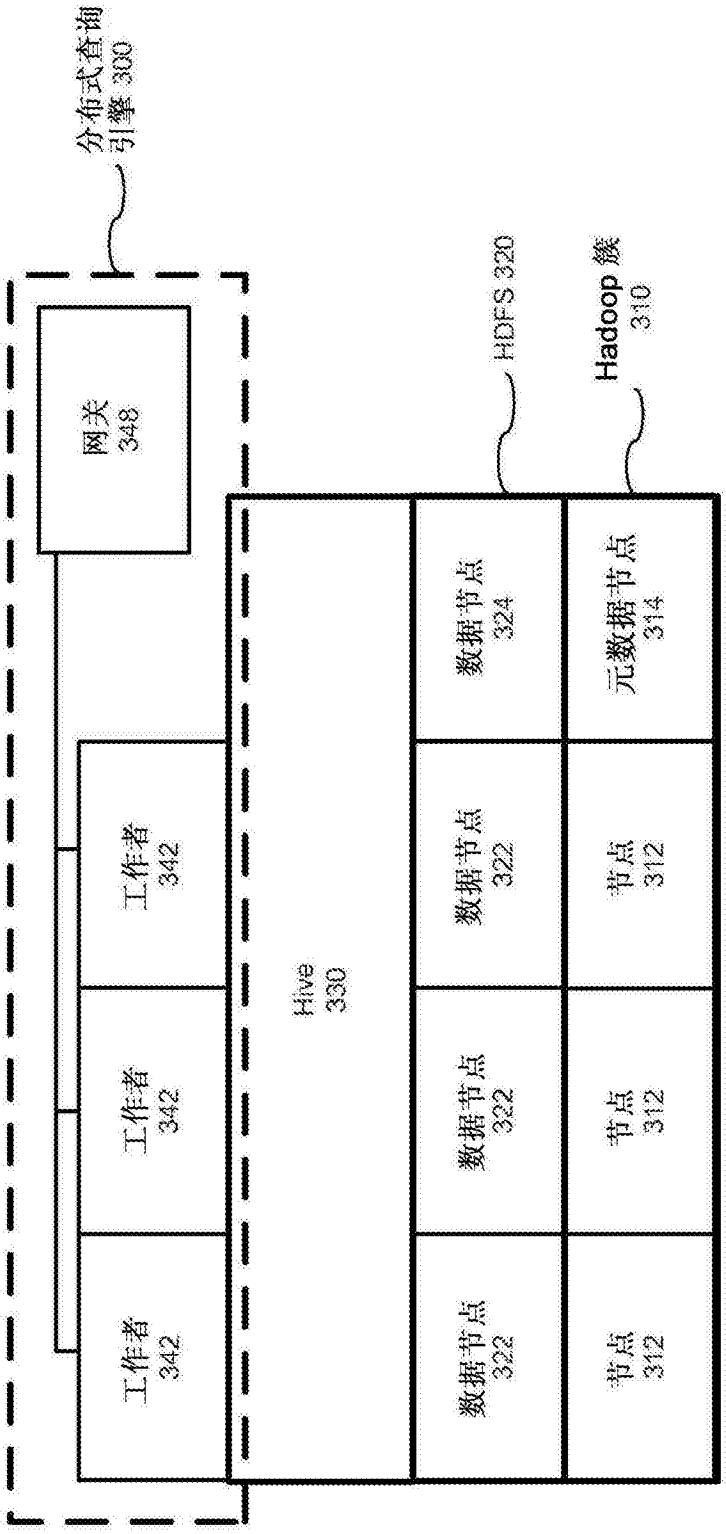

分布式查询
引擎 300
网关
348
工作者
342
工作者
342
工作者
342
Hive
330
HDFS 320
Hadoop 簇
310
数据节点
324
数据节点
322
数据节点
322
数据节点
322
元数据节点
314
节点
312
节点
312
节点
312

图3

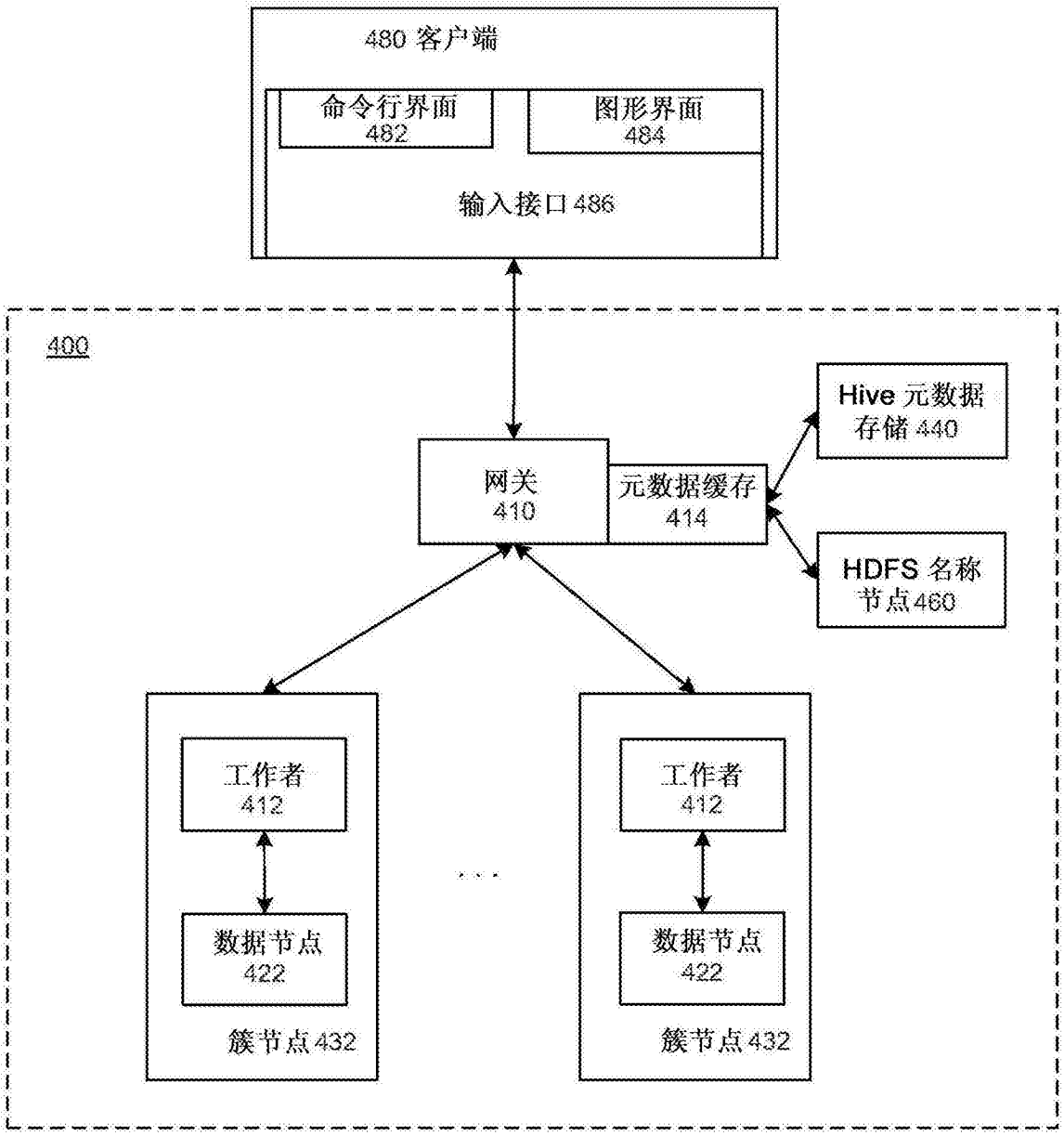

480 客户端
命令行界面
482
图形界面
484
输入接口 486
400
网关
410
元数据缓存
414
Hive 元数据
存储 440
HDFS 名称
节点 460
工作者
412
数据节点
422
簇节点 432
工作者
412
数据节点
422
簇节点 432

图4

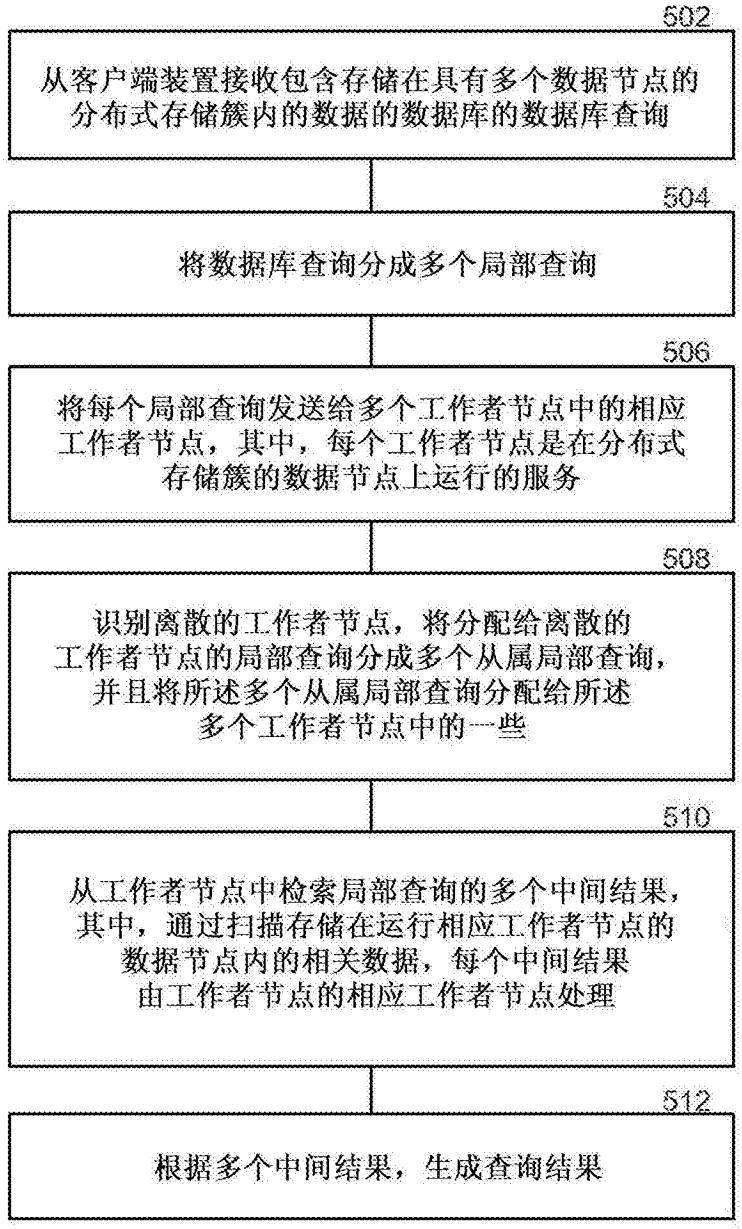

502
从客户端装置接收包含存储在具有多个数据节点的分布式存储簇内的数据的数据库的数据库查询
504
将数据库查询分成多个局部查询
506
将每个局部查询发送给多个工作者节点中的相应工作者节点，其中，每个工作者节点是在分布式存储簇的数据节点上运行的服务
508
识别离散的工作者节点，将分配给离散的工作者节点的局部查询分成多个从属局部查询，并且将所述多个从属局部查询分配给所述多个工作者节点中的一些
510
从工作者节点中检索局部查询的多个中间结果，其中，通过扫描存储在运行相应工作者节点的数据节点内的相关数据，每个中间结果由工作者节点的相应工作者节点处理
512
根据多个中间结果，生成查询结果

图5

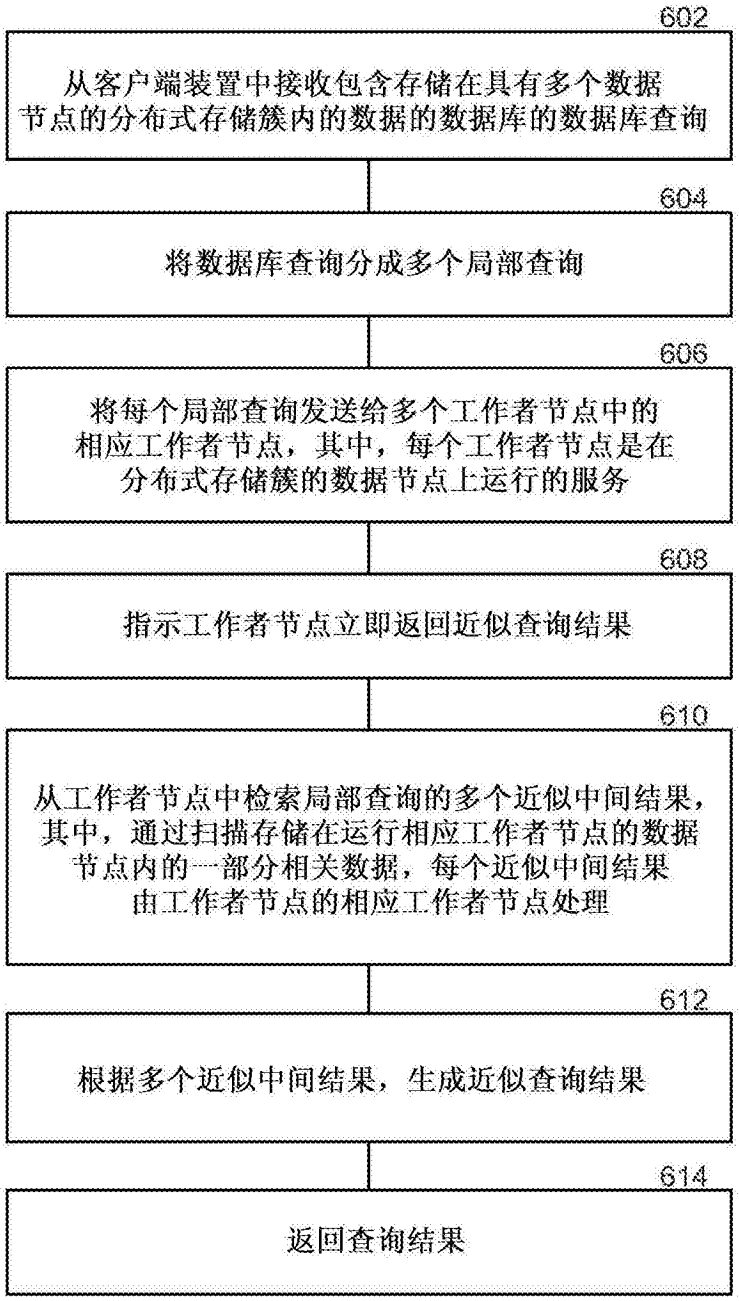

602
从客户端装置中接收包含存储在具有多个数据节点的分布式存储簇内的数据的数据库的数据库查询
604
将数据库查询分成多个局部查询
606
将每个局部查询发送给多个工作者节点中的相应工作者节点，其中，每个工作者节点是在分布式存储簇的数据节点上运行的服务
608
指示工作者节点立即返回近似查询结果
610
从工作者节点中检索局部查询的多个近似中间结果，其中，通过扫描存储在运行相应工作者节点的数据节点内的一部分相关数据，每个近似中间结果由工作者节点的相应工作者节点处理
612
根据多个近似中间结果，生成近似查询结果
614
返回查询结果

图6

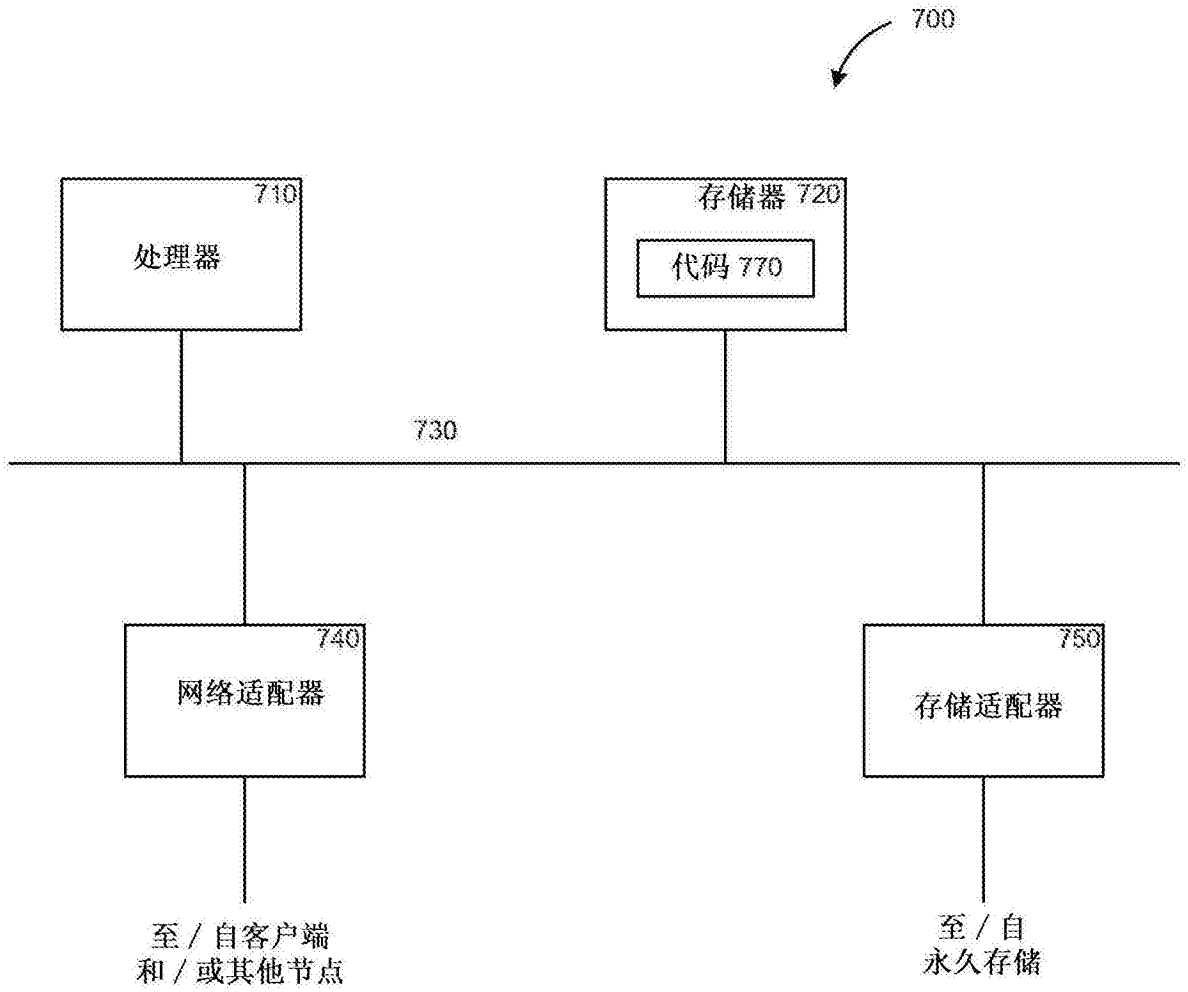

700
710
处理器
存储器 720
代码 770
730
740
网络适配器
750
存储适配器
至 / 自客户端
和 / 或其他节点
至 / 自
永久存储

图7