



(12) 发明专利

(10) 授权公告号 CN 109564564 B

(45) 授权公告日 2024. 04. 05

(21) 申请号 201780034461.2

(22) 申请日 2017.06.01

(65) 同一申请的已公布的文献号
申请公布号 CN 109564564 A

(43) 申请公布日 2019.04.02

(30) 优先权数据
15/171,859 2016.06.02 US

(85) PCT国际申请进入国家阶段日
2018.12.03

(86) PCT国际申请的申请数据
PCT/US2017/035531 2017.06.01

(87) PCT国际申请的公布数据
W02017/210477 EN 2017.12.07

(73) 专利权人 斯诺弗雷克公司
地址 美国加利福尼亚州

(72) 发明人 A·莫蒂维拉 本诺特·戴奇维勒

(74) 专利代理机构 隆天知识产权代理有限公司
72003

专利代理师 石海霞 金鹏

(51) Int.Cl.
G06F 16/13 (2019.01)
G06F 16/22 (2019.01)
G06F 16/28 (2019.01)

(56) 对比文件
US 2015370641 A1, 2015.12.24
US 6209128 B1, 2001.03.27
US 2008256311 A1, 2008.10.16
CN 103221924 A, 2013.07.24
US 2012130949 A1, 2012.05.24
CN 1758219 A, 2006.04.12
熊金波等. 云环境下的数据多副本安全共享
与关联删除方案.《通信学报》.2015,全文.

审查员 高悦

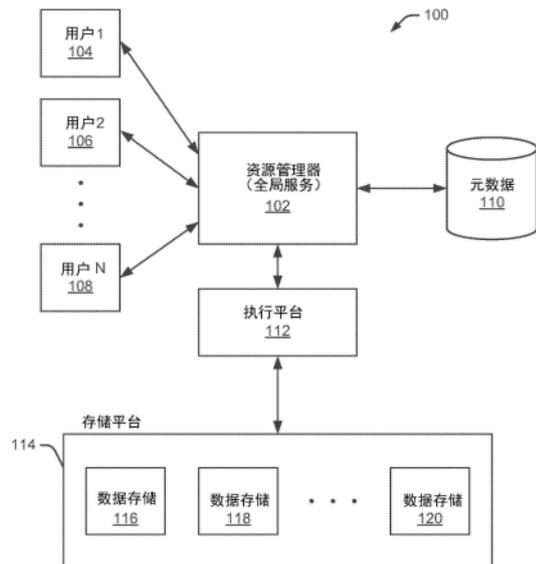
权利要求书2页 说明书13页 附图11页

(54) 发明名称

克隆目录对象

(57) 摘要

本发明描述了用于克隆目录对象的示例系统和方法。在一个实现中,方法识别与一数据集相关联的原始目录对象,并创建原始目录对象的副本而不复制数据本身。该方法允许使用副本目录对象访问数据集,并支持独立于副本目录对象修改与原始目录对象关联的数据。完成对与原始目录对象关联的数据的修改后,可以删除副本目录对象。



1. 一种用于克隆数据库内对象的系统,所述系统包括:
 - 用于识别与所述数据库内的数据集相关联的原始对象的装置,其中每个原始对象表示数据存储系统中的数据的逻辑分组;
 - 用于通过复制与所述数据集相关联的原始对象而不复制包括在所述数据库内的所述数据集中的数据来创建副本对象的装置;
 - 用于通过下述操作修改与所述原始对象相关联的数据的装置:
 - 使用所述原始对象的对应副本对象访问数据存储系统中的数据集;
 - 独立于所述对应副本对象来修改与所述原始对象关联的数据,使得当写入文件时,就无法更新或修改;以及
 - 响应于需要更改文件,删除所述文件并将所述文件替换为包含更改的不同文件;以及
 - 用于在完成对与所述原始对象相关联的数据的修改后删除所述对应副本对象的装置。
2. 根据权利要求1所述的系统,其中用于创建所述副本对象的装置将与所述数据集相关联的元数据复制到所述副本对象。
3. 根据权利要求2所述的系统,其中元数据包括存储在所述数据集中的数据的清单。
4. 根据权利要求2所述的系统,其中所述系统使用所述元数据来确定是否有必要检索包括在所述数据集中的数据来处理查询。
5. 根据权利要求1所述的系统,其中所述原始对象包括表、数据库、模式、帐户、约束、文件格式、函数、角色、序列、阶段、列、用户、卷或目录对象中的一个。
6. 一种用于克隆数据库内的对象的系统,所述系统包括:
 - 存储平台,被配置为存储与数据集相关联的原始对象;以及
 - 资源管理器,被配置为:
 - 识别与所述数据集相关联的原始对象,其中每个原始对象表示数据存储系统中的数据的逻辑分组;
 - 通过复制与所述数据集相关联的原始对象而不复制包括在所述数据库内的所述数据集中的数据来创建副本对象;
 - 通过下述操作修改与所述原始对象相关联的数据:
 - 使用所述原始对象的对应副本对象访问数据存储系统中的数据集;
 - 独立于所述对应副本对象来修改与所述原始对象关联的数据,使得当写入文件时,就无法更新或修改;以及
 - 响应于需要更改文件,删除所述文件并将所述文件替换为包含更改的不同文件;以及
 - 在完成对与所述原始对象相关联的数据的修改后删除所述对应副本对象。
7. 根据权利要求6所述的系统,其中所述资源管理器还被配置为将与所述数据集相关联的元数据复制到所述副本对象。
8. 根据权利要求7所述的系统,其中所述元数据包括存储在所述数据集中的数据的清单。
9. 根据权利要求7所述的系统,其中所述资源管理器使用所述元数据来确定是否有必要检索包括在所述数据集中的数据来处理查询。
10. 根据权利要求6所述的系统,其中所述原始对象包括表、数据库、模式、帐户、约束、文件格式、函数、角色、序列、阶段、列、用户、卷或目录对象中的一个。

11. 一种用于克隆数据库内的对象的方法,包括:
 - 识别与数据集相关联的原始对象,每个原始对象表示数据存储系统中的数据的逻辑分组;
 - 通过复制与所述数据集相关联的原始对象而不复制包括在所述数据库内的所述数据集中的数据来创建副本对象;
 - 通过下述操作修改与所述原始对象相关联的数据:
 - 使用所述原始对象的对应副本对象访问数据存储系统中的数据集;
 - 独立于所述对应副本对象来修改与所述原始对象关联的数据,使得当写入文件时,就无法更新或修改;以及
 - 响应于需要更改文件,删除所述文件并将所述文件替换为包含更改的不同文件;以及在完成对与所述原始对象相关联的数据的修改后删除所述对应副本对象。
12. 根据权利要求11所述的方法,其中创建所述副本对象还包括将与所述数据集相关联的元数据复制到所述副本对象。
13. 根据权利要求12所述的方法,其中创建所述副本对象还包括复制存储在所述数据集中的数据的清单。
14. 根据权利要求12所述的方法,其中所述方法还包括:使用所述元数据来确定是否有必要检索包括在所述数据集中的数据来处理查询。
15. 根据权利要求11所述的方法,其中所述原始对象包括表、数据库、模式、帐户、约束、文件格式、函数、角色、序列、阶段、列、用户、卷或目录对象中的一个。

克隆目录对象

[0001] 相关申请的交叉引用

[0002] 本申请是2014年10月20日提交的发明名称为“Resource management systems and methods(资源管理系统和方法)”的美国专利申请序列号14/518,884的部分继续申请,其公开内容整体通过引用结合于此。该申请要求2014年2月19日提交的发明名称为“Apparatus and method for enterprise data warehouse data processing on cloud infrastructure(用于云基础设施上的企业数据仓库数据处理的装置和方法)”的美国临时申请序列号61/941,986的权益,其公开内容整体也通过引用结合于此。

技术领域

[0003] 本公开涉及支持数据存储和检索的系统和方法。

背景技术

[0004] 现今有许多现有的数据存储和检索系统可用。例如,在共享磁盘系统中,所有数据都存储在可从数据集群中的所有处理节点访问的共享存储设备上。在这种类型的系统中,所有数据更改都被写入共享存储设备,以确保数据集群中的所有处理节点都访问一致的数据版本。随着共享磁盘系统中处理节点的数量增加,共享存储设备(以及处理节点和共享存储设备之间的通信链路)成为减慢数据读取和数据写入操作的瓶颈。随着更多处理节点的添加,这一瓶颈进一步恶化。因此,由于这个瓶颈问题,现有的共享磁盘系统具有有限的可扩展性。

[0005] 另一现有数据存储和检索系统被称为“无共享架构”。在该架构中,数据分布在多个处理节点上,使得每个节点存储整个数据库中数据的子集。当添加或删除新的处理节点时,无共享架构必须在多个处理节点间重新排列数据。这种数据重新排列可能是耗时的并且对数据重新排列期间执行的数据读取和写入操作具有破坏性。并且,数据与特定节点的亲和性可以在数据集群上为流行数据创建“热点”。此外,由于每个处理节点还执行存储功能,因此该架构需要至少一个处理节点来存储数据。因此,如果去除所有处理节点,则无共享架构将无法存储数据。此外,由于数据分布在许多不同处理节点上,所以无共享架构中的数据管理很复杂。

[0006] 本文描述的系统和方法提供了一种改进的数据存储和数据检索方法,其减轻了现有系统的上述限制。

附图说明

[0007] 参考以下附图描述本公开的非限制性和非穷举性实施例,其中除非另有说明,否则相同的附图标记在各个附图中指代相同的部件。

[0008] 图1是描绘本文描述的系统和方法的示例实施例的框图。

[0009] 图2是描绘资源管理器的实施例的框图。

[0010] 图3是描绘执行平台的实施例的框图。

- [0011] 图4是描绘具有多个用户通过多个虚拟仓库访问多个数据库的示例操作环境的框图。
- [0012] 图5是描绘多个用户通过负载均衡器和虚拟仓库组中包含的多个虚拟仓库访问多个数据库的另一示例操作环境的框图。
- [0013] 图6是描绘具有多个分布式虚拟仓库和虚拟仓库组的另一示例操作环境的框图。
- [0014] 图7是描绘用于管理数据存储和检索操作的方法的实施例的流程图。
- [0015] 图8是描绘用于克隆目录对象的方法的实施例的流程图。
- [0016] 图9是描绘用于创建临时副本目录对象的方法的实施例的流程图。
- [0017] 图10A-图10D描绘了访问多个文件的原始表对象和克隆表对象的实施例。
- [0018] 图11是描绘示例计算设备的框图。

具体实施方式

[0019] 本文描述的系统和方法提供了用于存储和检索数据的新平台,而没有现有系统面临的问题。例如,这个新平台支持添加新节点,而无需根据无共享架构的要求重新安排数据文件。此外,可以将节点添加到平台,而不会产生共享磁盘系统中常见的瓶颈。这个新平台始终可用于数据读取和数据写入操作,即使某些节点脱机进行维护或出现故障也是如此。所描述的平台将数据存储资源与计算资源分离,从而可以存储数据而无需使用专用计算资源。这是对无共享架构的改进,如果去除所有计算资源,则无共享架构无法存储数据。因此,即使计算资源不再可用或正在执行其他任务,新平台仍继续存储数据。

[0020] 如下面更详细描述,所描述的系统和方法支持复制存储在数据库仓库中的大量数据。克隆技术用于创建与数据副本相关的新对象,而不实际复制数据本身。这种克隆技术简化了更新大批量数据和用特定数据集进行试验的过程。特别地,所描述的系统和方法能够识别与数据集相关联的原始目录对象,并且创建原始目录对象的副本而不复制数据本身。系统允许使用副本目录对象访问数据存储系统中的数据集。可以独立于副本目录对象来修改与原始目录对象关联的数据。在修改与原始目录对象关联的数据之后,系统可以删除副本目录对象或删除原始目录对象。如本文所讨论的,示例目录对象包括数据库实例,该数据库实例包括定义数据库对象的元数据,例如模式、表、视图、列、约束、序列、函数、文件格式、阶段等。其他示例目录对象可以包含任何类型的数据或数据结构。在讨论各种系统和方法时,“目录对象”也可以称为“模式对象”。

[0021] 在以下描述中,参考形成其一部分的附图,并且其中通过说明的方式示出了可以实践本公开的特定示例性实施例。足够详细地描述了这些实施例,以使得本领域技术人员能够实践本文公开的概念,并且应当理解,在不脱离本公开范围的情况下,可以对各种公开的实施例进行修改,并且可以利用其他实施例。因此,以下详细描述不应被视为具有限制意义。

[0022] 本说明书中对“一个实施例”、“实施例”、“一个示例”或“示例”的引用意味着结合该实施例或示例描述的特定特征、结构或特性包括在本公开的至少一个实施例中。因此,贯穿本说明书在各个地方出现的短语“在一个实施例中”、“在实施例中”、“一个示例”或“示例”不一定都指代相同的实施例或示例。另外,应当理解,本文提供的附图仅用于对本领域普通技术人员解释的目的,并且附图不一定按比例绘制。

[0023] 根据本公开的实施例可以体现为装置、方法或计算机程序产品。因此,本公开可以采取完全由硬件构成的实施例、完全由软件构成的实施例(包括固件、驻留软件、微代码等)的形式或者组合软件和硬件方面的实施例,所有这些在本文通常都称为“电路”、“模块”或“系统”。此外,本公开的实施例可以采用体现在任何有形表达介质中的计算机程序产品的形式,该介质具有包含在介质中的计算机可用程序代码。

[0024] 可以利用一个或多个计算机可用或计算机可读介质的任何组合。例如,计算机可读介质可以包括便携式计算机磁盘、硬盘、随机存取存储器(RAM)设备、只读存储器(ROM)设备、可擦除可编程只读存储器(EPROM或闪存)设备、便携式光盘只读存储器(CDROM)、光学存储设备和磁存储设备中的一个或多个。用于执行本公开的操作的计算机程序代码可以以一种或多种编程语言的任何组合来编写。这样的代码可以从源代码编译成适合于将在其上执行代码的设备或计算机的计算机可读汇编语言或机器代码。

[0025] 实施例还可以在云计算环境中实现。在本说明书和以下权利要求中,“云计算”可以被定义为用于实现对可配置计算资源的共享池(例如,网络、服务器、存储、应用和服务)的无处不在、方便的按需网络访问的模型,其可以通过虚拟化快速设置,并以最少的管理工作或服务提供商交互发布,然后相应地进行扩展。云模型可以由各种特性(例如,按需自助服务、广泛网络访问、资源池、快速弹性和测量服务)、服务模型(例如,软件即服务(“SaaS”)、平台即服务(“PaaS”)和基础架构即服务(“IaaS”))和部署模型(例如,私有云、社区云、公共云和混合云)构成。

[0026] 附图中的流程图和框图示出了根据本公开各种实施例的系统、方法和计算机程序产品的可能实现的架构、功能和操作。在这方面,流程图或框图中的每个框可以表示代码的模块、片段或部分,其包括用于实现指定的逻辑功能的一个或多个可执行指令。还应注意,可以由执行特定功能或动作的基于专用硬件的系统或专用硬件和计算机指令的组合来实现框图和/或流程图的每个框以及框图和/或流程图中的框的组合。这些计算机程序指令还可以存储在计算机可读介质中,该计算机可读介质可以指示计算机或其他可编程数据处理装置以特定方式起作用,使得存储在计算机可读介质中的指令产生包括实现流程图和/或框图块中指定的功能/动作的指令装置的制品。

[0027] 本文描述的系统和方法使用新的数据处理平台提供灵活且可扩展的数据仓库。在一些实施例中,所描述的系统和方法利用支持基于云的存储资源、计算资源等的云基础设施。示例的基于云的存储资源以低成本提供按需可用的大量存储容量。此外,这些基于云的存储资源可以是容错的并且高度可扩展,这在私有数据存储系统中的实现是成本高的。示例的基于云的计算资源是按需可用的,并且可以基于资源的实际使用级别来定价。通常,云基础架构以快速方式动态部署、重新配置和停用。

[0028] 在所描述的系统和方法中,数据存储系统利用基于SQL(结构化查询语言)的关系数据库。然而,这些系统和方法使用任何数据存储架构并使用任何语言来存储和检索数据存储和检索平台内的数据,适用于任何类型的数据库、以及任何类型的数据存储和检索平台。本文描述的系统和方法还提供多租户系统,其支持在不同客户/客户端之间以及同一客户/客户端内的不同用户之间隔离计算资源和数据。

[0029] 图1是描绘新数据处理平台100的示例实施例的框图。如图1所示,资源管理器102耦接到多个用户104、106和108。在特定实现中,资源管理器102可以支持希望访问数据处理

平台100的任何数量的用户。用户104-108可以包括,例如,提供数据存储和检索请求的最终用户、管理本文描述的系统和方法的系统管理员、以及与资源管理器102交互的其他组件/设备。资源管理器102提供支持数据处理平台100内所有系统和组件的操作的各种服务和功能。如本文所使用的,资源管理器102还可以被称为执行本文讨论的各种功能的“全局服务系统”。

[0030] 资源管理器102还耦接到元数据110,元数据110与整个数据处理平台100上存储的全部数据相关联。在一些实施例中,元数据110包括存储在远程数据存储系统中的数据目录以及从本地缓存可用的数据。在特定实施例中,存储在远程数据存储系统中的数据目录包括存储在远程数据存储系统中的数据的摘要。另外,元数据110可以包括关于如何在远程数据存储系统和本地高速缓存中组织数据的信息。元数据110允许系统和服务确定是否需要访问一条数据,而无需从存储设备加载或访问实际数据。

[0031] 资源管理器102还耦接到执行平台112,执行平台112提供执行各种数据存储和数据检索任务的多个计算资源,如下面更详细地讨论的。执行平台112耦接到作为存储平台114的一部分的多个数据存储设备116、118和120。尽管图1中示出了三个数据存储设备116、118和120,但是执行平台112能够与任意数量的数据存储设备通信。在一些实施例中,数据存储设备116、118和120是位于一个或多个地理位置中的基于云的存储设备。例如,数据存储设备116、118和120可以是公共云基础设施或私有云基础设施的一部分。数据存储设备116、118和120可以是硬盘驱动器(HDD)、固态驱动器(SSD)、存储集群、Amazon S3™存储系统或任何其他数据存储技术。另外,存储平台114可以包括分布式文件系统(诸如Hadoop分布式文件系统(HDFS))、对象存储系统等。

[0032] 在特定实施例中,经由一个或多个数据通信网络实现资源管理器102与用户104-108、元数据110和执行平台112之间的通信链路。类似地,经由一个或多个数据通信网络实现执行平台112与存储平台114中的数据存储设备116-120之间的通信链路。这些数据通信网络可以使用任何通信协议和任何类型的通信介质。在一些实施例中,数据通信网络是彼此耦接的两个或更多个数据通信网络(或子网络)的组合。在替代实施例中,使用任何类型的通信介质和任何通信协议来实现这些通信链路。

[0033] 如图1所示,数据存储设备116、118和120从与执行平台112相关联的计算资源解耦。该架构支持基于变化的数据存储/检索需求对数据处理平台100的动态改变以及访问数据处理平台100的用户和系统的变化的需求。动态变化的支持允许数据处理平台100响应于对数据处理平台100内的系统和组件的变化的需求而快速扩展。计算资源从数据存储设备解耦支持大量数据的存储,而不需要相应的大量计算资源。类似地,这种资源的解耦支持在特定时间利用的计算资源的显著增加,而不需要相应增加可用数据存储资源。

[0034] 资源管理器102、元数据110、执行平台112和存储平台114在图1中示为单独的组件。然而,资源管理器102、元数据110、执行平台112和存储平台114中的每一个可以实现为分布式系统(例如,分布在多个地理位置处的多个系统/平台上)。另外,资源管理器102、元数据110、执行平台112和存储平台114中的每一个可以根据从用户104-108接收的请求的变化以及数据处理平台100的变化需求而按比例放大或缩小(彼此独立)。因此,在所描述的实施例中,数据处理平台100是动态的并且支持定期改变以满足当前数据处理需求。

[0035] 在典型操作期间,数据处理平台100处理从任何用户104-108接收的多个查询(或

请求)。这些查询由资源管理器102管理,以确定何时以及如何执行查询。例如,资源管理器102可以确定处理查询所需的数据,并进一步确定执行平台112内的哪些节点最适合处理查询。一些节点可能已经缓存了处理查询所需的数据,因此是处理查询的良好候选者。元数据110帮助资源管理器102确定执行平台112中的哪些节点已经缓存处理查询所需的至少一部分数据。执行平台112中的一个或多个节点使用由节点缓存的数据来处理查询,并且如果需要,使用从存储平台114检索的数据。期望从执行平台112内的高速缓存中检索尽可能多的数据,因为检索速度通常比从存储平台114检索数据快得多。

[0036] 如图1所示,数据处理平台100将执行平台112与存储平台114分离。在该布置中,执行平台112中的处理资源和缓存资源独立于存储平台114中的数据存储空间116-120而操作。因此,计算资源和缓存资源不限于特定的数据存储空间116-120。相反,所有计算资源和所有缓存资源可以从存储平台114中的任何数据存储空间检索数据并将数据存储在存储平台114中的任何数据存储空间。另外,数据处理平台100支持向执行平台112添加新的计算资源和缓存资源,而不需要对存储平台114进行任何改变。类似地,数据处理平台100支持向存储平台114添加数据存储空间,而不需要对执行平台112中的节点进行任何改变。

[0037] 图2是描绘资源管理器102的实施例的框图。如图2所示,资源管理器102包括耦接到数据存储设备206的密钥管理器204和访问管理器202。访问管理器202处理本文描述的系统的认证和授权任务。密钥管理器204管理在认证和授权任务期间使用的密钥的存储和认证。例如,访问管理器202和密钥管理器204管理用于访问存储在远程存储设备(例如,存储平台114中的数据存储设备)中的数据的密钥。如本文所使用的,远程存储设备也可以称为“持久存储设备”。请求处理服务208管理接收的数据存储请求和数据检索请求(例如,数据库查询)。例如,请求处理服务208可以确定处理接收的数据存储请求或数据检索请求所需的数据。必要的可以存储在执行平台112内的高速缓存中(如下面更详细地讨论的)或存储平台114中的数据存储设备中。管理控制台服务210支持管理员和其他系统管理器对各种系统和过程的访问。另外,管理控制台服务210可以从用户104-108接收请求,以发出查询并监测系统上的工作负载。在一些实施例中,特定用户可以发出请求以监测其在系统上放置特定查询的工作负载。

[0038] 资源管理器102还包括SQL编译器212、SQL优化器214和SQL执行器216。SQL编译器212解析SQL查询并生成用于查询的执行代码。SQL优化器214基于需要处理的数据确定执行查询的最佳方法。SQL优化器214还处理各种数据修剪操作和其他数据优化技术,以提高执行SQL查询的速度和效率。SQL执行器216为资源管理器102接收的查询执行查询代码。

[0039] 查询调度器和协调器218将所接收的查询发送到适当的服务或系统以用于编译、优化和分派到执行平台112。例如,可以按优先顺序对查询进行优先级排序和处理。在一些实施例中,查询调度器和协调器218识别或分配执行平台112中的特定节点以处理特定查询。虚拟仓库管理器220管理在执行平台112中实现的多个虚拟仓库的操作。如下所讨论的,每个虚拟仓库包括多个执行节点,每个执行节点包括高速缓存和处理器。

[0040] 另外,资源管理器102包括配置和元数据管理器222,其管理与存储在远程数据存储设备和本地高速缓存(即,执行平台112中的高速缓存)中的数据有关的信息。如下面更详细讨论的,配置和元数据管理器222使用元数据来确定需要访问哪些数据文件以检索用于处理特定查询的数据。监测和工作负载分析器224监视由资源管理器102执行的过程,并管

理在执行平台112中的虚拟仓库和执行节点之间的任务(例如,工作负载)的分配。监测和工作负载分析器224还基于整个数据处理平台100中的工作负载的改变,根据需要重新分配任务。配置和元数据管理器222以及监测和工作负载分析器224耦接到数据存储设备226。图2中的数据存储设备206和226表示数据处理平台100内的任何数据存储设备。例如,数据存储设备206和226可以表示执行平台112中的高速缓存、存储平台114中的存储设备或任何其他存储设备。

[0041] 资源管理器102还包括事务管理和访问控制模块228,其管理与数据存储请求和数据访问请求的处理相关联的各种任务和其他活动。例如,事务管理和访问控制模块228提供多个用户或系统对数据的一致和同步访问。由于多个用户/系统可以同时访问相同的数据,因此必须同步对数据的改变,以确保每个用户/系统正在使用当前版本的数据。事务管理和访问控制模块228在资源管理器102中的单个集中位置处提供对各种数据处理活动的控制。在一些实施例中,事务管理和访问控制模块228与SQL执行器216交互,以支持由SQL执行器216正在执行的各种任务的管理。

[0042] 图3是描绘执行平台112的实施例的框图。如图3所示,执行平台112包括多个虚拟仓库302、304和306。每个虚拟仓库包括多个执行节点,每个执行节点包括数据高速缓存和处理器。虚拟仓库302、304和306能够通过使用多个执行节点并行地执行多个查询(和其他任务)。如本文所讨论的,执行平台112可以基于系统和用户的当前处理需求来实时添加新的虚拟仓库并丢弃现有的虚拟仓库。这种灵活性允许执行平台112在需要时快速部署大量计算资源,而不必在不再需要时继续为这些计算资源付出代价。所有虚拟仓库都可以从任何数据存储设备(例如,存储平台114中的任何存储设备)访问数据。

[0043] 尽管图3中示出的每个虚拟仓库302-306包括三个执行节点,但是特定虚拟仓库可以包括任何数量的执行节点。此外,虚拟仓库中的执行节点的数量是动态的,使得当存在附加需求时创建新的执行节点,并且当不再需要现有的执行节点时删除它们。

[0044] 每个虚拟仓库302-306能够访问图1中所示的任何数据存储设备116-120。因此,虚拟仓库302-306不一定被分配给特定数据存储设备116-120,而是可以从任何数据存储设备116-120访问数据。类似地,图3中所示的每个执行节点可以访问来自任何数据存储设备116-120的数据。在一些实施例中,可以将特定虚拟仓库或特定执行节点临时分配给特定数据存储设备,但是虚拟仓库或执行节点可以稍后从任何其他数据存储设备访问数据。

[0045] 在图3的示例中,虚拟仓库302包括三个执行节点308、310和312。执行节点308包括高速缓存314和处理器316。执行节点310包括高速缓存318和处理器320。执行节点312包括高速缓存322和处理器324。每个执行节点308-312与处理一个或多个数据存储和/或数据检索任务相关联。例如,特定虚拟仓库可以处理与特定用户或客户相关联的数据存储和数据检索任务。在其他实现中,特定虚拟仓库可以处理与特定数据存储系统或特定类别的数据相关联的数据存储和数据检索任务。

[0046] 类似于上面讨论的虚拟仓库302,虚拟仓库304包括三个执行节点326、328和330。执行节点326包括高速缓存332和处理器334。执行节点328包括高速缓存336和处理器338。执行节点330包括高速缓存340和处理器342。另外,虚拟仓库306包括三个执行节点344、346和348。执行节点344包括高速缓存350和处理器352。执行节点346包括高速缓存354和处理器356。执行节点348包括高速缓存358和处理器360。

[0047] 在一些实施例中,图3中所示的执行节点相对于执行节点正在高速缓存的数据是无状态的。例如,这些执行节点不存储或以其他方式维护关于执行节点的状态信息或由特定执行节点高速缓存的数据。因此,在执行节点发生故障的情况下,故障节点可以被另一个节点透明地替换。由于没有与故障的执行节点相关联的状态信息,所以新(替换)执行节点可以容易地替换故障的节点,而无需重新创建特定状态。

[0048] 尽管图3中所示的执行节点每个包括一个数据高速缓存和一个处理器,但是替代实施例可以包括包含任意数量的处理器和任何数量的高速缓存的执行节点。另外,不同的执行节点的高速缓存的大小可以不同。图3中所示的高速缓存在本地执行节点中存储从存储平台114(图1)中的一个或多个数据存储设备检索的数据。因此,高速缓存减少或消除了始终从远程存储系统检索数据的平台中出现的瓶颈问题。本文描述的系统和方法不是从远程存储设备重复访问数据,而是从执行节点中的高速缓存访问数据,这明显更快并且避免了上面讨论的瓶颈问题。在一些实施例中,使用提供对高速缓存数据的快速访问的高速存储器设备来实现高速缓存。每个高速缓存可以存储来自存储平台114中的任何存储设备的数据。

[0049] 此外,缓存资源和计算资源可以在不同的执行节点之间变化。例如,一个执行节点可以包含大量计算资源和最小的缓存资源,使得执行节点对于需要大量计算资源的任务有用。另一个执行节点可以包含大量缓存资源和最小的计算资源,使得该执行节点对需要缓存大量数据的任务有用。又一个执行节点可以包含提供更快输入-输出操作的高速缓存资源,对于需要快速扫描大量数据的任务是有益的。在一些实施例中,基于将由执行节点执行的预期任务,在创建执行节点时确定与特定执行节点相关联的缓存资源和计算资源。

[0050] 另外,与特定执行节点相关联的高速缓存资源和计算资源可以基于由执行节点执行的变化任务而随时间改变。例如,如果执行节点执行的任务变得更加处理器密集,则可以为其特定执行节点分配更多处理资源。类似地,如果执行节点执行的任务需要更大的高速缓存容量,则可以为其执行节点分配更多的高速缓存资源。

[0051] 尽管虚拟仓库302-306与相同的执行平台112相关联,但是可以使用多个地理位置处的多个计算系统来实现虚拟仓库。例如,可以由第一地理位置处的计算系统实现虚拟仓库302,而由第二地理位置处的另一计算系统实现虚拟仓库304和306。在一些实施例中,这些不同的计算系统是由一个或多个不同实体维护的基于云的计算系统。

[0052] 另外,每个虚拟仓库在图3中示为具有多个执行节点。可以使用多个地理位置处的多个计算系统来实现与每个虚拟仓库相关联的多个执行节点。例如,虚拟仓库302的特定实例在特定地理位置处的一个计算平台上实现执行节点308和310,并且在另一个地理位置处的不同计算平台处实现执行节点312。选择特定计算系统以实现执行节点可取决于各种因素,诸如特定执行节点所需的资源级别(例如,处理资源需求和高速缓存要求)、特定计算系统处可用的资源、地理位置内或地理位置之间的网络的通信能力、以及哪些计算系统已经在虚拟仓库中实现其他执行节点。

[0053] 执行平台112也是容错的。例如,如果一个虚拟仓库发生故障,则该虚拟仓库将很快被不同地理位置处的不同虚拟仓库替换。

[0054] 特定执行平台112可包括任何数量的虚拟仓库302-306。另外,特定执行平台中的虚拟仓库的数量是动态的,使得在需要额外的处理和/或高速缓存资源时创建新的虚拟仓

库。类似地,当不再需要与虚拟仓库相关联的资源时,可以删除现有的虚拟仓库。

[0055] 在一些实施例中,虚拟仓库302、304和306可以对存储平台114中的相同数据进行操作,但是每个虚拟仓库具有其自己的具有独立处理和高速缓存资源的执行节点。此配置允许独立处理不同虚拟仓库上的请求,并且请求之间不会产生干扰。这种独立处理与动态添加和去除虚拟仓库的能力相结合,支持为新用户增加新的处理能力,而不会影响现有用户观察到的性能。

[0056] 图4是描绘具有多个用户通过多个虚拟仓库访问多个数据库的示例操作环境400的框图。在环境400中,多个用户402、404和406通过多个虚拟仓库408、410和412访问多个数据库414、416、418、420、422和424。虽然图4中未示出,但是用户402、404和406可以通过资源管理器102(图1)访问虚拟仓库408、410和412。在特定实施例中,数据库414-424包含在存储平台114(图1)中,并且可由在执行平台112中实现的任何虚拟仓库访问。在一些实施例中,用户402-406使用诸如互联网的数据通信网络访问虚拟仓库408-412中的一个。在一些实现中,每个用户402-406指定要在特定时间工作的特定虚拟仓库408-412。在图4的示例中,用户402与虚拟仓库408交互,用户404与虚拟仓库410交互,并且用户406与虚拟仓库412交互。因此,用户402通过虚拟仓库408提交数据检索和数据存储请求。类似地,用户404和406分别通过虚拟仓库410和412提交数据检索和数据存储请求。

[0057] 每个虚拟仓库408-412被配置为与所有数据库414-424的子集通信。例如,在环境400中,虚拟仓库408被配置为与数据库414、416和422通信。类似地,虚拟仓库410被配置为与数据库416、418、420和424通信。并且,虚拟仓库412被配置为与数据库416、422和424通信。在替代实施例中,虚拟仓库408-412中的一个或多个与所有数据库414-424通信。图4中所示的布置允许各个用户通过单个虚拟仓库发送所有数据检索和数据存储请求。该虚拟仓库使用虚拟仓库中多个执行节点之一内的缓存数据处理数据检索和数据存储任务,或从适当的数据库检索(和缓存)必要的数据库。虚拟仓库之间的映射是逻辑映射,而不是硬件映射。此逻辑映射基于与安全性和资源访问管理设置相关的访问控制参数。无需重新配置虚拟仓库或存储资源即可轻松更改逻辑映射。

[0058] 尽管环境400示出了被配置为与数据库414-424的特定子集通信的虚拟仓库408-412,但是该配置是动态的。例如,可以基于由虚拟仓库408执行的变化任务来重新配置虚拟仓库408以与数据库414-424的不同子集通信。例如,如果虚拟仓库408接收到从数据库418访问数据的请求,则虚拟仓库408可以被重新配置为还与数据库418通信。如果在稍后的时间,虚拟仓库408不再需要从数据库418访问数据,则可以重新配置虚拟仓库408以删除与数据库418的通信。

[0059] 图5是描绘另一示例操作环境500的框图,其中多个用户通过负载平衡器和虚拟仓库组中包含的多个虚拟仓库访问多个数据库。环境500类似于环境400(图4),但另外包括虚拟仓库资源管理器508和布置在虚拟仓库组516中的多个虚拟仓库510、512和514。虚拟仓库资源管理器508可以包含在资源管理器102中。具体地,多个用户502、504和506通过虚拟仓库资源管理器508和虚拟仓库组516访问多个数据库518、520、522、524、526和528。在一些实施例中,用户502-506使用诸如互联网的数据通信网络访问虚拟仓库资源管理器508。尽管未在图5中示出,但是用户502、504和506可以通过资源管理器102(图1)访问虚拟仓库资源管理器508。在一些实施例中,虚拟仓库资源管理器508在资源管理器102内实现。

[0060] 用户502-506可以向虚拟仓库资源管理器508提交数据检索和数据存储请求,虚拟仓库资源管理器508将数据检索和数据存储请求按某路线发送到虚拟仓库组516中的适当虚拟仓库510-514。在一些实现中,虚拟仓库资源管理器508向虚拟仓库510-514提供用户502-506的动态分配。当提交数据检索或数据存储请求时,用户502-506可以指定虚拟仓库组516来处理请求,而不指定将处理该请求的特定虚拟仓库510-514。这种安排允许虚拟仓库资源管理器508基于虚拟仓库510-514内的高效、可用资源和高速缓存数据的可用性,在虚拟仓库510-514之间分发多个请求。当确定如何发送数据处理请求时,虚拟仓库资源管理器508考虑可用资源、当前资源负载、当前用户的数量等。

[0061] 在一些实施例中,容错系统响应于虚拟仓库的故障而创建新的虚拟仓库。新虚拟仓库可以位于相同的虚拟仓库组中,或者可以在不同地理位置处的不同虚拟仓库组中创建。

[0062] 每个虚拟仓库510-514被配置为与所有数据库518-528的子集通信。例如,在环境500中,虚拟仓库510被配置为与数据库518、520和526通信。类似地,虚拟仓库512被配置为与数据库520、522、524和528通信。并且,虚拟仓库514被配置为与数据库520、526和528通信。在替代实施例中,虚拟仓库510-514可以与任何(或所有)数据库518-528通信。

[0063] 尽管环境500示出了一个虚拟仓库组516,但是替代实施例可以包括任意数量的虚拟仓库组,每个虚拟仓库组与任意数量的虚拟仓库相关联。特定环境中的虚拟仓库组的数量是动态的,并且可以根据环境中其他系统和用户的变化需求而发生变化。

[0064] 图6是描绘具有多个分布式虚拟仓库和虚拟仓库组的另一示例操作环境600的框图。环境600包括通过数据通信网络602与虚拟仓库组604和606通信的资源管理器102。仓库组604包括两个虚拟仓库608和610,并且仓库组606包括另外两个虚拟仓库614和616。资源管理器102还通过数据通信网络602与虚拟仓库612(其不是虚拟仓库组的一部分)通信。

[0065] 虚拟仓库组604和606以及虚拟仓库612通过数据通信网络618与数据库620、622和624通信。在一些实施例中,数据通信网络602和618是相同的网络。环境600允许资源管理器102跨多个虚拟仓库608-616协调用户数据存储和检索请求,以在数据库620-624中存储和检索数据。虚拟仓库组604和606可以位于相同的地理区域中,或者可以在地理上分开。另外,虚拟仓库组604和606可以由同一实体或不同实体实现。

[0066] 本文描述的系统和方法允许将数据作为与计算(或处理)资源分开的服务来存储和访问。即使没有从执行平台分配计算资源,数据也可用于虚拟仓库,而无需从远程数据源重新加载数据。因此,数据可独立于与数据相关联的计算资源的分配而获得。所描述的系统和方法适用于任何类型的数据。在特定实施例中,数据以结构化的优化格式存储。数据存储/访问服务与计算服务的解耦还简化了不同用户和组之间的数据共享。如本文所讨论的,即使在其他虚拟仓库访问相同数据的同时,每个虚拟仓库也可以访问其具有访问权限的任何数据。此架构支持连续的查询,而无需在本地缓存中存储的任何实际数据。本文描述的系统和方法能够进行透明的动态数据移动,其根据需要以对系统用户透明的方式,将数据从远程存储设备移动到本地高速缓存。此外,该架构支持数据共享而无需先前的数据移动,因为由于数据存储服务与计算服务的解耦,任何虚拟仓库都可以访问任何数据。

[0067] 图7是描绘用于管理数据存储和检索操作的方法700的实施例的流程图。最初,方法700在702处从用户接收语句、请求或查询。语句是执行数据相关操作的任何请求或命令。

示例语句包括数据检索请求、数据存储请求、数据传输请求、数据查询等。在一些实施例中，语句被实现为SQL语句。资源管理器在704处创建查询协调器以管理接收到的语句。例如，查询协调器管理处理接收到的语句所需的各种任务，包括与执行平台和一个或多个数据存储设备交互。在一些实施例中，查询协调器是专门为管理接收到的语句而创建的临时例程。

[0068] 方法700继续，在706处资源管理器确定处理接收到的语句所需的多个任务。多个任务可包括例如从执行节点中的高速缓存访问数据、从远程存储设备检索数据、更新高速缓存中的数据、将数据存储到远程存储设备中等。资源管理器还在708处将多个任务分发到执行平台中的执行节点。如本文所讨论的，执行平台中的执行节点在虚拟仓库内实现。在710处每个执行节点执行分配的任务并将任务结果返回给资源管理器。在一些实施例中，执行节点将任务结果返回给查询协调器。资源管理器在712处接收多个任务结果并创建语句结果，并在714处将语句结果传送给用户。在一些实施例中，在将语句结果传送给用户之后，删除查询协调器。

[0069] 在一些实现中，同一文件由多个执行节点同时高速缓存。这种文件的多重高速缓存有助于多个执行节点之间的负载平衡（例如，平衡数据处理任务）。此外，在多个执行节点中缓存文件有助于避免在大量数据试图通过同一通信链路时出现潜在瓶颈。该实现还支持不同执行节点对相同数据的并行处理。

[0070] 本文描述的系统和方法利用共享磁盘系统和无共享架构的益处。一旦数据在本地缓存，所描述的用于存储和检索数据的平台就像无共享架构一样可扩展。它还具有共享磁盘架构的所有优点，其中可以在没有任何约束的情况下添加和去除处理节点（例如，对于0到N）并且不需要任何显式的数据改组。

[0071] 在先前的数据管理系统中，在数据仓库中复制大量数据是耗时的并且需要大量资源来维护和处理相同数据的多个副本。此外，在复制数据并将更改应用于其中一个数据集时，这些先前系统中存在数据不一致。

[0072] 然而，本文描述的用于克隆目录对象的系统和方法生成新的目录对象，该目录对象被快速激活并且独立于原始目录对象。如下所述，创建新目录对象时不会复制与目录对象关联的数据。相反，只复制与目录对象关联的元数据，这比尝试复制数据本身快得多。例如，通过仅复制与目录对象相关联的元数据可能比复制所有数据快几个数量级。在示例情况下，复制1KB的元数据而不是1MB的实际数据。

[0073] 如本文所使用的，表是数据的逻辑分组，而模式是一组表。示例目录对象包括例如表、数据库、模式、帐户、约束、文件格式、函数、角色、序列、阶段、列、用户和卷。以下层级表示与本文描述的系统和方法一起使用的对象的示例层级。

[0074] 帐户-数据库-模式-表/视图/序列-列/约束

[0075] 帐户-数据库-模式-函数/文件格式/阶段

[0076] 帐户-用户

[0077] 帐户-角色

[0078] 帐户-卷

[0079] 帐户-仓库

[0080] 在一些实施例中，这些目录对象由SQL标准定义。如本文所讨论的，表包含多个行和列。模式包含任意数量的表，数据库包含任意数量的模式。本文描述的系统和方法能够克

隆表、模式和数据库。在一些实现中,可以克隆各个表,而其他实现可以克隆整个数据库,这取决于正在执行的数据操作。此外,克隆操作可以克隆根目录对象下的所有子代。由于克隆的更高级别对象独立于原始对象,因此可以将新的子目录对象添加到克隆的对象(或者可以从克隆的对象中去除克隆的子目录对象),而不会影响原始的更高级别对象及其层次结构。

[0081] 克隆目录对象快速创建相同类型的新目录对象,而不创建数据的附加副本。例如,如果正在克隆数据库,则克隆过程会在该数据库对象下创建新的数据库对象以及所有表和模式以及其他目录对象,但不会复制数据库中包含的数据。克隆过程完成后,可以相互独立地修改克隆的目录对象和原始目录对象。此外,可以将对象添加至目录对象的每个副本以及原始目录对象并且可以从目录对象的每个副本以及原始目录对象中去除对象。

[0082] 图8是描绘用于克隆目录对象的方法的实施例的流程图。最初,方法800在802处识别要克隆的数据库。该数据库可以包含任何数量的模式。方法800继续,在804处识别与数据库相关联的多个模式并在806处选择所识别的多个模式中的一个。该方法然后在808处识别与所选择的模式相关联的多个表和其他目录对象。在810处选择所识别的多个表中的一个,在812处识别与所选择的表相关联的所有元数据。在一些实施例中,该元数据包括存储在数据库中的数据的摘要。如本文所讨论的,元数据可以包括关于如何在数据库、表或其他存储机制中组织数据的信息。元数据允许本文讨论的系统和方法确定关于存储数据的信息,而无需从存储设备加载或访问实际数据。

[0083] 方法800在814处通过复制与所选择的表相关联的元数据来克隆所选择的表。与复制与元数据相关联的所有数据所需的时间相比,此元数据的复制执行得很快速。在814处克隆表之后,该方法在816处确定所选择的模式中是否存在需要克隆的其他表。如果还有其他表要克隆,则方法800返回到810以选择模式中的下一个表。如果克隆了所选择的模式中的所有表,则该方法在818处确定在所识别的数据库中是否存在要克隆的其他模式。如果还有其他模式要克隆,则方法800返回到806以选择数据库中的下一个模式。此递归操作确保克隆所识别的数据库的所有模式中的所有表。在克隆了所识别的数据库中的所有模式之后,可以在820处独立于克隆的数据库来修改原始数据库、其模式、表和其他子目录对象。

[0084] 在一些实施例中,将克隆表中的数据用于当前数据访问请求,同时修改原始表中的数据。完成原始表的修改后,将删除克隆的表,并使用原始表处理所有数据访问请求。在这些实施例中,在不复制实际底层数据的情况下复制元数据显著地提高了数据修改过程的速度,并减少了执行数据修改过程所需的额外计算和存储资源。在一些实现中,数据修改过程由事务管理和访问控制模块228(图2)管理。另外,在一些实施例中,元数据由配置和元数据管理器222管理和访问。

[0085] 图9是描绘用于创建临时副本目录对象的方法900的实施例的流程图。最初,方法900在902处识别与一数据集相关联的原始目录对象。在一些实施例中,原始目录对象表示数据存储系统中的数据的逻辑分组。原始目录对象可以包括表、数据库、模式、帐户、约束、文件格式、函数、角色、序列、阶段、列、用户、卷、或其他目录对象中的一个或多个。在904处创建原始目录对象的副本而不复制数据本身。在一些实施例中,创建原始目录对象的副本包括将与该数据集相关联的元数据复制到副本目录对象。该元数据可以包括存储在该数据集中的数据的清单,并且可以识别关于该数据集的信息,而无需访问该数据集中包含的实

际数据。在特定实现中,副本目录对象是只读的。在一些实施例中,副本目录对象包括所有子代的副本层次结构。

[0086] 该方法允许使用副本目录对象访问(906)数据集,其中可以在908处独立于副本目录对象来修改与原始目录对象相关联的数据。在一些实施例中,与原始目录对象相关联的修改后的数据对副本目录对象不可见。此外,与副本目录对象相关联的修改后的数据可以对原始目录对象不可见。在其他实施例中,从原始目录对象删除的数据对于副本目录对象保持可见,并且从副本目录对象删除的数据对于原始目录对象保持可见。在特定实现中,与原始目录对象相关联的插入数据对于副本目录对象是不可见的,并且与副本目录对象相关联的插入数据对于原始目录对象是不可见的。在完成对与原始目录对象相关联的数据的修改之后,在910处删除副本目录对象。与制作包括所有相关联数据的原始目录对象的完整副本相比,创建副本目录对象而不复制数据本身可以节省大量的时间、带宽和计算资源。

[0087] 图10A-图10D说明了表的克隆。在图10A-图10D所示的实施例中,每个表包含多个文件的集合。如下所述,一旦写入文件,就无法更新或修改。如果需要更改文件,则删除该文件并替换为包含更改的其他文件。如果某个特定文件未被使用,则可能会将其删除。上面讨论的元数据可以包括关于表和与该表相关联的文件之间的关系的信息。表和文件之间的这种关系也可以称为文件的映射。

[0088] 图10A示出了具有*i*个文件(标记为F₁、F₂...F_i)的示例表T₁。图10B示出了克隆表T₁以创建克隆表T₂的结果。在该实施例中,克隆过程将来自表T₁的元数据复制到克隆表T₂。如本文所讨论的,复制元数据比复制所有底层数据快得多。图10B示出了紧接着在克隆过程之后,表T₁和表T₂与相同文件相关联。

[0089] 在创建克隆表T₂之后,可以独立于另一表将新文件添加到表T₁或T₂。此外,可以从表T₁或T₂中删除文件,而与另一表无关。例如,图10C示出了表T₁删除文件F₂和添加的新文件F_j之后的表和文件关联。删除的文件F₂对于表T₂仍然可见,但是新文件F_j对于表T₂不可见。在图10D中,表T₂删除了文件F₁并添加了新文件T_k。文件F₁对表T₁仍然可见,但新文件T_k对表T₁不可见。

[0090] 尽管图10A-图10D示出了表的克隆,但是替代实施例可以以类似的方式克隆目录对象、模式和数据库。在特定实施例中,维护模式中存在的的所有表的映射以及数据库中存在的所有模式的映射。在克隆之后,可以相互独立地修改两个模式(或两个数据库)。

[0091] 本文讨论的克隆系统和方法改善了新数据或实验数据的周期性加载(和分析)。另外,这些系统和方法支持新数据的快速更新,同时需要最少的额外计算或数据存储资源。

[0092] 图11是描绘示例计算设备1100的框图。在一些实施例中,计算设备1100用于实现本文讨论的系统和组件中的一个或多个。例如,计算设备1100可以允许用户或管理员访问资源管理器102。此外,计算设备1100可以与本文描述的任何系统和组件交互。因此,计算设备1100可以用于执行各种过程 and 任务,例如本文所讨论的那些过程和任务。计算设备1100可以用作服务器、客户端或任何其他计算实体。计算设备1100可以是各种计算设备中的任何一种,例如台式计算机、笔记本计算机、服务器计算机、手持式计算机、平板电脑等。

[0093] 计算设备1100包括一个或多个处理器1102、一个或多个存储器设备1104、一个或多个接口1106、一个或多个大容量存储设备1108以及一个或多个输入/输出(I/O)设备1110,所有这些都耦接到总线1112。处理器1102包括执行存储在存储器设备1104和/或大容量

量存储设备1108中的指令的一个或多个处理器或控制器。处理器1102还可以包括各种类型的计算机可读介质,例如高速缓冲存储器。

[0094] 存储器设备1104包括各种计算机可读介质,诸如易失性存储器(例如,随机存取存储器(RAM))和/或非易失性存储器(例如,只读存储器(ROM))。存储器设备1104还可以包括可重写ROM,例如闪存。

[0095] 大容量存储设备1108包括各种计算机可读介质,诸如磁带、磁盘、光盘、固态存储器(例如,闪存)等。各种驱动器也可以包括在大容量存储设备1108中,以使得能够从各种计算机可读介质读取和/或向各种计算机可读介质写入。大容量存储设备1108包括可移动介质和/或不可移动介质。

[0096] I/O设备1110包括允许将数据和/或其他信息输入到计算设备1100或从计算设备1100检索数据和/或其他信息的各种设备。示例I/O设备1110包括光标控制设备、键盘、小键盘、麦克风、监视器或其他显示设备、扬声器、打印机、网络接口卡、调制解调器、镜头、CCD或其他图像捕获设备等。

[0097] 接口1106包括允许计算设备1100与其他系统、设备或计算环境交互的各种接口。示例接口1106包括任何数量的不同网络接口,例如到局域网(LAN)、广域网(WAN)、无线网络和互联网的接口。

[0098] 总线1112允许处理器1102、存储器设备1104、接口1106、大容量存储设备1108和I/O设备1110彼此通信,以及与耦接到总线1112的其他设备或组件通信。总线1112表示若干类型的总线结构中的一种或多种,例如系统总线、PCI总线、IEEE 1394总线、USB总线等等。

[0099] 出于说明的目的,程序和其它可执行程序组件在本文中显示为离散块,但应理解,此类程序和组件可在不同时间驻留在计算设备1100的不同存储组件中,且由处理器1102执行。或者,本文描述的系统 and 过程可以用硬件或硬件、软件和/或固件的组合来实现。例如,一个或多个专用集成电路(ASIC)可以编程为执行本文描述的一个或多个系统和过程。

[0100] 尽管根据某些优选实施例描述了本公开,但是鉴于本公开的益处,其他实施例对于本领域普通技术人员将是明显的,包括不提供本文列举的所有益处和特征的实施例,这些也在本公开的范围之内。应当理解,在不脱离本公开的范围的情况下,可以使用其他实施例。

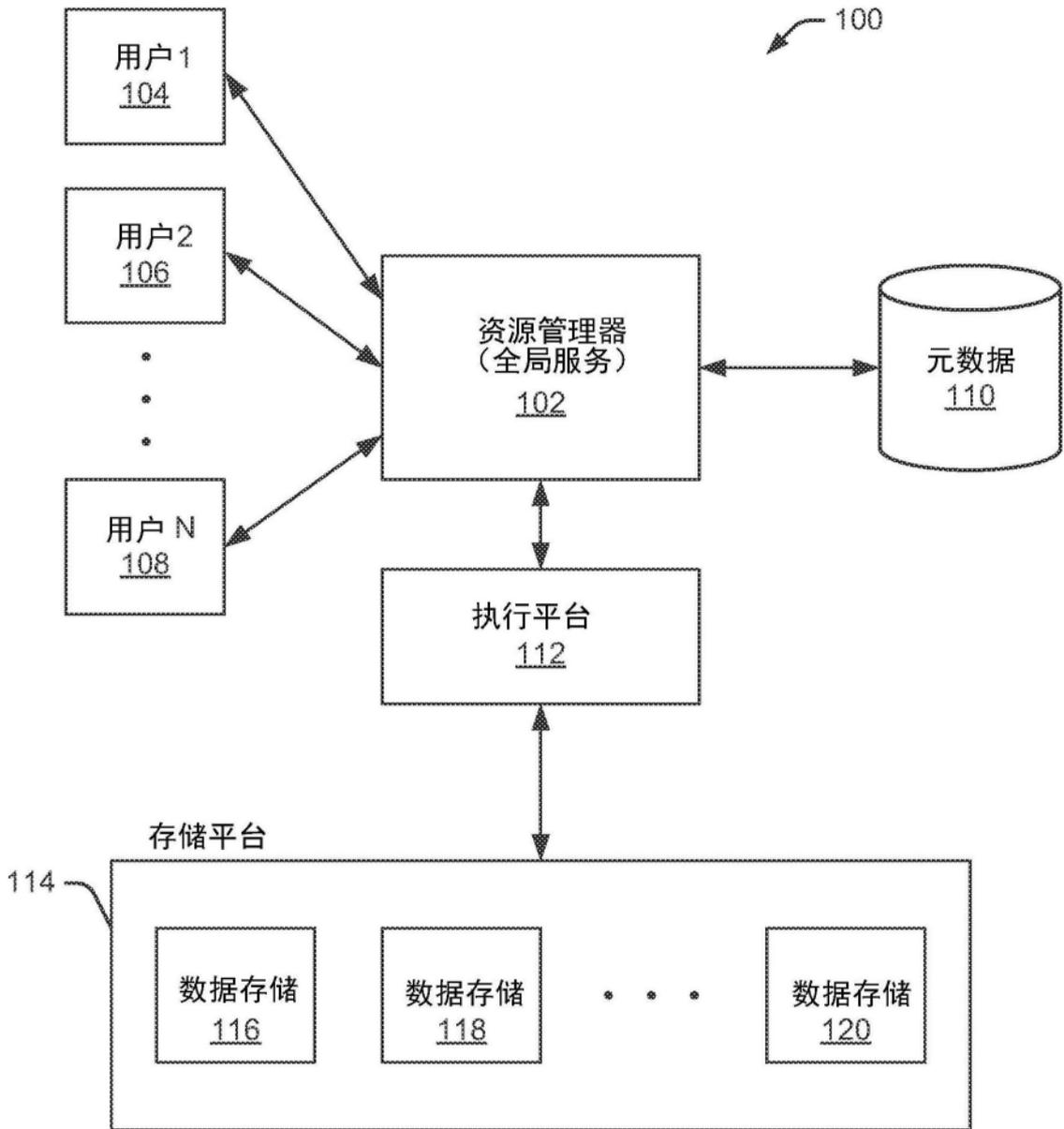


图1

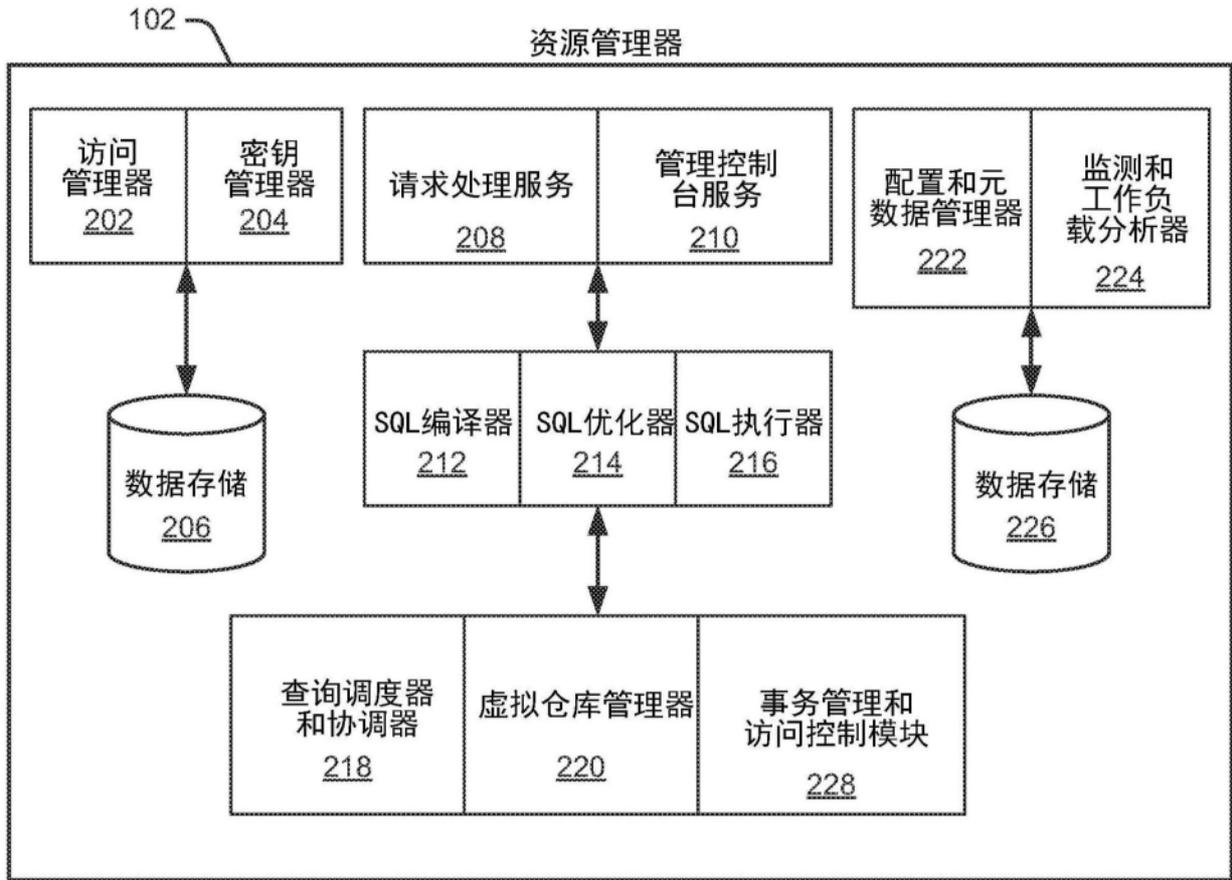


图2

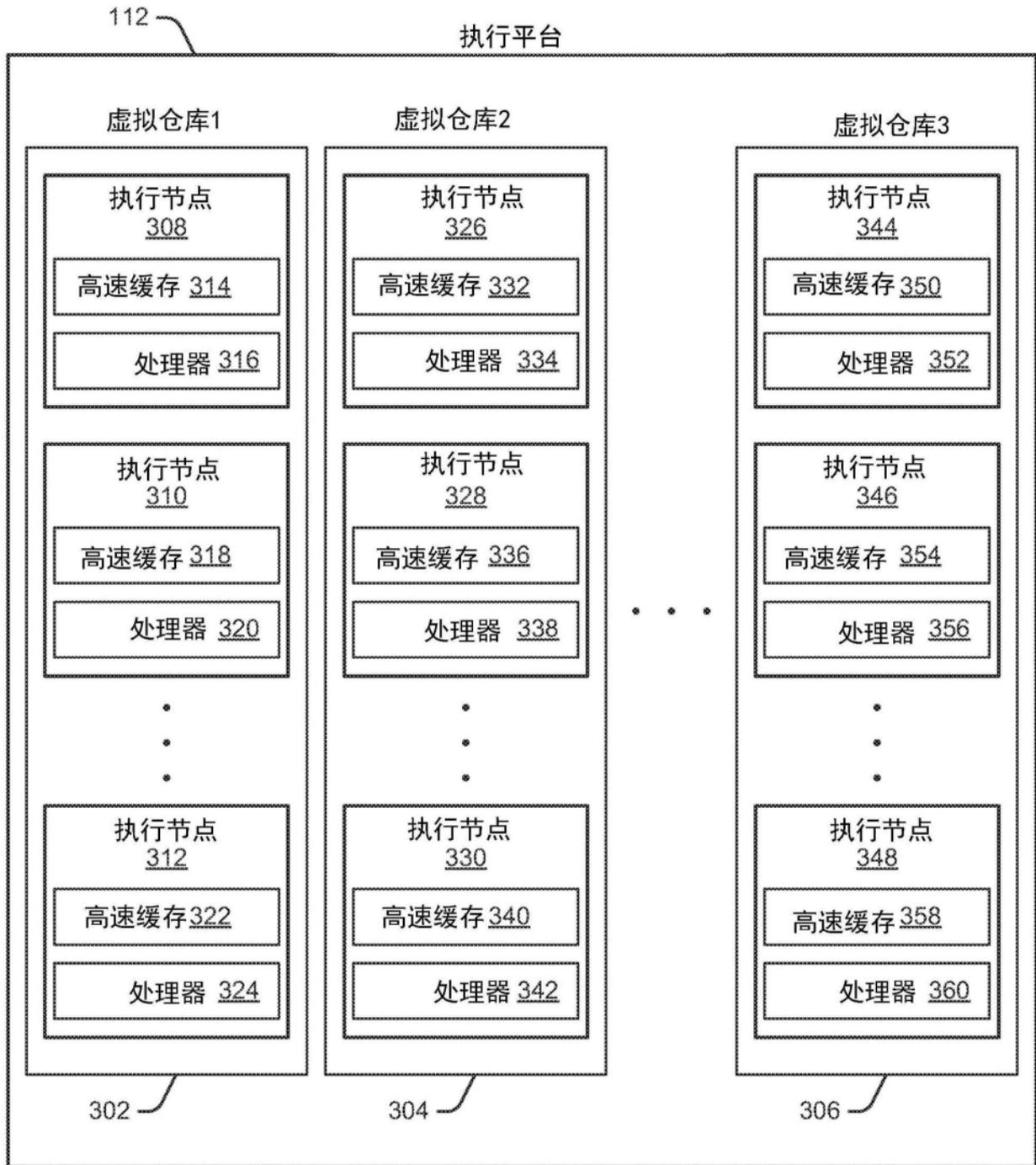


图3

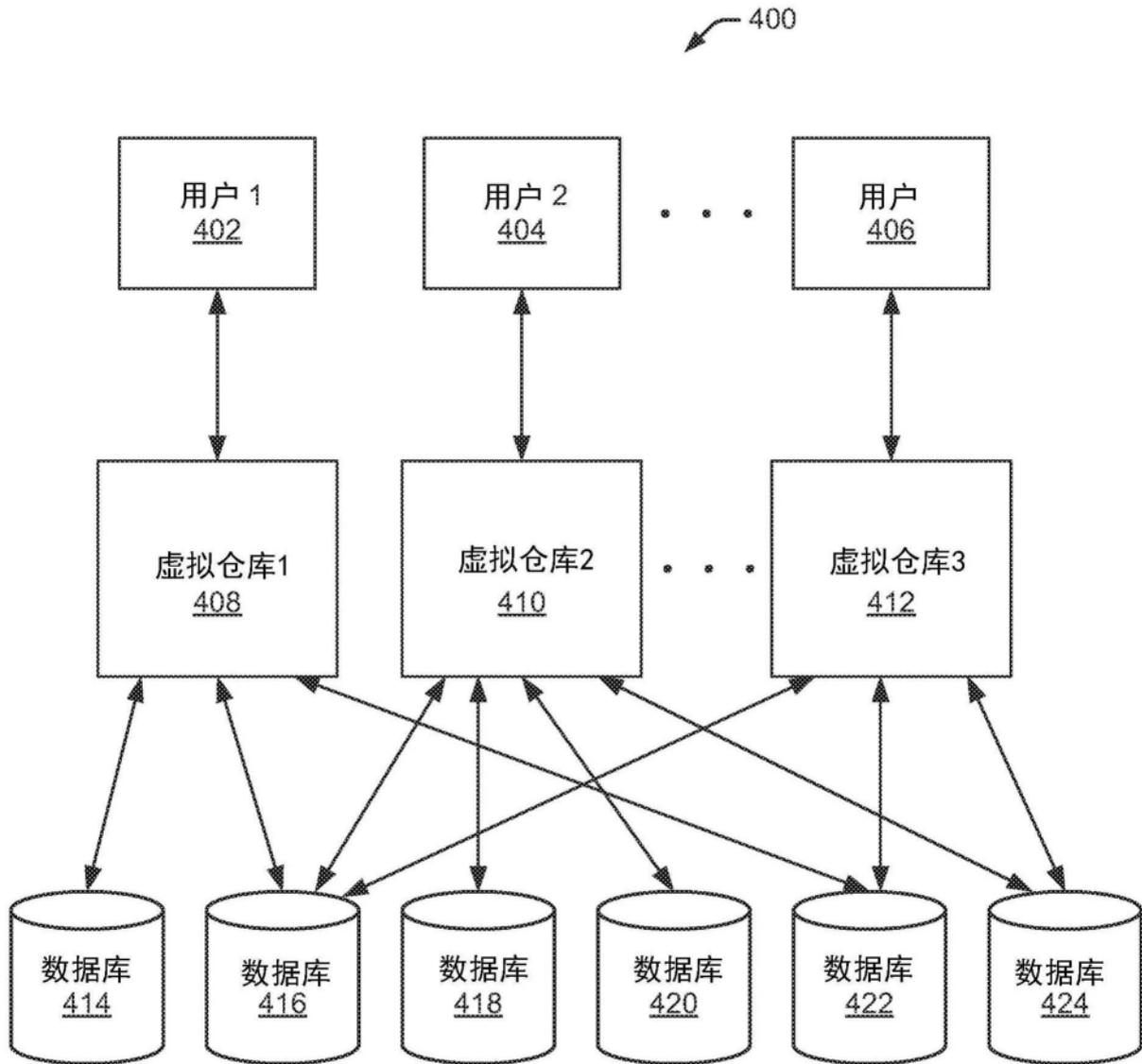


图4

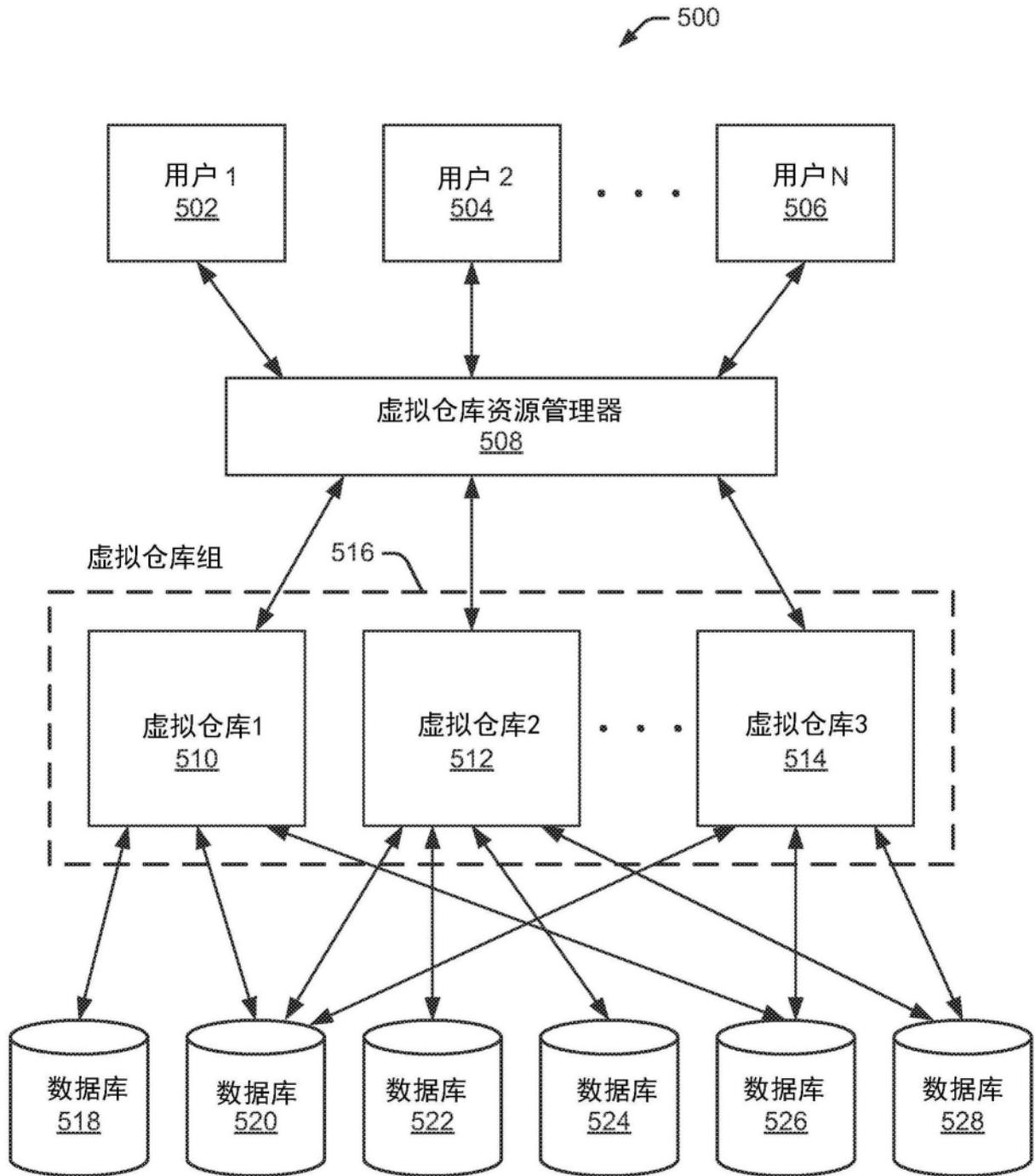


图5

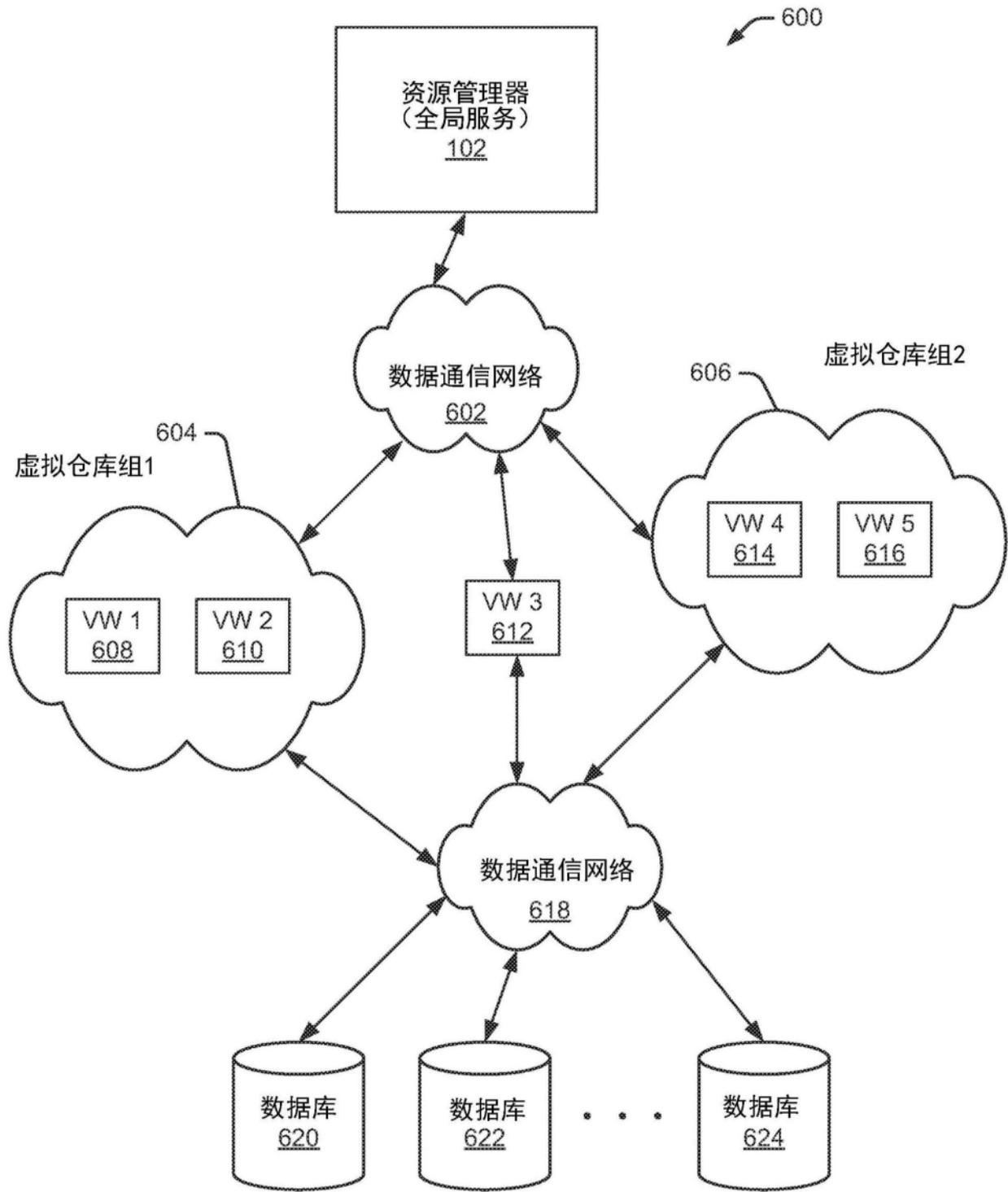


图6

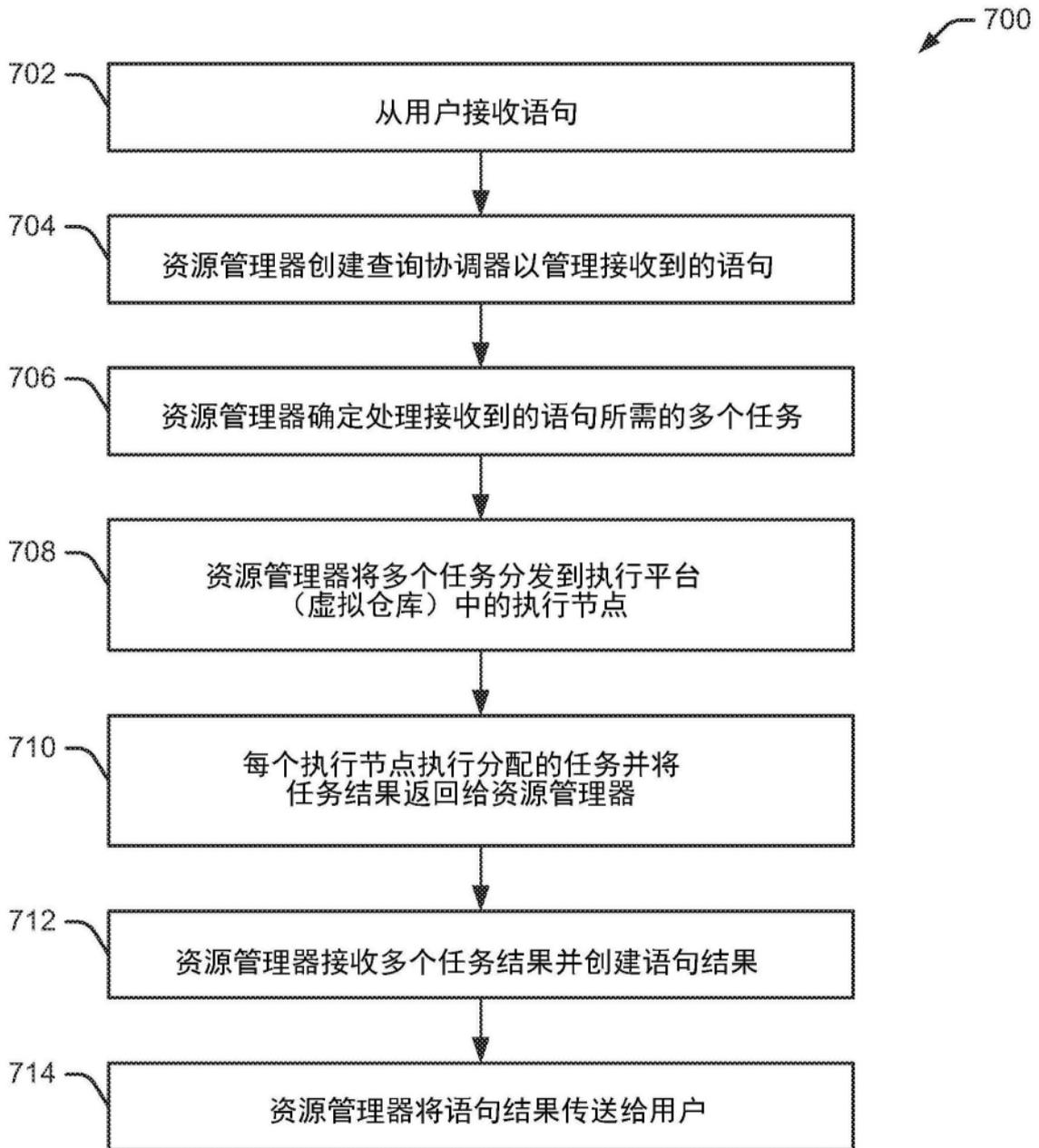


图7

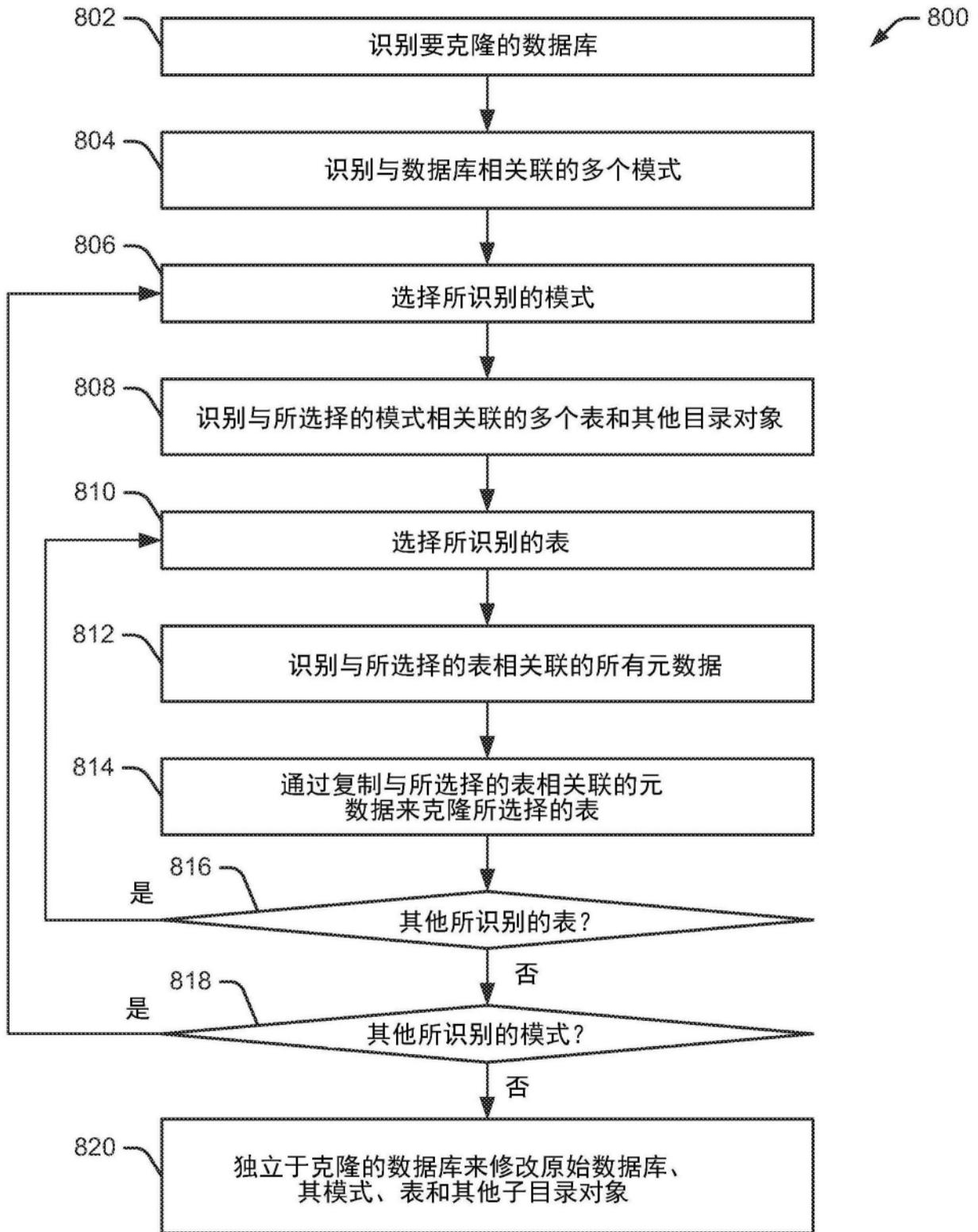


图8

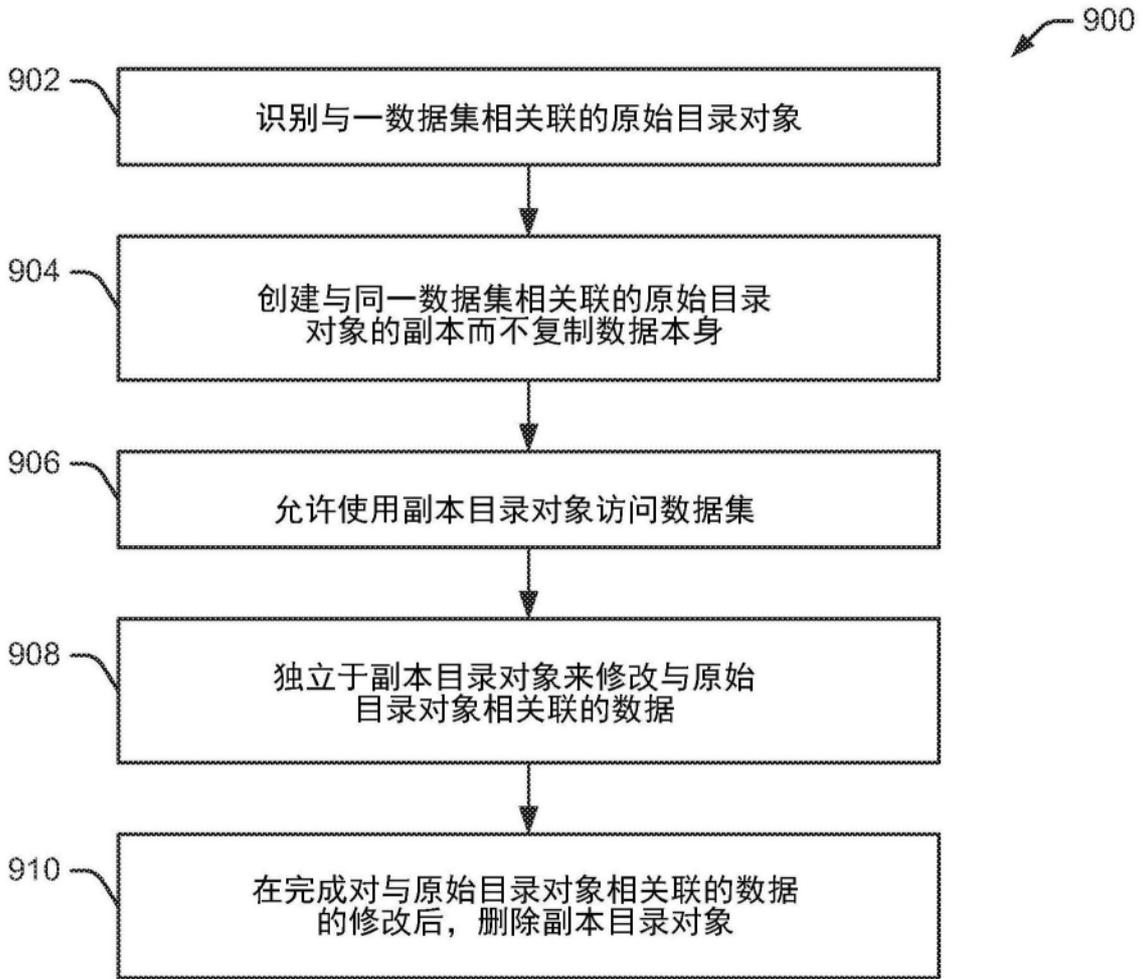


图9

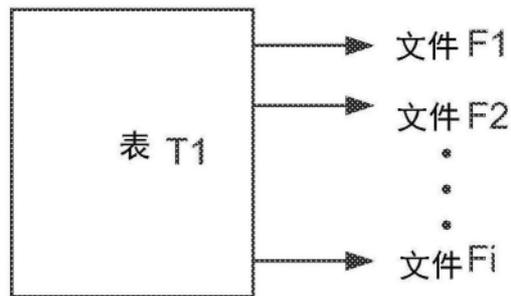


图10A

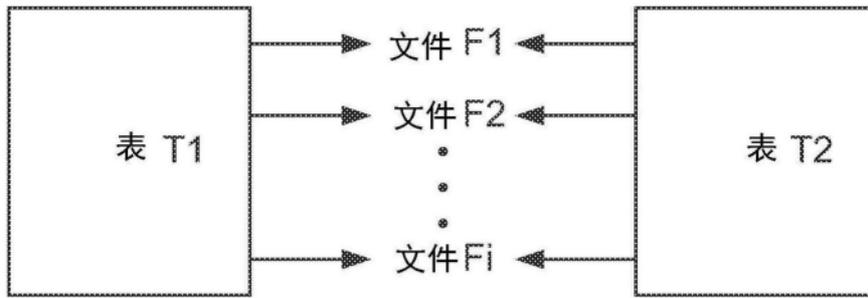


图10B

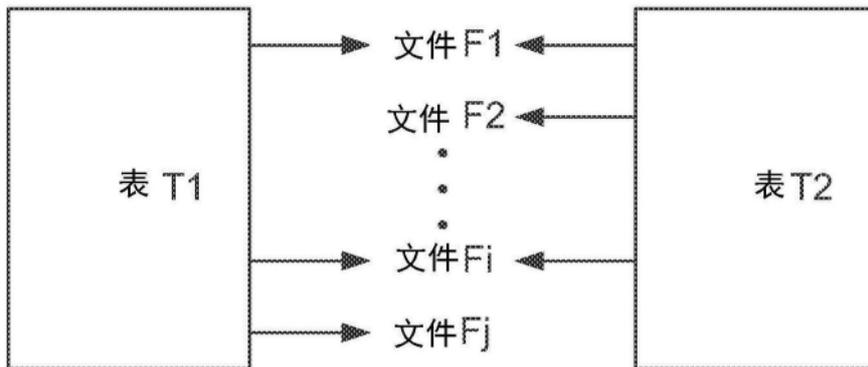


图10C

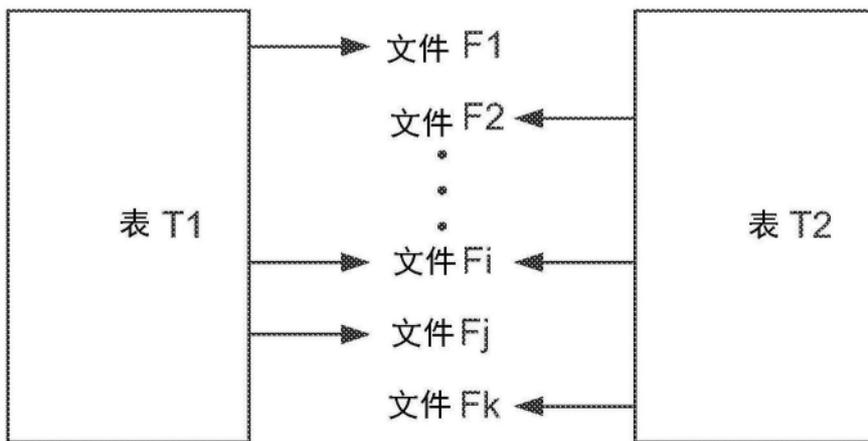


图10D

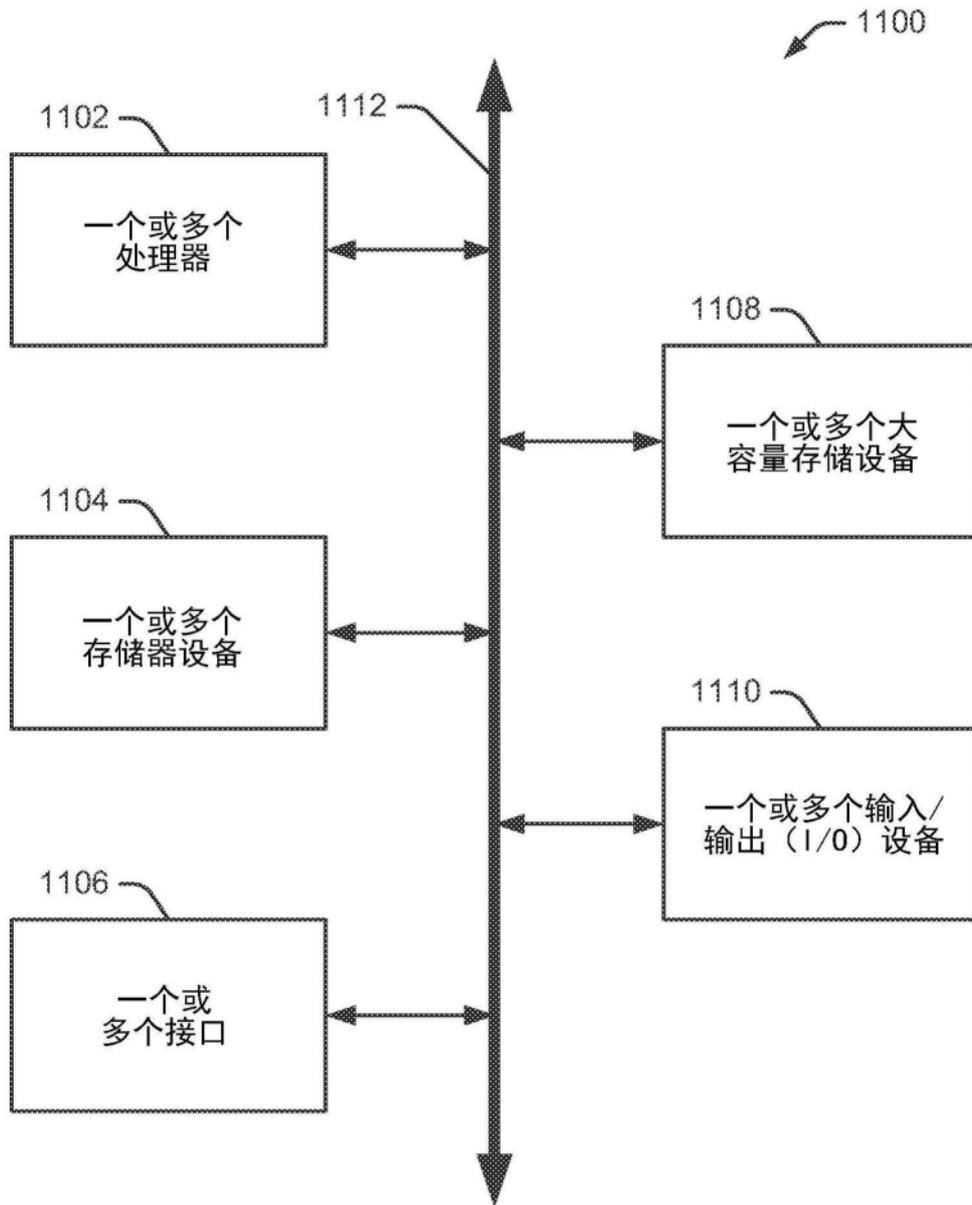


图11