



(12) 发明专利申请

(10) 申请公布号 CN 117461031 A

(43) 申请公布日 2024. 01. 26

(21) 申请号 202280040775.4

(22) 申请日 2022.06.06

(30) 优先权数据

63/197,803 2021.06.07 US

(85) PCT国际申请进入国家阶段日

2023.12.07

(86) PCT国际申请的申请数据

PCT/US2022/072771 2022.06.06

(87) PCT国际申请的公布数据

W02022/261616 EN 2022.12.15

(71) 申请人 斯诺弗雷克公司

地址 美国蒙大拿州

(72) 发明人 艾略特·布罗萨德

苏克鲁斯·科马尔拉苏库马尔

艾萨克·丘嫩 郭菊仪

乔纳森·李·梁 爱德华·马

斯库勒·詹姆斯·曼彻斯特

波利塔·保卢斯 绍林·沙阿

伊戈尔·津科夫斯基

(74) 专利代理机构 北京安信方达知识产权代理

有限公司 11262

专利代理师 俞立文 杨明钊

(51) Int.Cl.

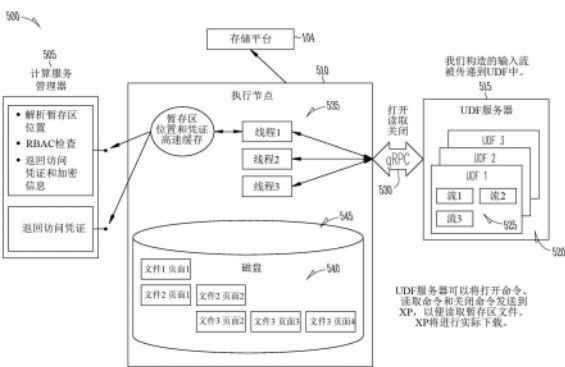
G06F 16/901 (2006.01)

(54) 发明名称

使用用户自定义函数访问文件

(57) 摘要

用户自定义函数 (UDF) 的文件访问系统可以在分布式数据库系统上实现。该系统可以存储可由其他用户调用的 UDF 签名和接口 (例如, 类、子类)。在 UDF 被调用时, 可以创建一个或多个接口对象 (例如, `InputStream`), 并且经由网络通道将请求传输到执行节点。执行节点可以实现多个被授权的线程, 并同时从暂存位置 (例如, 内部暂存区、外部暂存区) 下载文件数据。



权利要求书3页 说明书15页 附图8页

1. 一种方法,包括:

从分布式数据库的提供者数据库接收所述提供者数据库的多个用户自定义函数的多个用户自定义函数定义,所述多个用户自定义函数定义中的用户自定义函数定义包括用户自定义函数类的类型和将由所述分布式数据库上的请求者数据库执行的共享函数的函数名;

由所述分布式数据库接收用户自定义函数,所述用户自定义函数包括与使用所述用户自定义函数类的类型执行的至少一个操作相关的代码,所述用户自定义函数具有与所述多个用户自定义函数定义之一匹配的函数名;

接收对于执行所述用户自定义函数的请求,所述请求由所述分布式数据库的所述请求者数据库生成;

在所述请求者数据库的函数服务器上激活所述用户自定义函数类的类型的、对应于所请求的用户自定义函数的一个或更多个对象;

使用网络通道从所述函数服务器向所述请求者数据库的执行节点传输对文件的请求;

由所述请求者数据库的节点,使用到文件存储暂存区的路径检索在所述请求中指定的、对应于所述用户自定义类的类型的所述一个或更多个对象的文件;和

将所述文件存储在所述节点上。

2. 根据权利要求1所述的方法,还包括:

由所述请求者数据库的所述函数服务器,通过对存储在所述节点上的所述文件应用查询来生成结果数据,其中,通过所述节点中的网络通道将所述文件提供给所述函数服务器。

3. 根据权利要求1所述的方法,其中,所述文件存储暂存区对应于所述分布式数据库外部的的外部存储设备。

4. 根据权利要求1所述的方法,其中,所述用户自定义函数类的类型的所述一个或更多个对象在用户自定义函数服务器上被激活。

5. 根据权利要求1所述的方法,其中,所述网络通道是远程过程调用(RPC)通道。

6. 根据权利要求1所述的方法,还包括:

基于所述请求者数据库的用户先前被所述提供者数据库授予对所述文件之一中的每一个的共享访问,通过所述分布式数据库确定每个线程被授权访问所述文件之一。

7. 根据权利要求1所述的方法,还包括:

由所述请求者数据库,根据所述共享函数的代码文件中的一个或更多个指令来处理所述文件。

8. 根据权利要求7所述的方法,其中,所述代码文件是由所述共享函数执行的Java应用程序编程(JAR)文件。

9. 根据权利要求1所述的方法,其中,所述一个或更多个对象是Java虚拟机(JVM)。

10. 根据权利要求1所述的方法,其中,使用由所述请求者数据库生成的数据库查询来激活所述一个或更多个对象。

11. 根据权利要求10所述的方法,其中,所述数据库查询包括select语句,所述select语句包括所述共享函数的函数名。

12. 根据权利要求11所述的方法,其中,所述多个用户自定义函数中的每一个是由所述提供者数据库在所述分布式数据库上生成的可共享函数,其中,所述提供者数据库共享对

所述可共享函数的访问,以便在所述请求者数据库上执行。

13. 根据权利要求1所述的方法,其中,所述用户自定义函数类的类型是输入流类。

14. 一种系统,包括:机器的一个或多个处理器;以及存储器,所述存储器存储指令,所述指令当由所述一个或多个处理器执行时,使所述机器执行包括以下项的操作:

从分布式数据库的提供者数据库接收所述提供者数据库的多个用户自定义函数的多个用户自定义函数定义,所述多个用户自定义函数定义中的用户自定义函数定义包括用户自定义函数类的类型和将由所述分布式数据库上的请求者数据库执行的共享函数的函数名;

由所述分布式数据库接收用户自定义函数,所述用户自定义函数包括与使用所述用户自定义函数类的类型执行的至少一个操作相关的代码,所述用户自定义函数具有与所述多个用户自定义函数定义之一匹配的函数名;

接收对于执行所述用户自定义函数的请求,所述请求由所述分布式数据库的所述请求者数据库生成;

在所述请求者数据库的函数服务器上激活所述用户自定义函数类的类型的、对应于所请求的用户自定义函数的一个或多个对象;

使用网络通道从所述函数服务器向所述请求者数据库的执行节点传输对文件的请求;

由所述请求者数据库的节点,使用到文件存储暂存区的路径检索在所述请求中指定的、对应于所述用户自定义类的类型的所述一个或多个对象的文件;和

将所述文件存储在所述节点上。

15. 根据权利要求14所述的系统,还包括:

由所述请求者数据库的所述函数服务器,通过对存储在所述节点上的所述文件应用查询来生成结果数据,其中,通过所述节点中的网络通道将所述文件提供给所述函数服务器。

16. 根据权利要求14所述的系统,其中,所述文件存储暂存区对应于所述分布式数据库外部的存储设备。

17. 根据权利要求14所述的系统,其中,所述用户自定义函数类的类型的所述一个或多个对象在用户自定义函数服务器上被激活。

18. 根据权利要求14所述的系统,其中,所述网络通道是远程过程调用(RPC)通道。

19. 根据权利要求14所述的系统,所述操作还包括:

基于所述请求者数据库的用户先前被所述提供者数据库授予对所述文件之一中的每一个的共享访问,通过所述分布式数据库确定每个线程被授权访问所述文件之一。

20. 一种包含指令的机器存储介质,所述指令在被机器执行时使所述机器执行包括以下项的操作:

从分布式数据库的提供者数据库接收所述提供者数据库的多个用户自定义函数的多个用户自定义函数定义,所述多个用户自定义函数定义中的用户自定义函数定义包括用户自定义函数类的类型和将由所述分布式数据库上的请求者数据库执行的共享函数的函数名;

由所述分布式数据库接收用户自定义函数,所述用户自定义函数包括与使用所述用户自定义函数类的类型执行的至少一个操作相关的代码,所述用户自定义函数具有与所述多个用户自定义函数定义之一匹配的函数名;

接收对于执行所述用户自定义函数的请求,所述请求由所述分布式数据库的所述请求者数据库生成;

在所述请求者数据库的函数服务器上激活所述用户自定义函数类的类型的、对应于所请求的用户自定义函数的一个或更多个对象;

使用网络通道从所述函数服务器向所述请求者数据库的执行节点传输对文件的请求;

由所述请求者数据库的节点,使用到文件存储暂存区的路径检索在所述请求中指定的、对应于所述用户自定义类的类型的所述一个或更多个对象的文件;和

将所述文件存储在所述节点上。

使用用户自定义函数访问文件

[0001] 优先权声明

[0002] 本申请要求2021年6月7日提交的序列号为63/197,803的美国临时专利申请的优先权利益,该美国临时专利申请的全部内容通过引用并入本文。

技术领域

[0003] 本公开的实施例总体上涉及基于网络的数据库系统或云数据平台,且更具体地,涉及实现对数据库的用户自定义函数(user defined function)的文件访问。

[0004] 背景

[0005] 基于云的数据系统(诸如数据仓库)为用户提供了跟踪和管理大量数据的能力。用户可以实现处理数据的函数,例如用户自定义函数。然而,在数据库上实现用户自定义函数受到限制,并且扩展使用用户自定义函数以能够访问大文件(例如,1TB的文件)或大量文件(例如,1000000个图像文件)通常是不切实际的。

[0006] 附图简述

[0007] 从下面给出的详细描述和本公开的各种实施例的附图中,将更全面地理解本公开。

[0008] 图1示出了根据本公开一些实施例的示例计算环境,其包括与云存储平台通信的基于网络的数据仓库系统。

[0009] 图2是示出根据本公开一些实施例的计算服务管理器的组件的框图。

[0010] 图3是示出根据本公开一些实施例的执行平台的组件的框图。

[0011] 图4是根据本公开的一些实施例的计算环境,其概念性地示出了由运行在执行平台的给定执行节点上的进程执行用户自定义函数(UDF)的示例软件架构。

[0012] 图5根据本公开的一些实施例示出了用于在暂存位置(staging location)实现对UDF的文件访问的数据架构。

[0013] 图6A和图6B示出了根据本公开的一些实施例的用于实现针对用户自定义函数的文件访问的示例流程图。

[0014] 图7示出了根据本公开的一些实施例的计算机系统形式的机器的图解表示,在该计算机系统形式的机器内可以执行指令集,以用于使机器执行本文所讨论方法中的任何一种或更多种方法。

[0015] 详细描述

[0016] 现在将详细参考用于实现本发明主题的具体示例实施例。这些具体实施例的示例在附图中示出,并且具体细节在以下描述中阐述,以便提供对主题的透彻理解。应当理解,这些示例并不旨在将权利要求的范围限制于所示的实施例。相反,它们旨在覆盖可能包含在本公开范围内的替代、修改和等同物。

[0017] 如前所述,用可扩展的方法实现可以访问大文件的数据库用户自定义函数(UDF)可能比较困难。为此,函数文件访问系统可以存储一个或多个UDF签名,其指定函数名并且具有到暂存区(stage)(例如,内部存储装置、外部存储装置)上的文件的路径。UDF的指令

可以被上传并被存储在数据库系统上(例如,对于Java UDF,Java代码被存储)。可以在用户之间、在分布式数据库中共享和执行UDF。例如,提供者账户用户可以创建UDF并共享对该UDF的访问,以使消费者账户用户能够调用UDF。当消费者账户用户调用UDF时,系统确定字符串对应于UDF接口(例如,Java `InputStream`),该UDF接口具有指向暂存区文件(stage file)的指针(例如,在UDF签名中指定了暂存区名称,以及其他参数,例如在流中的位置(例如,“0”、“1000”)。执行节点可以通过网络通道接收UDF请求,并验证用户有权访问文件以及是否将凭证传输到执行节点的线程。然后,执行节点的线程可以检索文件内容并将其作为页面(page)缓存在执行节点的本地存储器中。在一些示例实施例中,页面被预先缓存使得稍后被请求的页面已经被存储在执行节点上。在一些示例实施例中,系统230执行实时检索,并且页面被预先缓存,但是仅在被给定UDF(例如,给定UDF中的一个或多个流)请求时才被下载。

[0018] 图1示出了根据本公开的一些实施例的示例计算环境100,其包括基于网络的数据仓库系统102的示例形式的数据库系统。为了避免用不必要的细节模糊本发明主题,图1中省略了与传达对本发明主题的理解不密切相关的各种功能组件。然而,本领域技术人员将容易认识到,各种附加功能组件可以被包括作为计算环境100的一部分,以促进本文没有具体描述的附加功能。在其他实施例中,计算环境可以包括另一种类型的基于网络的数据库系统或云数据平台。

[0019] 如图所示,计算环境100包括与云存储平台104(例如,AWS[®]、Microsoft Azure Blob Storage[®]或谷歌云存储(Google Cloud Storage))通信的基于网络的数据仓库系统102。基于网络的数据仓库系统102是基于网络的系统,用于报告和分析来自包括云存储平台104内的一个或多个存储位置的一个或多个不同源的集成数据。云存储平台104包括多个计算机器,并向基于网络的数据仓库系统102按需提供计算机系统资源,如数据存储和计算能力。

[0020] 基于网络的数据仓库系统102包括计算服务管理器108、执行平台110和一个或多个元数据数据库112。基于网络的数据仓库系统102托管并向多个客户端账户提供数据报告和分析服务。

[0021] 计算服务管理器108协调和管理基于网络的数据仓库系统102的操作。计算服务管理器108还执行查询优化和编译,以及管理提供计算资源的计算服务的集群(也称为“虚拟仓库”)。计算服务管理器108可以支持任意数量的客户端账户,如提供数据存储和检索请求的终端用户、管理本文描述的系统和方法的系统管理员以及与计算服务管理器108交互的其他组件/设备。

[0022] 计算服务管理器108还与客户端设备114通信。客户端设备114对应于由基于网络的数据仓库系统102支持的多个客户端账户之一的用户。用户可以利用客户端设备114向计算服务管理器108提交数据存储、检索和分析请求。

[0023] 计算服务管理器108还耦合到一个或多个元数据数据库112,元数据数据库112存储与基于网络的数据仓库系统102及其用户相关联的各种函数和方面相关的元数据。例如,元数据数据库112可以包括存储在远程数据存储系统中的数据以及可从本地高速缓存获得的数据的概要。另外,元数据数据库112可以包括关于数据如何在远程数据存储系统(例如,云存储平台104)和本地高速缓存中组织的信息。由元数据数据库112存储的信息允

许系统和服务确定是否需要在不从存储设备加载或访问实际数据的情况下访问一段数据。

[0024] 计算服务管理器108还耦合到执行平台110,执行平台110提供执行各种数据存储和数据检索任务的多个计算资源。执行平台110耦合到云存储平台104的存储平台104。存储平台104包括多个数据存储设备120-1至120-N。在一些实施例中,数据存储设备120-1至120-N是位于一个或更多个地理位置的基于云的存储设备。例如,数据存储设备120-1至120-N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120-1至120-N可以是硬盘驱动器(HDD)、固态驱动器(SSD)、存储集群、亚马逊(Amazon)S3™存储系统或任何其他数据存储技术。另外,云存储平台104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。

[0025] 执行平台110包括多个计算节点。计算节点上的进程集合执行由计算服务管理器108编译的查询计划。进程集合可以包括:执行查询计划的第一进程;第二进程,其使用最近最少使用(least recently used,LRU)策略来监视和删除缓存文件,并实现内存不足(OOM)错误缓解进程;从进程日志和状态提取健康信息以发送回计算服务管理器108的第三进程;在系统引导(boot)之后建立与计算服务管理器108的通信的第四进程;以及第五进程,该第五进程处理与由计算服务管理器108提供的给定作业的计算集群的所有通信,并将信息传送给计算服务管理器108和执行平台110的其他计算节点。

[0026] 在一些实施例中,计算环境100的元件之间的通信链路经由一个或更多个数据通信网络来实现。这些数据通信网络可以利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合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。

[0027] 计算服务管理器108、元数据数据库112、执行平台110和存储平台104在图1中示为单个的分立组件。然而,计算服务管理器108、元数据数据库112、执行平台110和存储平台104中的每一个都可以被实现为分布式系统(例如,分布在多个地理位置处的多个系统/平台上)。另外,计算服务管理器108、元数据数据库112、执行平台110和存储平台104中的每一个可以根据所接收请求的变化和基于网络的数据仓库系统102的变化需求而(彼此独立地)使其规模放大或缩小。因此,在所描述的实施例中,基于网络的数据仓库系统102是动态的,并且支持频繁变化以满足当前的数据处理需求。

[0028] 在典型操作期间,基于网络的数据仓库系统102处理由计算服务管理器108确定的多个作业。这些作业由计算服务管理器108调度和管理,以确定何时以及如何执行该作业。例如,计算服务管理器108可以将作业划分为多个离散任务,并且可以确定需要什么数据来执行多个离散任务中的每个离散任务。计算服务管理器108可以将多个离散任务中的每个离散任务分配给执行平台110的一个或更多个节点以处理任务。计算服务管理器108可以确定需要哪些数据来处理任务,并且进一步确定执行平台110内的哪些节点最适合于处理该任务。一些节点可能已经高速缓存了处理任务所需的数据,因此是用于处理任务的不错的候选者。存储在元数据数据库112中的元数据帮助计算服务管理器108确定执行平台110中的哪些节点已经高速缓存了处理该任务所需的至少一部分数据。执行平台110中的一个或更多个节点使用由节点高速缓存的数据以及必要时从云存储平台104检索到的数据来处理任务。期望从执行平台110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云存储平台104中检索数据快得多。

[0029] 如图1所示,计算环境100将执行平台110与存储平台104分开。在这种布置中,执行平台110中的处理资源和高速缓存资源独立于云存储平台104中的数据存储设备120-1至120-N进行操作。因此,计算资源和高速缓存资源不限于具体的数据存储设备120-1至120-N。相反,所有计算资源和所有高速缓存资源可以从云存储平台104中的任何数据存储资源检索数据,以及将数据存储到其中。

[0030] 图2是示出根据本公开一些实施例的计算服务管理器108的组件的框图。如图2所示,计算服务管理器108包括耦合到访问元数据数据库206的访问管理器202和凭证管理系统204,访问元数据数据库206是元数据数据库112的示例。访问管理器202处理针对本文描述的系统的认证和授权任务。凭证管理系统204便于使用远程存储的凭证来访问外部资源,例如远程存储设备中的数据资源。如本文所用,远程存储设备也可以被称为“永久存储设备”或“共享存储设备”。例如,凭证管理系统204可以(例如,在访问元数据数据库206中)创建和维护远程凭证存储器定义和凭证对象。远程凭证存储器定义标识远程凭证存储器,并包括从远程凭证存储器访问安全凭证的访问信息。凭证对象使用非敏感信息(例如,文本字符串)来标识一个或多个安全凭证,这些信息将从远程凭证存储器中检索以用于访问外部资源。当在运行时接收到调用外部资源的请求时,凭证管理系统204和访问管理器202使用存储在访问元数据数据库206中的信息(例如,凭证对象和凭证存储器定义)来从远程凭证存储器中检索用于访问外部资源的安全凭证。

[0031] 请求处理服务208管理接收到的数据存储请求和数据检索请求(例如,将对数据库数据执行的作业)。例如,请求处理服务208可以确定用于处理接收到的查询的数据(例如,数据存储请求或数据检索请求)。数据可以存储在执行平台110内的高速缓存中,或者存储在存储平台104中的数据存储设备中。

[0032] 管理控制台服务210支持管理员和其他系统管理员对各种系统和进程的访问。另外,管理控制台服务210可以接收对于执行作业并监视系统上的工作负载的请求。

[0033] 计算服务管理器108还包括作业编译器212、作业优化器214和作业执行器216。作业编译器212将作业解析为多个离散任务,并为多个离散任务中的每个离散任务生成执行代码。作业优化器214基于需要处理的数据来确定执行多个离散任务的最好方法。作业优化器214还处理各种数据修剪操作和其他数据优化技术,以提高执行作业的速度和效率。作业执行器216执行从队列接收的或由计算服务管理器108确定的作业的执行代码。

[0034] 作业调度器和协调器218将接收到的作业发送到适当的服务或系统,以进行编译、优化并分派到执行平台110。例如,作业可以进行优先级排序然后按优先级顺序进行处理。在实施例中,作业调度器和协调器218确定由计算服务管理器108与其他“外部”作业(诸如可以由数据库中的其他系统调度但可以利用执行平台110中的相同处理资源的用户查询)调度的内部作业的优先级。在一些实施例中,作业调度器和协调器218识别或分配执行平台110中的特定节点以处理特定任务。虚拟仓库管理器220管理在执行平台110中实现的多个虚拟仓库的操作。例如,虚拟仓库管理器220可以生成用于执行接收到的查询的查询计划。函数文件访问系统230被配置成使用指向暂存区的UDF和用户自定义函数签名来管理对暂存区(例如,内部暂存区或外部暂存区)中的文件的文件访问,如下文进一步详细讨论的。

[0035] 另外,计算服务管理器108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缓冲器(例如,执行平台110中的缓冲器)中的数据相关的信息。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来检索用于处理特定任务或作业的数据。监视器和工作负载分析器224监管由计算服务管理器108执行的进程,并管理任务(例如,工作负载)在执行平台110中的执行节点和虚拟仓库之间的分配。监视器和工作负载分析器224还根据需要,基于整个基于网络的数据仓库系统102中不断变化的工作负载来重新分配任务,并且还可以基于也可以由执行平台110处理的用户(例如,“外部”)查询工作负载来重新分配任务。配置和元数据管理器222以及监视器和工作负载分析器224耦合到数据存储设备226。图2中的数据存储设备226表示基于网络的数据仓库系统102中的任何数据存储设备。例如,数据存储设备226可以表示执行平台110中的缓冲器、存储平台104中的存储设备或任何其他存储设备。

[0036] 如本文实施例中所述,计算服务管理器108验证来自执行平台(例如,执行平台110)的所有通信,以验证该通信的内容和上下文与已知分配给执行平台的任务一致。例如,执行查询A的执行平台的实例不应被允许请求访问与查询A无关的数据源D(例如,数据存储设备226)。类似地,给定的执行节点(例如,执行节点302-1)可能需要与另一个执行节点(例如,执行节点302-2)通信,并且应该被禁止与第三执行节点(例如,执行节点312-1)通信,并且可以(例如,在日志或其他位置中)记录任何这样的非法通信。此外,存储在给定执行节点上的信息仅限于与当前查询相关的数据,任何其他数据都不可用,在密钥不可用的情况下,通过销毁或加密来实现。

[0037] 图3是示出根据本公开一些实施例的执行平台110的组件的框图。如图3所示,执行平台110包括多个虚拟仓库,包括虚拟仓库1、虚拟仓库2和虚拟仓库n。每个虚拟仓库都包括多个执行节点,每个执行节点都包括数据高速缓存和处理器。虚拟仓库可以通过使用多个执行节点并行执行多个任务。如本文所讨论的,执行平台110可以基于系统和用户的当前处理需求实时添加新的虚拟仓库和删除现有的虚拟仓库。这种灵活性允许执行平台110在需要时快速部署大量计算资源,而不必在不再需要它们时继续为那些计算资源付费。所有虚拟仓库都可以访问任何数据存储设备(例如,云存储平台104中的任何存储设备)中的数据。

[0038] 尽管图3中所示的每个虚拟仓库包括三个执行节点,但是特定的虚拟仓库可以包括任意数量的执行节点。此外,虚拟仓库中执行节点的数量是动态的,使得在存在额外需求时创建新的执行节点,并在不再需要现有执行节点时将其删除。

[0039] 每个虚拟仓库能够访问图1中所示的数据存储设备120-1至120-N中的任何一个。因此,虚拟仓库不必被分配给特定的数据存储设备120-1至120-N,而是可以从云存储平台104内的数据存储设备120-1至120-N中的任何一个访问数据。类似地,图3中所示的每个执行节点可以从数据存储设备120-1至120-N中的任何一个访问数据。在一些实施例中,特定的虚拟仓库或特定的执行节点可以被临时分配给具体的数据存储设备,但是该虚拟仓库或执行节点后续可以从任何其他数据存储设备访问数据。

[0040] 在图3的示例中,虚拟仓库1包括三个执行节点302-1、302-2和302-N。执行节点302-1包括高速缓存304-1和处理器306-1。执行节点302-2包括高速缓存304-2和处理器306-2。执行节点302-N包括高速缓存304-N和处理器306-N。每个执行节点302-1、302-2和302-N与处理一个或更多个数据存储和/或数据检索任务相关联。例如,虚拟仓库可以处理与内部服务(诸如,聚类服务、实例化视图刷新服务、文件压缩服务、存储程序服务或文件升级服务)相关联的数据存储和数据检索任务。在其他实施方式中,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。

[0041] 类似于上面讨论的虚拟仓库1,虚拟仓库2包括三个执行节点312-1、312-2和312-N。执行节点312-1包括高速缓存314-1和处理器316-1。执行节点312-2包括高速缓存314-2和处理器316-2。执行节点312-N包括高速缓存314-N和处理器316-N。另外,虚拟仓库3包括三个执行节点322-1、322-2和322-N。执行节点322-1包括高速缓存324-1和处理器326-1。执行节点322-2包括高速缓存324-2和处理器326-2。执行节点322-N包括高速缓存324-N和处理器326-N。

[0042] 在一些实施例中,图3所示的执行节点相对于由执行节点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有关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,可以用另一个节点透明地替换发生故障的节点。由于没有与发生故障的执行节点相关联的状态信息,因此新的(替换)执行节点可以轻松替换发生故障的节点,而无需考虑重新创建特定状态。

[0043] 尽管图3中所示的执行节点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意数量的高速缓存的执行节点。另外,高速缓存的大小在不同执行节点之间可能有所不同。图3中所示的高速缓存在本地执行节点中存储从云存储平台104中的一个或多个数据存储设备检索到的数据。因此,高速缓存减小或消除了从远程存储系统中不断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的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云存储平台104中的任何存储设备的数据。

[0044] 此外,高速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,一个执行节点可包含大量的计算资源和最少的高速缓存资源,从而使该执行节点可用于需要大量计算资源的任务。另一个执行节点可包含大量的高速缓存资源和最少的计算资源,从而使该执行节点可用于需要高速缓存大量数据的任务。又一个执行节点可包含提供更快输入-输出操作的高速缓存资源,这对于需要快速扫描大量数据的任务很有用。在一些实施例中,基于执行节点将要执行的预期任务,在创建执行节点时确定与特定执行节点相关联的高速缓存资源和计算资源。

[0045] 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改变的任务而随时间改变。例如,如果执行节点执行的任务变得更加处理器密集,则可以认为执行节点分配更多的处理资源。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,则可以为执行节点分配更多的高速缓存资源。

[0046] 尽管虚拟仓库1、2和n与同一执行平台110相关联,但是可以使用在多个地理位置处的多个计算系统来实现虚拟仓库。例如,虚拟仓库1可以由第一地理位置处的计算系统来实现,而虚拟仓库2和n由第二地理位置处的另一计算系统来实现。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的实体维护的基于云的计算机系统。

[0047] 另外,每个虚拟仓库在图3中被示为具有多个执行节点。可以使用位于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1的实例在一个地理位置处的一个计算平台上实现执行节点302-1和302-2,而在另一个地理位置

处的不同计算平台处实现执行节点302-N。选择特定的计算系统以实施执行节点可以取决于各种因素,诸如特定的执行节点所需的资源水平(例如,处理资源要求和高速缓存要求)、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在虚拟仓库中实现其他执行节点。

[0048] 执行平台110也是容错的。例如,如果一个虚拟仓库发生故障,那么该虚拟仓库将很快被位于不同地理位置处的不同虚拟仓库替换。

[0049] 特定执行平台110可以包括任意数量的虚拟仓库。另外,特定执行平台中虚拟仓库的数量是动态的,使得在需要附加的处理和/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关联的资源时,可以删除现有的虚拟仓库。

[0050] 在一些实施例中,虚拟仓库可以在云存储平台104中的相同数据上操作,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。此配置允许对不同虚拟仓库上的请求进行独立处理,并且请求之间不会相互干扰。这种独立的处理与动态添加和移除虚拟仓库的能力相结合,支持为新用户添加新的处理能力,而不会影响现有用户观察到的性能。

[0051] 图4是根据本公开的一些实施例的计算环境400,其概念性地示出了由运行在执行平台110的给定执行节点上的进程执行用户自定义函数(UDF)的示例软件架构。

[0052] 如图所示,来自执行平台110的执行节点302-1包括执行节点进程410,在一个实施例中,执行节点进程410在处理器306-1上运行,并且还可以利用来自高速缓存304-1(或另一个存储器设备或存储装置)的存储器。如本文所述,“进程”或“计算进程”可以指由执行节点或执行平台的一个或更多个线程执行的计算机程序的实例。

[0053] 如前所述,计算服务管理器108验证来自执行平台110的所有通信,以验证该通信的内容和上下文与已知分配给执行平台110的任务一致。例如,执行查询A的执行平台110不被允许请求访问与查询A无关的特定数据源(例如,数据存储设备226或云存储平台104中的任何一个存储设备)。在一个示例中,执行节点302-1可能需要与第二执行节点(例如,执行节点302-2)通信,但是本文描述的安全机制可以不允许与第三执行节点(例如,执行节点312-1)通信。此外,可以(例如,在日志444中或在其他位置)记录任何这样的非法通信。此外,存储在给定执行节点上的信息被限制为与当前查询相关的数据,并且在密钥不可用的情况下,通过销毁或加密,任何其他数据是不能使用的。

[0054] 在图4的示例中,执行节点进程410正在执行UDF客户端412。在一个实施例中,UDF客户端412被实现为支持用诸如JAVA等特定编程语言编写的UDF。在一个实施例中,UDF客户端412以与用户代码430不同的编程语言(例如,C或C++)实现,这可以通过使用不同的代码库(例如,没有相同或更少潜在安全漏洞的代码库)来进一步提高计算环境400的安全性。

[0055] 用户代码430可以作为包被提供,例如以JAR(JAVA归档)文件的形式,其包括一个或更多个UDF的代码。在一个实施例中,服务器实现代码432是JAR文件,其启动负责从执行节点进程410接收请求、分配工作线程来执行用户代码、返回结果以及其他类型的服务器任务的服务器。

[0056] 在一个实现中,来自UDF(例如,基于JAVA的UDF)的操作可以由在沙箱进程420(例如,UDF服务器)内执行的用户代码运行时(user code runtime)424来执行。在一个实施例中,用户代码运行时424被实现为虚拟机,例如JAVA虚拟机(JVM)。由于用户代码运行时424

有利地在相对于执行节点进程410的单独进程中执行,因此操纵执行节点进程410的风险较低。除了其他类型的信息或消息之外,执行操作的结果可以存储在日志444中以供查看和检索。在一个实施例中,日志444可以本地存储在执行节点302-1处的存储器中,或者存储在诸如存储平台104的单独位置。此外,可以利用进一步提供了数据传输的认证和加密的高性能协议(例如,没有数据的串行化(serialization)或去串行化,没有存储器副本;对记录批次进行操作,而不必访问单独的列、记录或单元格;利用有效的远程过程调用技术和用于数据传输的网络协议)进行数据传输(例如,分布式数据集),来将这样的结果从用户代码运行时424返回给UDF客户端412。在一个实施例中,UDF客户端412使用支持用户代码运行时424之间的柱状数据(columnar data)的网络传输的数据传送机制(反之亦然),具有上述优点。

[0057] 在一个实施例中,沙箱进程420是执行节点进程410的子进程(或单独的进程)。在一个实施例中,子进程(sub-process)是指给定父进程(例如,在该示例中,执行节点进程410)的子进程(child process)。在一个示例中,沙箱进程420是通过使用诸如名称空间和安全计算模式的安全机制(例如,对正在执行的进程及其所有后代使用系统调用过滤器,从而减少对给定操作系统的内核的攻击面)来限制不可信应用的运行环境来降低安全漏洞风险的程序。此外,在一个示例中,与执行节点进程410相比,沙箱进程420是轻量级进程,并且被优化(例如,紧密耦合到给定操作系统内核的安全机制)以在沙箱环境中以安全的方式处理数据库查询。

[0058] 在一个实施例中,沙箱进程420可以利用虚拟网络连接来与主题系统内的其他组件通信。可以针对主题系统的其他组件为虚拟网络连接配置一组特定的规则。例如,可以为特定UDF配置虚拟网络连接的这种规则,以限制由UDF执行的操作可访问的位置(例如,互联网上的特定站点或UDF可以通信的组件)。因此,在这个示例中,UDF可以被拒绝访问互联网上的特定网络位置或站点。

[0059] 沙箱进程420可以被理解为向沙箱内的进程(或多个进程)提供受约束的计算环境,其中这些受约束的进程可以被控制和限制以限制对某些计算资源的访问。

[0060] 安全机制的示例可包括:命名空间的实现,其中在沙箱环境中执行的每个相应的进程组可访问相应的计算资源(例如,进程ID、主机名、用户ID、文件名、与网络访问相关的名称、和进程间通信),该计算资源不可由另一进程组访问(可访问前一进程组不可访问的不同资源组);其他容器实现等。通过使沙箱进程420作为执行节点进程410的子进程来执行,在一些实施例中,与可以单独利用虚拟机解决方案的其他技术相比,处理给定数据库查询的等待时间可以被显著减少(例如,在一些情况下,等待时间减少为10分之一)。

[0061] 如进一步所示,沙箱进程420可以利用沙箱策略440来实施给定的安全策略。沙箱策略440可以是具有与沙箱进程420的配置相关的信息以及关于访问和利用系统资源的限制(如果有的话)和权限的细节的文件。示例限制可以包括对网络访问或文件系统访问的限制(例如,重新映射文件系统以将文件放置在可能不可访问的不同位置,其他文件可以安装在不同位置,等等)。沙箱进程420限制用户代码运行时424的存储器和处理器(例如CPU)的使用,确保同一执行节点上的其他操作可以执行而不会耗尽资源。

[0062] 如上所述,沙箱进程420是执行节点进程410的子进程(或独立进程),这实际上意味着沙箱进程420驻留在与执行节点进程410不同的独立存储器空间中。在与沙箱进程420相关的安全漏洞(例如,来自给定UDF的错误或恶意代码)发生时,如果任意存储器被恶意行

为者访问,则由执行节点进程存储的数据或信息受到保护。

[0063] 尽管图4的以上讨论描述了使用JAVA (例如,面向对象编程语言) 实现的组件,但是应当理解,计算环境400支持其他编程语言 (例如,解释型编程语言)。在一个实施例中,支持PYTHON在计算环境400中实现和执行UDF。在该示例中,用户代码运行时424可以用PYTHON解释器代替,用于在沙箱进程420内执行来自UDF (例如,用PYTHON编写的) 的操作。

[0064] 图5根据一些示例实施例示出了用于在暂存位置实现对UDF的文件访问的数据架构500。UDF服务器515 (例如,运行给定执行节点的XP进程的子进程) 可以管理多个UDF 520 (例如,UDF1、UDF2、UD3),每个UDF 520可以使用UDF签名或定义来定义,下面在图6中进一步详细讨论。根据一些示例实施例,每个UDF管理一个或更多个流525 (例如,Java InputStream),其中流的数量可以被指定为由定义UDF签名的终端用户所指定的函数或应用的参数。例如,如果给定UDF是用于比较两个不同文件 (例如,文件页面2 (File Page2); 和文件3页面4 (File 3page 4)) 的差异比较函数,则可以创建两个流 (例如,流1和流2) 来处理这两个文件 (例如,经由线程)。

[0065] UDF服务器515可以经由诸如RPC (例如,gRPC、Google远程过程调用、其他RPC客户端) 的网络接口通道530来打开、读取或关闭对执行节点510 (例如,执行节点302-1) 的请求。在执行节点510中,多个线程535服务于从网络接口通道530接收的请求 (例如,以执行所请求数据的并发处理)。在一些示例实施例中,每个线程通过计算服务管理器接入点来访问计算服务管理器505 (例如,计算服务管理器108),以确认用户 (例如,调用UDF的用户) 对于由线程处理的给定文件具有访问权限。如果线程具有访问权限,则计算服务管理器505将凭证和访问信息传递给线程,并且线程访问存储平台104 (例如,内部暂存区、外部对象存储装置、外部表等) 并下载文件,这些文件作为页面540存储在执行节点510的本地存储器545 (例如,磁盘) 上。

[0066] 图6A示出了根据一些示例环境的用于在分布式数据库上实现对用户自定义函数的文件访问的示例方法600的流程图。在操作605,函数文件访问系统230生成一个或更多个用户以查找函数定义。例如,终端用户对UDF签名进行定义,如下所示:

[0067] :::::::::::::::代码开始:::::::::::::

[0068] public static int Func_123(InputStream s1,int pos1,InputStream s2,int pos2)

[0069] :::::::::::::::代码结束:::::::::::::

[0070] 在操作610,函数文件访问系统230存储与操作605的定义相对应的一个或更多个用户自定义函数 (UDF)。可以在操作610上传和存储的示例UDF (例如,Java程序) 包括:

.....代码开始.....

```
import java.io.*;
```

```
import com.berryworks.edireader.json.fromedi. EditoJson;
```

```
public class Func_123{
```

```
    public static String Parse(InputStream in)
```

```
    {
```

```
        final EditoJson editoJson = new EditoJson();
```

```
        editoJson.setFormatting(true); //格式
```

```
        editoJson.setAnnotated(false); //注释
```

```
        editoJson.setSummarize(false); //汇总
```

[0071]

```
        StringWriter sw = new StringWriter();
```

```
        try (Reader reader = new BufferedReader(new InputStreamReader(in, "UTF-8"));
```

```
            Writer writer = new BufferedWriter(sw)) {
```

```
            editoJson.asJson(reader, writer);
```

```
        } catch (Exception e) {
```

```
            throw new RuntimeException(e.getMessage());
```

```
        }
```

```
        return sw.toString();
```

```
    }
```

[0072] }

.....代码结束.....

[0073] 在操作615,函数文件访问系统230接收对UDF之一的调用。例如,有权调用函数的终端用户输入SQL来调用函数:

[0074] :::::::::::::::代码开始:::::::::::::

[0075] SELECT Func_123('@stage1/path/file1' ,0, '@stage2/path/file2' ,1000);

[0076] :::::::::::::::代码结束:::::::::::::

[0077] 在操作620,文件访问系统230的函数处理对UDF的请求,如下面参考图6B进一步详细讨论的。例如,在运行时,系统检查函数签名,并将字符串转换为UDF接口(例如,“InputStream”),其中字符串输入参数必须包含有效的暂存区文件句柄(例如,@stage1)。尽管InputStream在这里作为示例UDF Java接口进行讨论,但是应该理解,该系统也具有其他用户自定义函数接口,例如Java文件、RandomAccessFile等。此外,尽管这里讨论Java是在UDF中实现的示例语言,但是应当理解,根据一些示例实施例,函数文件访问系统230实现其他语言(例如,PYTHON)和给定语言的其他函数接口。

[0078] 图6B示出了根据一些示例实施例的处理对UDF的请求的示例方法650的流程图。

[0079] 在操作655,UDF服务器515生成对节点的请求。例如,UDF服务器515根据所请求的UDF打开一个或多个流,并且UDF服务器515经由通道530将来自流(例如,InputStreams)的一个或多个打开请求发送到执行节点510。

[0080] 在操作660,执行节点510接收请求。例如,线程535从通道530接收打开请求。

[0081] 在操作665,计算机服务管理器505授权请求。例如,线程535的线程1请求对访问存储平台104中的文件的授权,并且计算机服务管理器505授权线程的访问(例如,用户的访问)并给予线程访问数据(例如,凭证)以访问存储平台104中的文件。

[0082] 在操作670,执行节点510从暂存区检索文件数据。例如,每个线程535并发请求授权和凭证数据,并且然后将数据从存储平台104下载到节点的本地存储器(例如,磁盘545),以便在操作675根据被调用的UDF中的指令进行进一步处理。

[0083] 图7示出了根据示例实施例的计算机系统形式的机器700的图解表示,在该计算机系统形式的机器700内可以执行指令集以使机器700执行本文所讨论方法中的任何一种或更多种方法。具体地,图7以计算机系统的示例形式示出了机器700的图解表示,用于使机器700执行本文所讨论方法中的任何一种或更多种方法的指令716(例如,软件、程序、应用、小程序、app、或其他可执行代码)可以在该计算机系统内执行。例如,指令716可以使机器700执行方法600中的任何一个或多个操作。作为另一个示例,指令716可以使机器700实现至少在图4中示出的数据流的部分。以这种方式,指令716将通用的、未编程的机器转换成特定的机器700(例如,计算服务管理器108或执行平台110中的节点),该特定的机器700被特别配置成以这里描述的方式执行所描述和示出的功能中的任何一个。

[0084] 在替代实施例中,机器700作为独立的设备操作或者可以耦合(例如,联网)到其它机器。在联网部署中,机器700可以在服务器-客户端网络环境中以服务器机器或客户端机器的身份来操作,或者作为对等(或分布式)网络环境中的对等机器来操作。机器700可以包括但不限于服务器计算机、客户端计算机、个人计算机(PC)、平板计算机、膝上型计算机、上网本(netbook)、智能电话、移动设备、网络路由器、网络交换机、网络桥接器(network bridge)或能够顺序地或以其它方式执行指令716的任何机器,指令716指定要由机器700采取的动作。此外,虽然仅示出了单个机器700,但术语“机器”还应被理解为包括单独或联合地执行指令716以执行本文所讨论方法中的任何一种或更多种方法的机器700的集合。

[0085] 机器700包括处理器710、存储器730和输入/输出(I/O)部件750,它们被配置成例

如经由总线702彼此通信。在示例实施例中,处理器710(例如,中央处理单元(CPU)、精简指令集计算(RISC)处理器、复杂指令集计算(CISC)处理器、图形处理单元(GPU)、数字信号处理器(DSP)、专用集成电路(ASIC)、射频集成电路(RFIC)、另一处理器或其任何合适的组合)可以包括例如可以执行指令716的处理器712和处理器714。术语“处理器”旨在包括多核处理器710,多核处理器710可以包括可以同时执行指令716的两个或更多个独立处理器(有时称为“核”)。尽管图7示出了多个处理器710,但是机器700可以包括具有单核的单个处理器、具有多核的单个处理器(例如,多核处理器)、具有单核的多个处理器、具有多核的多个处理器或者它们的任意组合。

[0086] 存储器730可以包括主存储器732、静态存储器734和存储单元736,它们都可以由处理器710例如经由总线702访问。主存储器732、静态存储器734和存储单元736存储指令716,指令716体现本文描述的方法或功能中的任何一种或更多种。在由机器700执行期间,指令716也可以全部或部分地驻留在主存储器732内、静态存储器734内、存储单元736的机器存储介质738内、至少一个处理器710内(例如,在处理器的高速缓冲存储器内)或其任何合适的组合内。

[0087] I/O部件750包括用于接收输入、提供输出、产生输出、传输信息、交换信息、捕获测量值等的部件。包括在特定机器700中的具体的I/O部件750将取决于机器的类型。例如,诸如移动电话的便携式机器将有可能包括触摸输入设备或其他这样的输入机构,而无头服务器(headless server)机器将不太可能包括这样的触摸输入设备。将认识到的是,I/O部件750可以包括图7中未示出的许多其他部件。I/O部件750根据功能进行分组仅仅是为了简化下面的讨论,并且该分组决不是限制性的。在各种示例实施例中,I/O部件750可以包括输出部件752和输入部件754。输出部件752可以包括视觉部件(例如,诸如等离子体显示面板(PDP)、发光二极管(LED)显示器、液晶显示器(LCD)、投影仪或阴极射线管(CRT)的显示器)、声学部件(例如,扬声器)、其他信号发生器等。输入部件754可以包括字母数字输入部件(例如,键盘、被配置成接收字母数字输入的触摸屏、光电键盘或其他字母数字输入部件)、基于指向的输入部件(例如,鼠标、触控板、轨迹球、操纵杆、运动传感器或另一指向仪器)、触觉输入部件(例如,物理按钮、提供触摸或触摸手势的位置和/或力的触摸屏或其他触觉输入部件)、音频输入部件(例如,麦克风)等。

[0088] 可以使用多种技术来实现通信。I/O部件750可以包括通信部件764,通信部件764可操作来分别经由耦合782和耦合772将机器700耦合到网络780或设备770。例如,通信部件764可以包括网络接口部件或与网络780对接的另一合适的设备。在进一步的示例中,通信部件764可以包括有线通信部件、无线通信部件、蜂窝通信部件和经由其他模态提供通信的其他通信部件。设备770可以是另一台机器或多种外围设备中的任一种(例如,经由通用串行总线(USB)耦合的外围设备)。例如,如上所述,机器700可以对应于计算服务管理器108或执行平台110中的任何一个,并且设备770可以包括客户端设备114或这里描述为与基于网络的数据仓库系统102或云存储平台104通信的任何其他计算设备。

[0089] 可执行指令和机器存储介质

[0090] 各种存储器(例如,730、732、734和/或处理器710的存储器和/或存储单元736)可以存储指令716和数据结构(例如,软件)的一个或更多个集合,这些指令716和数据结构(例如软件)体现本文所描述的方法或功能中的任何一种或更多种或被本文所描述的方法或功

能中的任何一种或更多种利用。当由处理器710执行时,这些指令716使各种操作实现所公开的实施例。

[0091] 如本文所使用的,术语“机器存储介质”、“设备存储介质”和“计算机存储介质”意思相同,并且可以在本公开中互换使用。这些术语指的是存储可执行指令和/或数据的单个或多个存储设备和/或介质(例如,集中式或分布式数据库和/或相关联的高速缓存和服务器等)。相应地,这些术语应当被认为包括但不限于固态存储器以及光介质和磁介质,包括在处理器内部或外部的存储器。机器存储介质、计算机存储介质和/或设备存储介质的具体示例包括非易失性存储器,非易失性存储器包括例如:半导体存储器设备,诸如可擦除可编程只读存储器(EPROM)、电可擦除可编程只读存储器(EEPROM)、现场可编程门阵列(FPGA)和闪存设备;磁盘,例如内部硬盘和可移除磁盘;磁光盘;以及CD-ROM和DVD-ROM光盘。术语“机器存储介质”、“计算机存储介质”和“设备存储介质”具体排除了载波、经调制的数据信号和其他这样的介质(其中的至少一些包含在下面讨论的术语“信号介质”中)。

[0092] 传输介质

[0093] 在各种示例实施例中,网络780的一个或多个部分可以是自组织网络(ad hoc network)、内联网、外联网、虚拟专用网络(VPN)、局域网(LAN)、无线LAN(WLAN)、广域网(WAN)、无线WAN(WWAN)、城域网(MAN)、互联网、互联网的一部分、公共交换电话网络(PSTN)的一部分、普通老式电话服务(POTS)网络、蜂窝电话网络、无线网络、Wi-Fi®网络、另一种类型的网络、或两种或更多种这样的网络的组合。例如,网络780或网络780的一部分可以包括无线或蜂窝网络,并且耦合782可以是码分多址(CDMA)连接、全球移动通信系统(GSM)连接或另一种类型的蜂窝或无线耦合。在该示例中,耦合782可以实现多种类型的数据传输技术中的任何一种,例如单载波无线电传输技术(1xRTT)、演进数据优化(Evolution-Data Optimized,EVDO)技术、通用分组无线电服务(GPRS)技术、增强型数据速率GSM演进(EDGE)技术、包括3G、第四代无线(4G)网络、通用移动通信系统(UMTS)、高速分组接入(HSPA)、全球微波接入互操作性(WiMAX)、长期演进(LTE)标准的第三代合作伙伴计划(3GPP)、由各种标准设置组织定义的其他技术、其他长程协议、或其他数据传输技术。

[0094] 指令716可以使用传输介质经由网络接口设备(例如,包括在通信部件764中的网络接口部件)并利用多种众所周知的传输协议(例如,超文本传输协议(HTTP))中的任何一种在网络780上被传输或接收。类似地,指令716可以使用传输介质经由耦合772(例如,对等耦合)被传输或接收到设备770。术语“传输介质”和“信号介质”意思相同,并且在本公开中可以互换使用。术语“传输介质”和“信号介质”应被理解为包括能够存储、编码或携带用于由机器700执行的指令716的任何无形介质,并且包括数字或模拟通信信号或便于这样的软件的通信的其他无形介质。因此,术语“传输介质”和“信号介质”应被理解为包括任何形式的经调制的数据信号、载波等。术语“经调制的数据信号”意指这样的信号:其具有以对信号中的信息进行编码的这样的方式设置或更改的特性中的一个或多个特性。

[0095] 计算机可读介质

[0096] 术语“机器可读介质”、“计算机可读介质”和“设备可读介质”意思相同,并且在本公开中可以互换使用。这些术语被定义为包括机器存储介质和传输介质二者。因此,这些术语包括存储设备/介质和载波/经调制的数据信号二者。

[0097] 本文描述的示例方法的各种操作可以至少部分地由被临时配置(例如,通过软件)

或永久配置为执行相关操作的一个或更多个处理器来执行。类似地,本文描述的方法可以至少部分地是处理器实现的。例如,方法600的至少一些操作可以由一个或更多个处理器执行。某些操作的执行可以在一个或更多个处理器之间分配,该一个或更多个处理器不仅驻留在单个机器内,而且跨多个机器部署。在一些示例实施例中,一个或更多个处理器可以位于单个位置(例如,在家庭环境、办公室环境或服务器场内)中,而在其他实施例中,处理器可以分布在多个位置上。鉴于上面的公开内容,下面阐述了各种示例。应当注意,一个示例的一个或更多个特征,无论是孤立地还是组合地,都应当被认为是在本申请的公开内容之内。

[0098] 示例1.一种方法,包括:在网络数据库的第一计算进程中接收多个用户自定义函数定义(user defined function definition),每个用户自定义函数定义包括用户自定义函数类的类型(user defined function class type)和将在第二计算进程中执行的函数的函数名;由网络数据库接收用户自定义函数,该用户自定义函数包括与使用用户自定义函数类的类型执行的至少一个操作相关的代码,用户自定义函数具有与多个用户自定义函数定义之一匹配的函数名;接收对于执行用户自定义函数的请求,该请求由网络数据库的第二计算进程生成;激活用户自定义函数类的类型的、对应于所请求的用户自定义函数的一个或更多个对象;由联网数据库的节点,使用到文件存储暂存区的路径检索在请求中指定的、对应于用户自定义类的类型的一个或更多个激活的对象的文件;并将文件存储在节点上。

[0099] 示例2.根据示例1所述的方法,其中,文件存储暂存区对应于联网数据库外部的的外部存储设备。

[0100] 示例3.根据示例1或2中任一项或更多项所述的方法,其中,用户自定义函数类的类型的一个或更多个对象在用户自定义函数服务器上被激活。

[0101] 示例4.根据示例1-3中任一项或更多项所述的方法,还包括:使用节点内的网络通道将来自一个或更多个对象的请求传输到节点的多个线程。

[0102] 示例5.根据示例1-4中任一项或更多项所述的方法,其中,网络通道是远程过程调用(RPC)通道。

[0103] 示例6.根据示例1-5中任一项或更多项所述的方法,还包括:基于第二计算进程的用户先前被第一计算进程的另一用户授予访问文件之一中的每一个,通过网络数据库确定每个线程被授权访问所述文件之一。

[0104] 示例7.根据示例1-6中任一项或更多项所述的方法,还包括:通过第二计算进程,根据至少一个操作处理文件。

[0105] 示例8.一种系统,包括:机器的一个或更多个处理器;以及存储器,该存储器存储指令,该指令当被一个或更多个处理器执行时使机器执行包括根据示例1-7中所述的方法中的任一项或更多项的操作。

[0106] 示例9.一种包含指令的机器存储介质,该指令当由机器执行时使机器执行包括根据示例1-7中所述的方法中的任一项或更多项的操作。

[0107] 尽管参考具体的示例实施例描述了本公开的实施例,但显然地,在不偏离本发明主题的更广泛范围的情况下,可以对这些实施例做出各种修改和改变。因此,说明书和附图被认为是说明性的而不是限制性的。形成本申请的一部分的附图通过说明而非限制的方式

示出了其中可以实施主题的具体实施例。所示出的实施例以充分的细节被描述,以使本领域的技术人员能够实施本文公开的教导。可以使用其他实施例和从其中衍生的实施例,使得可以做出结构和逻辑的替换和改变而不偏离本公开的范围。因此,此详细描述不应被理解为限制性意义,并且各种实施例的范围仅由所附权利要求连同这些权利要求所享有的完整范围的等同物一起限定。

[0108] 本发明主题的这样的实施例可以在本文中单独地和/或共同地由术语“发明”提到,这仅仅是为了便利,并不意图自愿将本申请的范围局限于任何单一发明或发明构思(如果实际上公开了多于一种发明或发明构思的话)。因此,虽然本文中已经说明和描述了特定实施例,但是应当理解,可以用经计算以实现相同目的的任何布置来替换所示出的特定实施例。本公开旨在覆盖各种实施例的任何和所有修改或变型。在阅读以上描述后,本领域的技术人员将明白以上实施例的组合以及本文未具体描述的其他实施例。

[0109] 在本文档中,术语“一个(a)”或“一个(an)”,如专利文档中常见的那样,用于包括一个或多于一个,与“至少一个”或“一个或更多个”的任何其他实例或用法无关。在本文档中,术语“或”用于指非排他性的或,因此除非另有说明,否则“A或B”包括“A但不包括B”、“B但不包括A”和“A和B”。在所附权利要求中,术语“包括(including)”和“其中(in which)”被用作相应术语“包括(comprising)”和“其中(wherein)”的纯英语等效词。此外,在所附权利要求中,术语“包括(including)”和“包括(comprising)”是开放式的;也就是说,包括除了权利要求中在这样的术语之后列出的那些元素之外的元素的系统、设备、物品或过程仍然被认为落入该权利要求的范围内。

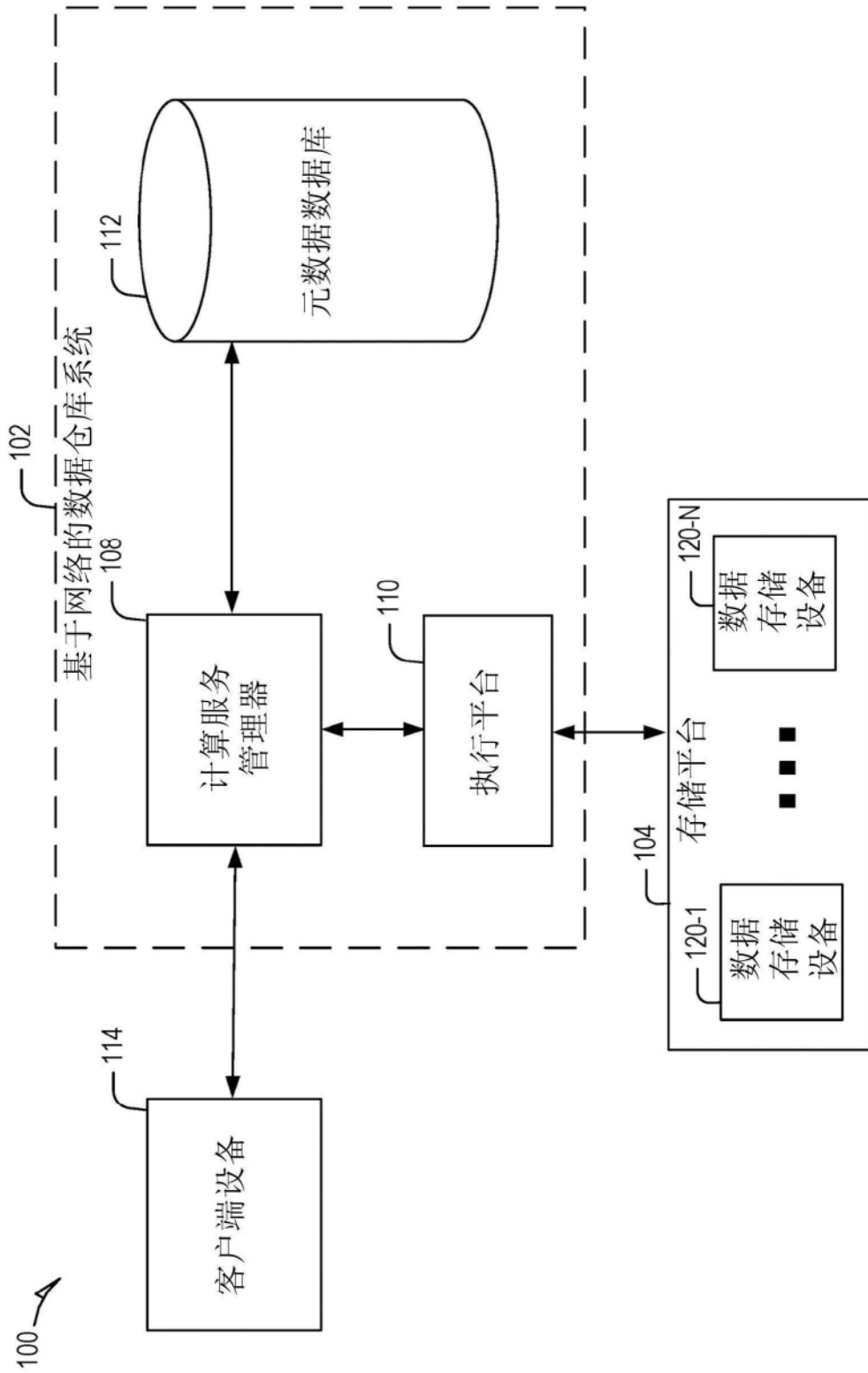


图1

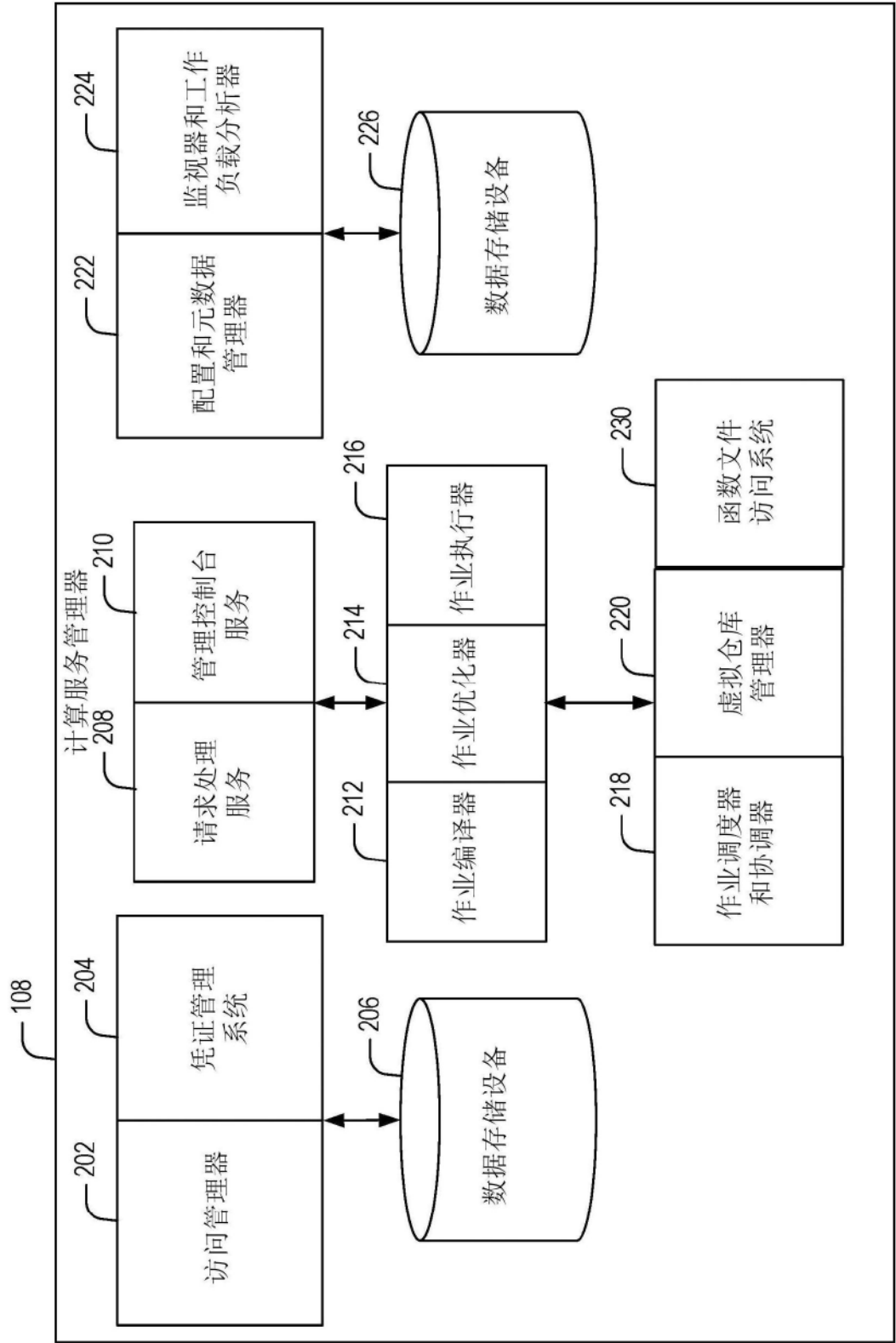


图2

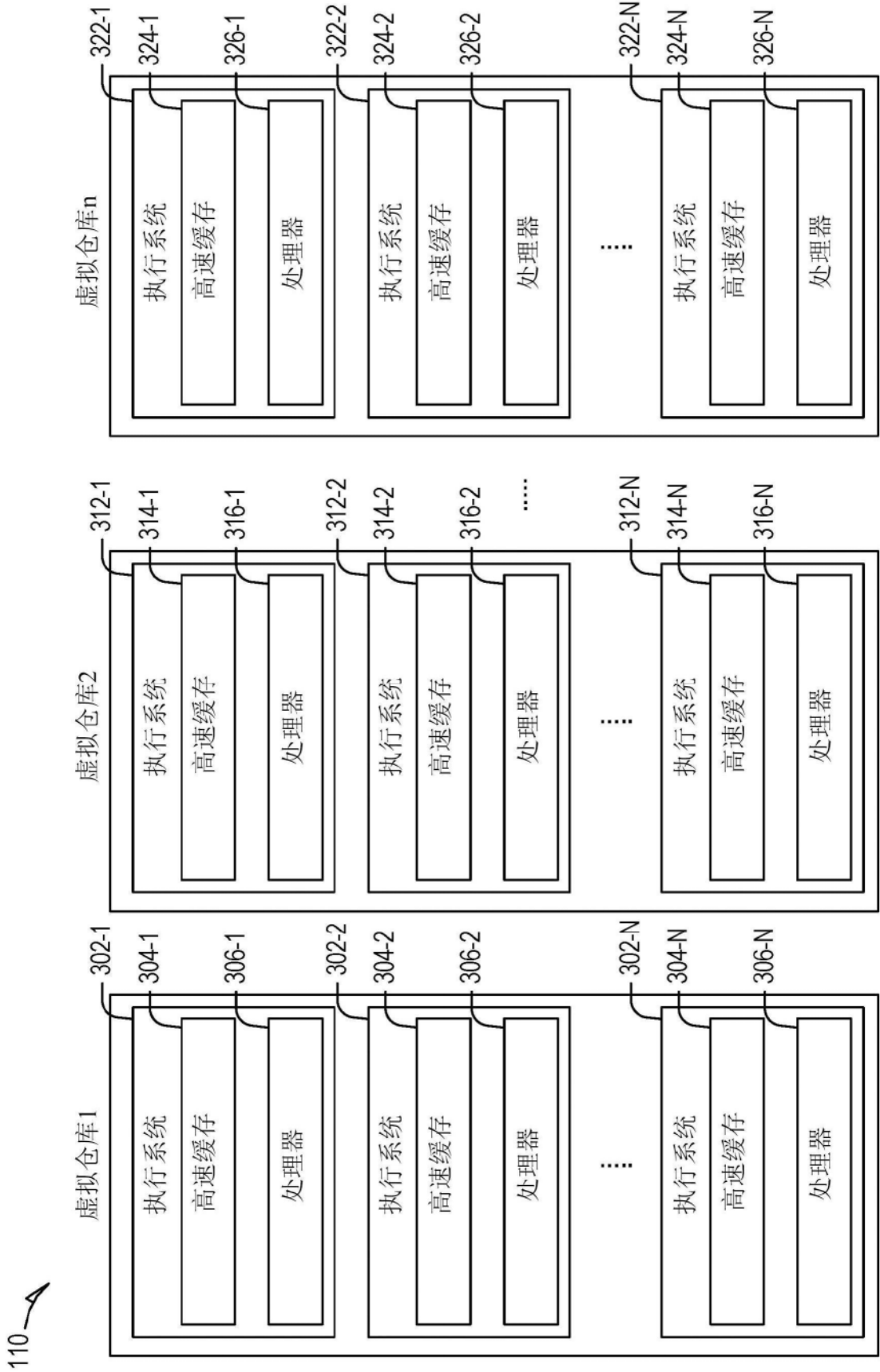


图3

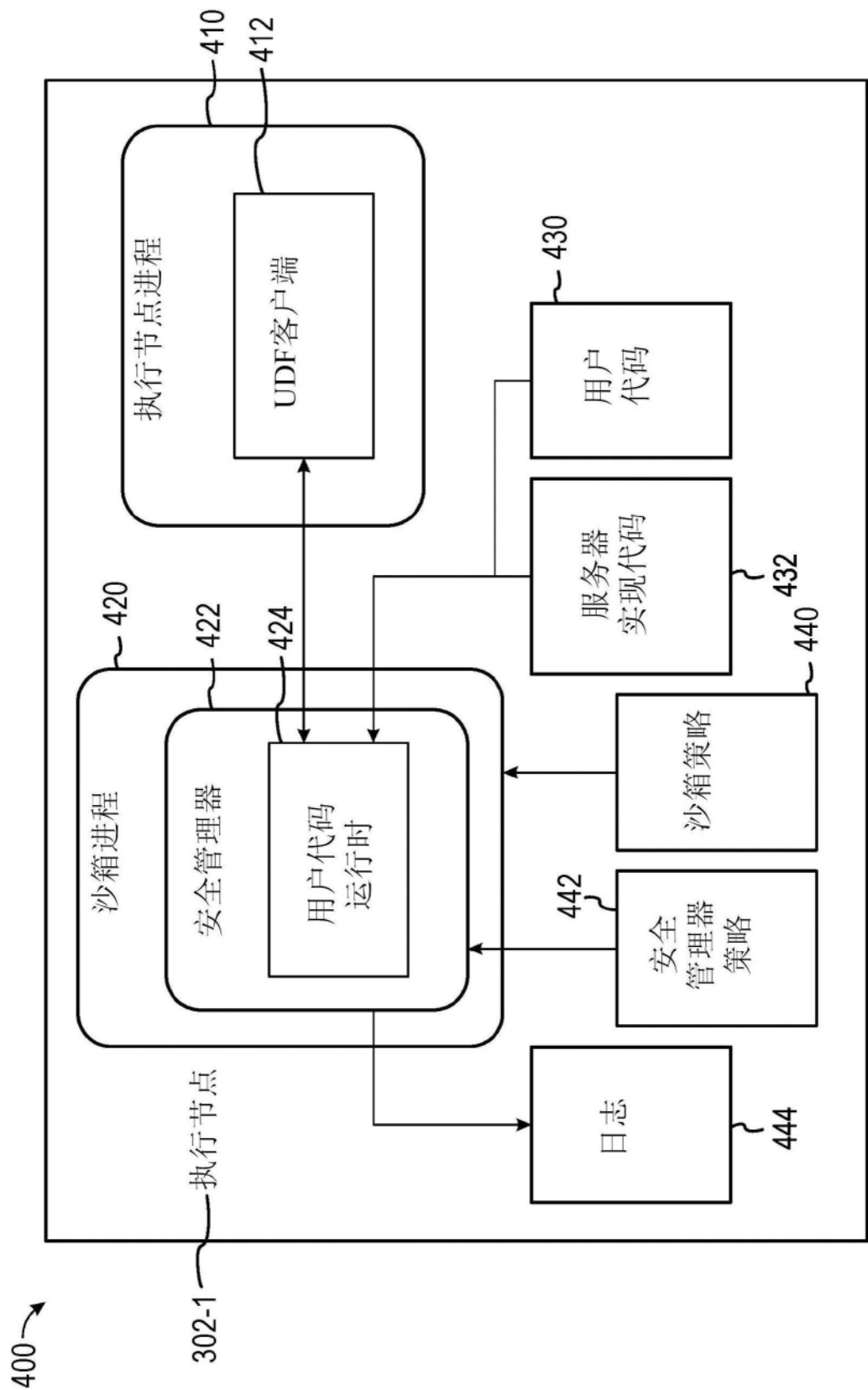


图4

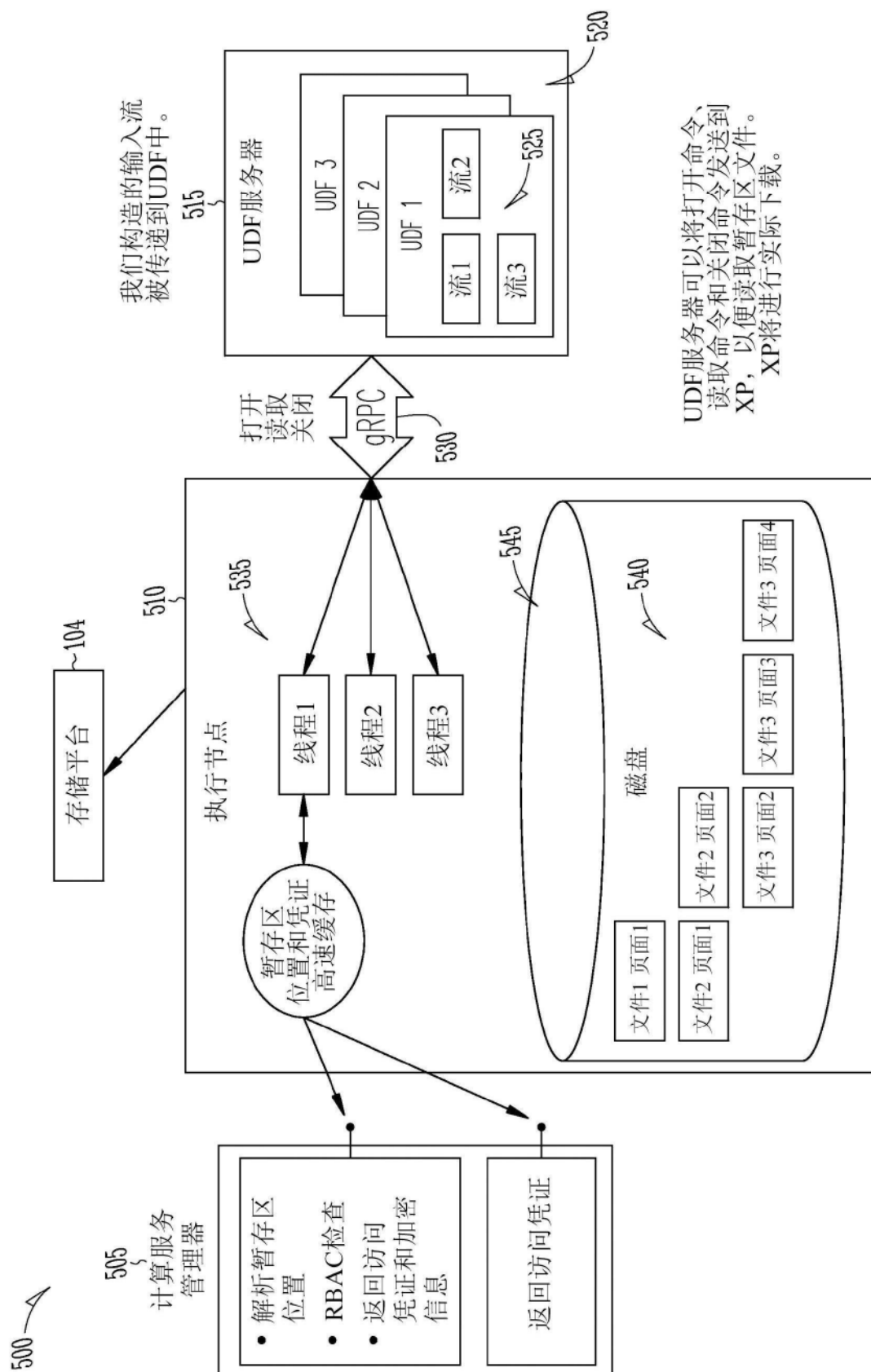


图5

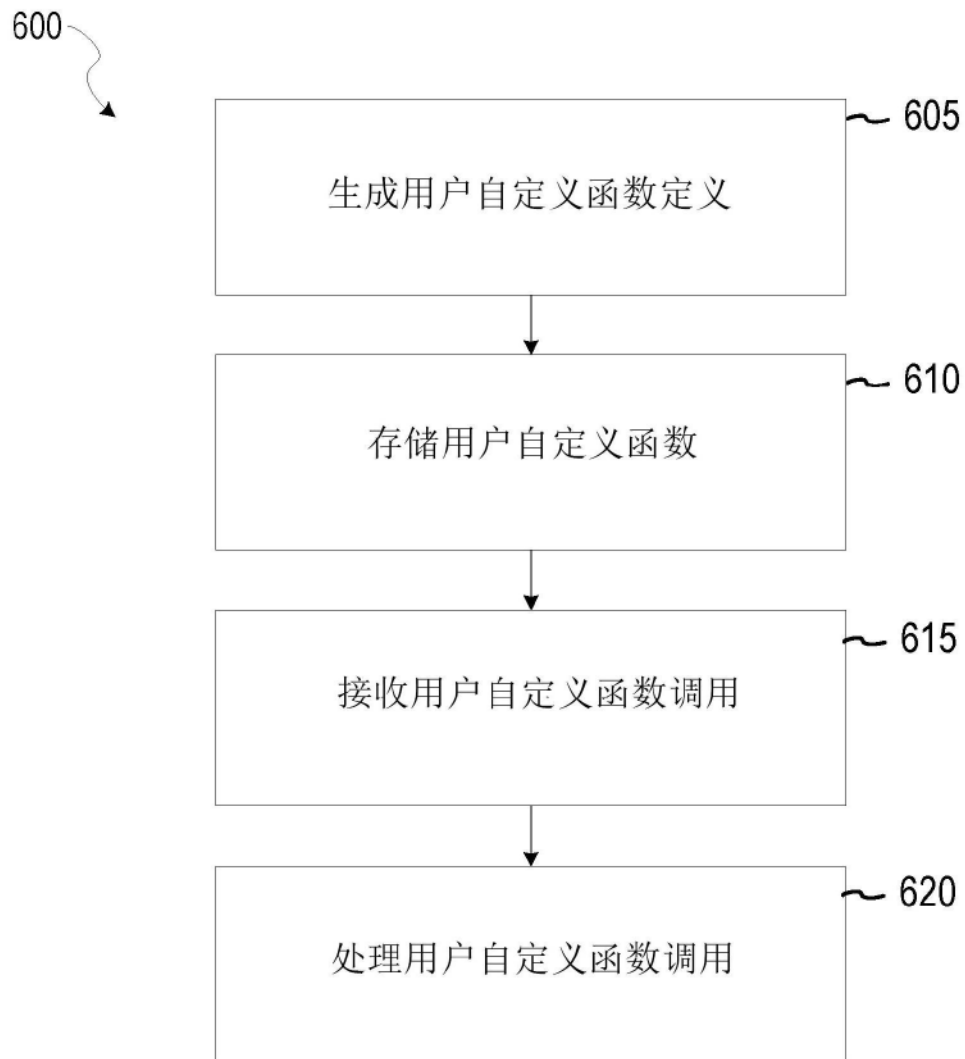


图6A

650

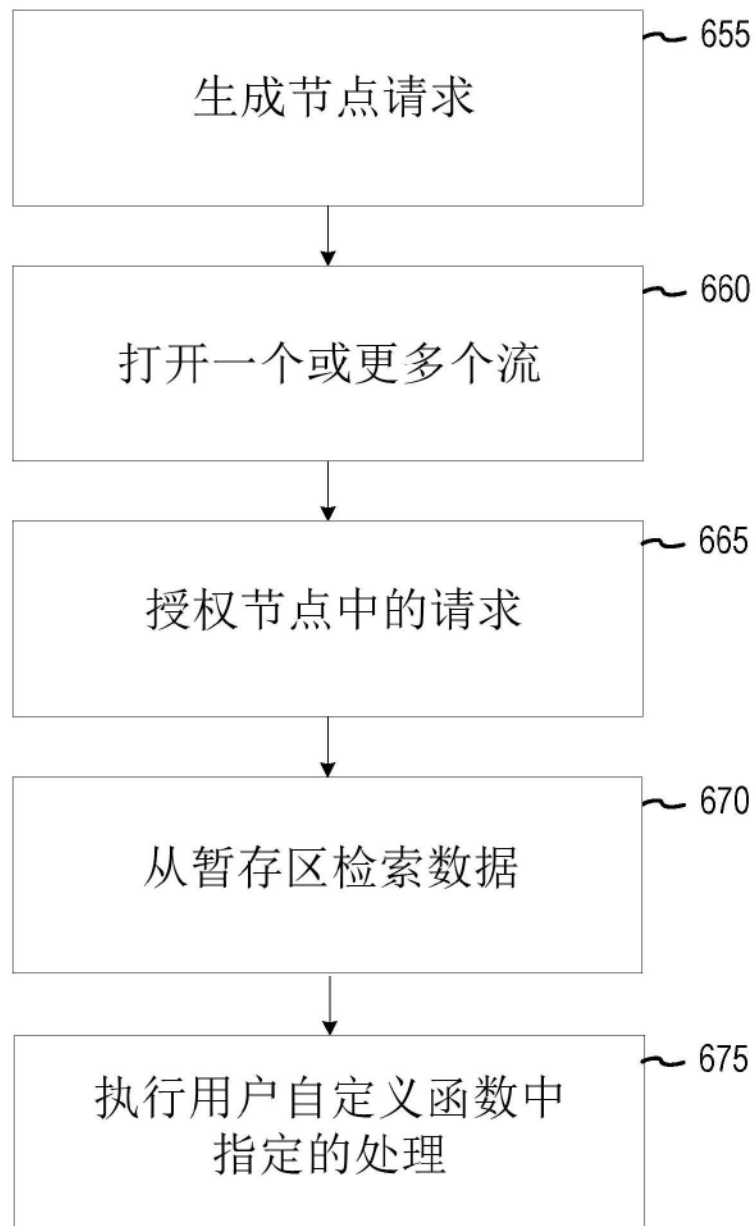


图6B

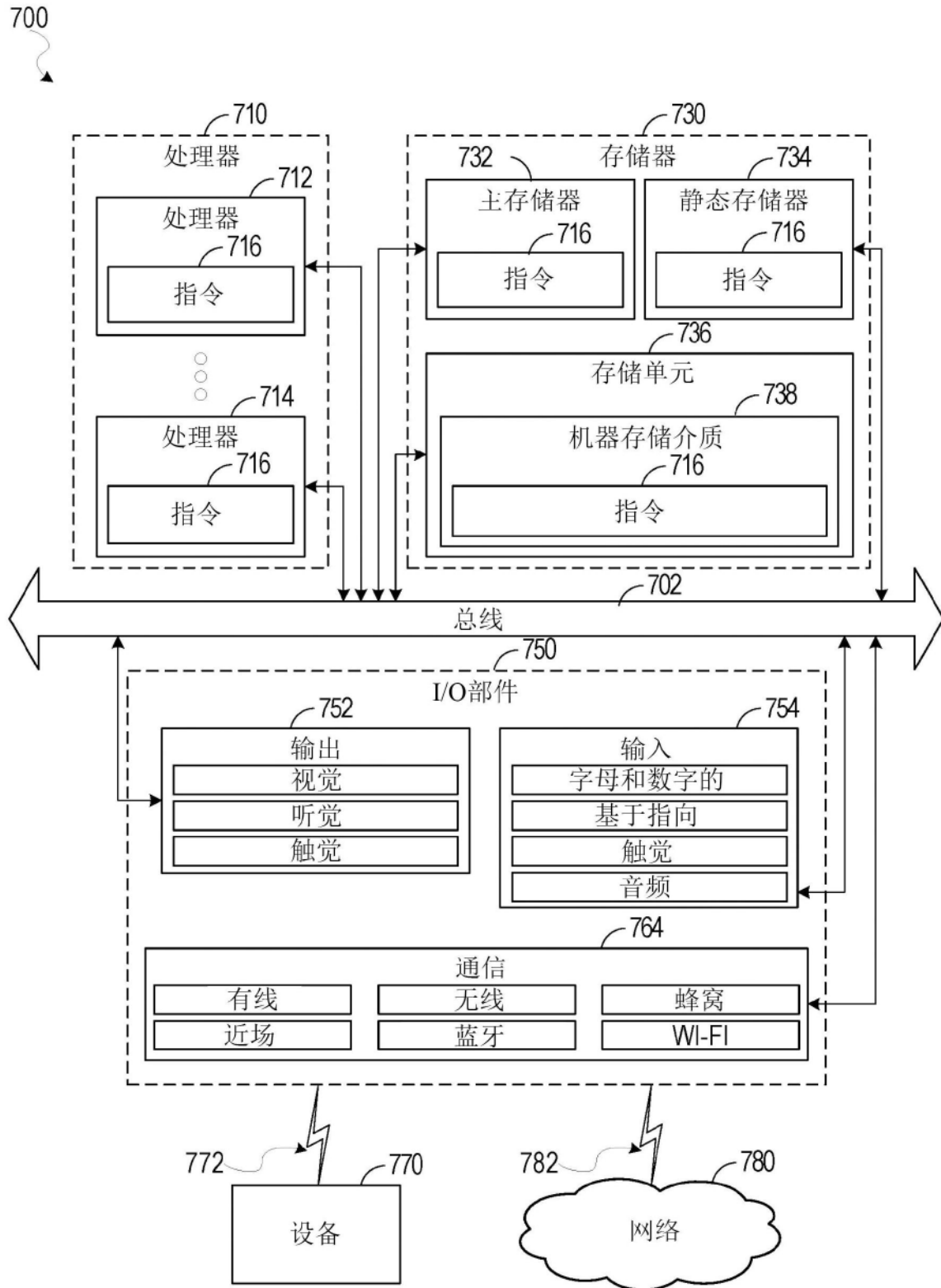


图7