



(12) 发明专利申请

(10) 申请公布号 CN 117235339 A

(43) 申请公布日 2023. 12. 15

(21) 申请号 202310700375.8

G06F 16/953 (2019.01)

(22) 申请日 2023.06.13

G06F 21/53 (2013.01)

G06F 21/60 (2013.01)

(30) 优先权数据

63/366,266 2022.06.13 US

18/053,956 2022.11.09 US

(71) 申请人 斯诺弗雷克公司

地址 美国蒙大拿州

(72) 发明人 陈宇睿 恩美西·杰戈塔普

威廉·A·普 布莱恩·史密斯

徐徐

(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 16/951 (2019.01)

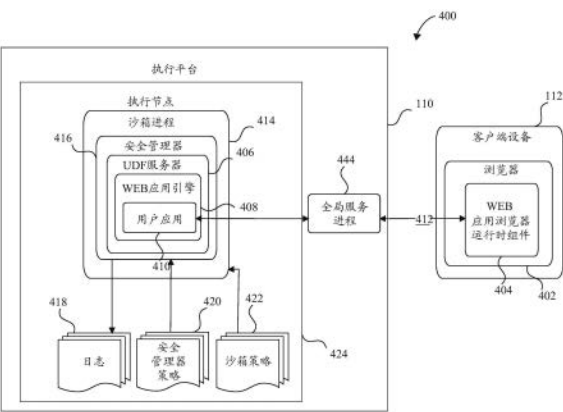
权利要求书1页 说明书19页 附图9页

(54) 发明名称

具有统一特权的数据平台

(57) 摘要

本申请涉及具有统一特权的数据平台。一种用于在统一安全上下文内开发和部署用户应用的数据平台。该数据平台基于安全上下文的安全策略授权第一用户使用编辑器访问用户应用的源代码,并基于安全上下文的安全策略授权第一用户使用应用和数据管理器来设置使第二用户使用用户应用的使用特权。为了向第二用户提供用户应用,数据平台通过以下方式部署用户应用:在安全上下文内实例化用户定义函数(UDF)服务器和UDF服务器的应用引擎,在安全上下文内将用户应用实例化为应用引擎的应用,以及基于安全上下文的安全策略授权用户应用对数据的访问。



1. 一种数据平台,包括:

一个或多个处理器;以及

存储指令的至少一个存储器,所述指令当由所述一个或多个处理器执行时使得所述数据平台执行包括以下项的操作:

基于安全上下文的安全策略授权第一用户使用编辑器来访问用户应用的源代码;

基于所述安全上下文的所述安全策略授权所述第一用户使用应用和数据管理器来设置使第二用户使用所述用户应用的使用特权;以及

通过执行包括以下项的操作,基于所述安全上下文的所述安全策略向所述第二用户提供所述用户应用:

在所述安全上下文内实例化用户定义函数服务器;

在所述安全上下文内实例化所述用户定义函数服务器的应用引擎;

在所述安全上下文内将所述用户应用实例化为所述应用引擎的应用;以及

基于所述安全上下文的所述安全策略授权所述用户应用对数据的访问。

2. 一种计算机实现的方法,包括:

由一个或多个处理器基于安全上下文的安全策略授权第一用户使用编辑器访问用户应用的源代码;

由所述一个或多个处理器基于所述安全上下文的所述安全策略来授权所述第一用户使用应用和数据管理器来设置使第二用户使用所述用户应用的使用特权;以及

由所述一个或多个处理器通过执行包括以下项的操作来基于所述安全上下文的所述安全策略向所述第二用户提供所述用户应用:

在所述安全上下文内实例化用户定义函数服务器;

在所述安全上下文内实例化所述用户定义函数服务器的应用引擎;

在所述安全上下文内将所述用户应用实例化为所述应用引擎的应用;以及

基于所述安全上下文的所述安全策略授权所述用户应用对数据的访问。

3. 一种计算机存储介质,其包括指令,所述指令在由机器的一个或多个处理器执行时,将所述机器配置成执行包括以下项的操作:

基于安全上下文的安全策略授权第一用户使用编辑器访问用户应用的源代码;

基于所述安全上下文的所述安全策略授权所述第一用户使用应用和数据管理器来设置使第二用户使用所述用户应用的使用特权;以及

通过执行包括以下项的操作,基于所述安全上下文的所述安全策略向所述第二用户提供所述用户应用:

在所述安全上下文内实例化用户定义函数服务器;

在所述安全上下文内实例化所述用户定义函数服务器的应用引擎;

在所述安全上下文内将所述用户应用实例化为所述应用引擎的应用;以及

基于所述安全上下文的所述安全策略授权所述用户应用对数据的访问。

具有统一特权的数据平台

技术领域

[0001] 本公开的示例总体上涉及数据库,更具体地,涉及数据库安全性。

[0002] 背景

[0003] 数据平台广泛用于计算和通信环境中的数据存储和数据访问。关于架构,数据平台可以是本地化部署的数据平台 (on-premises data platform)、基于网络的数据平台 (例如,基于云的数据平台)、两者的组合,和/或包括另一种类型的架构。关于数据处理,数据平台可以实现在线事务处理 (OLTP)、在线分析处理 (OLAP)、两者的组合和/或另一种类型的数据处理。此外,数据平台可以是或包括关系数据库管理系统 (RDBMS) 和/或一个或多个其他类型的数据库管理系统。

[0004] 数据平台上数据的提供者可能希望有一种方法来方便地保护他们的数据。

[0005] 附图简述

[0006] 从下面给出的详细描述和本公开的各种示例的附图中,将更全面地理解本公开。

[0007] 图1示出了根据本公开的一些示例的示例计算环境,其包括与云存储提供商系统通信的基于网络的数据平台。

[0008] 图2是示出根据本公开的一些示例的计算服务管理器的组件的框图。

[0009] 图3是示出根据本公开的一些示例的执行平台的组件的框图。

[0010] 图4A是根据本公开的一些示例的用于提供应用的计算环境的部署图。

[0011] 图4B、图4C和图4D是根据本公开的一些示例的用于提供用户应用的计算环境的交互和数据流图。

[0012] 图5是根据本公开的一些示例的数据平台的计算环境的部署图,该计算环境为用户应用的开发和部署提供统一安全上下文。

[0013] 图6是示出根据本公开的一些示例的由数据平台102在统一安全上下文中开发和部署用户应用的方法的活动图。

[0014] 图7示出了根据本公开的一些示例的计算机系统形式的机器的图解表示,可以在该计算机系统形式的机器内执行指令集合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。

[0015] 详细描述

[0016] 有许多现有的解决方案用于构建访问数据平台上数据的应用,但大多数解决方案涉及用于编写应用代码、部署代码、运行应用、更新应用和使用应用的不同许可边界。跨越安全边界的这些间隙 (gap) 是可能出错的地方,或者是需要存储授权材料以便与系统交互的地方。例如,用户可能需要在持续集成作业 (job) 中存储授权材料,以便将构建的工件 (artifact) 推送到工件管理系统。即使在简单的系统 (其中应用的创建和使用经常发生在同一个组织中) 中,也需要存储数据平台的凭证。通过提供以下项的组合,用户能够创建、部署和执行应用,而无需在平台外位置 (off-platform location) 存储凭证或其他授权材料: 用于在数据平台内编辑和创建应用的编辑器; 用于提供对象作为本机数据库对象的方法,该本机数据库对象作为用户界面 (UI) 应用运行; 用于打包和共享应用的方法; 和用于跨数

据平台的帐户共享数据的方法,所有这些都在相同的安全上下文内操作。

[0017] 在一些示例中,开发、操作和最终用户在相同的安全上下文中共享,该安全上下文具有定义该安全上下文的安全边界的统一特权(privileges)。系统的各种组件可以直接在数据平台上工作。这提供了跨用户应用生命周期的各个方面的授权边界。诸如分发用户应用的操作可以在数据平台的本机应用框架内操作,以减少附加安全边界的数量。因此,用户可以避免在平台外的数据存储器中保存用于集成系统的认证或授权材料。任何时候为以下操作存储安全凭证时,都存在潜在的弱点:诸如在系统之间推送代码、在系统之间推送构建的工件、启动计算资源或操作正在运行的系统。这些弱点包括授权材料可能被盗,授权材料可能过期,导致可操作系统在更换之前停止工作。

[0018] 在一些示例中,应用的从概念到交付的整个生命周期是在数据平台上的统一安全上下文内管理的。由于在工件的整个生命周期中对工件的所有工作都存储在数据平台上,并且使用数据平台的应用和数据共享管理器共享应用的分发版本(distribution build),因此提供了单一的安全边界。

[0019] 在一些示例中,数据平台提供应用编辑表面,该应用编辑表面将文件写入数据平台内的数据存储器,以存储应用的源代码。

[0020] 在一些示例中,通过连接到由数据平台提供的安全Web应用形式的用户应用来提供应用源代码的交互式编辑会话。

[0021] 在一些示例中,使用由数据平台提供的编辑器来提供应用源代码的交互式编辑会话。

[0022] 在一些示例中,通过修改用户应用或存储用户应用的源代码的数据存储器以改变当前附到由用户应用使用的默认版本的源代码来实现将开发代码推广到生产。最终用户命中(hit)用户应用端点,其中他们的认证上下文被用于启动用户应用,并且用户应用在数据平台内已经配置的安全上下文中运行。

[0023] 在一些示例中,一种用于在统一安全上下文中开发和部署用户应用的数据平台。该数据平台基于安全上下文的安全策略授权第一用户使用编辑器访问用户应用的源代码,并基于安全上下文的安全策略授权第一用户使用应用和数据管理器来设置使第二用户使用用户应用的使用特权。为了向第二用户提供用户应用,数据平台通过以下方式部署用户应用:在安全上下文内实例化用户定义函数(UDF)服务器和UDF服务器的应用引擎,在安全上下文内将用户应用实例化为应用引擎的应用,以及基于安全上下文的安全策略授权用户应用访问数据平台的数据。

[0024] 在一些示例中,数据平台包括一个或多个处理器和存储指令的存储器。当指令由一个或多个处理器执行时,数据平台基于安全上下文的安全策略授权第一用户使用编辑器访问用户应用的源代码,并且基于安全上下文的安全策略授权第一用户使用应用和数据管理器来设置使第二用户使用用户应用的使用特权。数据平台通过以下操作来基于安全上下文的安全策略将用户应用提供给第二用户:在安全上下文内实例化用户定义函数(UDF)服务器,在安全上下文内实例化UDF服务器的应用引擎,在安全上下文内将用户应用实例化为应用引擎的应用,以及基于安全上下文的安全策略授权用户应用访问数据平台的数据。

[0025] 现在将详细参考用于实现本发明主题的具体示例。这些具体示例的示例在附图中

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到所示出的示例。相反,他们旨在覆盖如可以包括在本公开范围内的这种替代方案、修改和等同物。

[0026] 图1示出了根据本公开的一些示例的示例计算环境100,其包括与客户端设备112通信的数据平台102。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。

[0027] 如图所示,数据平台102包括数据库存储装置(database storage)106、计算服务管理器104、执行平台110和元数据数据库114。数据库存储装置106包括多个(a plurality of)计算机,并按需向数据平台102提供计算机系统资源,例如数据存储和计算能力。如图所示,数据库存储装置106包括多个数据存储设备,即数据存储设备1 108a到数据存储设备N 108d。在一些示例中,数据存储设备1至N是位于一个或多个地理位置的基于云的存储设备。例如,数据存储设备1至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1至N可以包括硬盘驱动器(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,数据库存储装置106可以包括分布式文件系统(例如,Hadoop分布式文件系统(HDFS))、对象存储系统等。

[0028] 数据平台102用于报告和分析来自一个或多个不同源的集成数据,该一个或多个不同源包括数据库存储装置106内的存储设备1至N。数据平台102托管数据报告和分析服务并向多个客户帐户提供数据报告和分析服务。管理用户可以创建和管理身份(例如,用户、角色和组),并使用特权来允许或拒绝身份对资源和服务的访问。通常,数据平台102为多个相应客户维护多个客户帐户。数据平台102在数据库存储装置106的一个或多个存储设备中维护每个客户帐户。此外,数据平台102可以在元数据数据库114中维护与客户帐户相关联的元数据。每个客户帐户包括多个数据对象,数据对象的示例包括用户、角色、特权、数据存储器或其他数据位置(本文称为“暂存区(stage)”或“多个暂存区(stages)”)等。

[0029] 计算服务管理器104协调和管理数据平台102的操作。计算服务管理器104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104可以支持任意数量和任何类型的客户端,例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104交互的其他组件/设备。作为示例,计算服务管理器104与客户端设备112通信。由数据平台102支持的多个客户帐户中的一个帐户的用户可以使用客户端设备112来与数据平台102交互并利用数据平台102的功能。在一些示例中,计算服务管理器104不从客户端设备112接收任何直接通信,并且仅从数据平台102内的队列接收关于作业的通信。

[0030] 计算服务管理器104还耦合到元数据数据库114。元数据数据库114存储关于与数据平台102及其用户相关联的各种功能和方面的数据。在一些示例中,元数据数据库114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(summary)。另外,元数据数据库114可以包括关于数据在远程数据存储系统(例如,数据库存储装置106)和本地高速缓存中如何被组织的信息。元数据数据库114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。

[0031] 计算服务管理器104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。在一些示例中,计算服务管理器104使用数据平台102内的队列与执行平台110就作业和任务进行通信。这隔离了执行平台110和客户端设备112的操作。执行平台110耦合到数据库存储装置106。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104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;使用最近最少使用(least recently used,LRU)策略来监视和删除微分区文件,并实现内存不足(OOM)错误缓解进程的第二进程;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104的第三进程;在系统引导(boot)之后建立与计算服务管理器104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104提供的给定作业的计算集群的所有通信,并将信息传送回计算服务管理器104和执行平台110的其他计算节点。

[0032] 在一些示例中,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。在一些示例中,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。在替代示例中,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(数据存储设备1 108a至数据存储设备N108d)从与执行平台110相关联的计算资源解耦(decouple)。该架构支持数据平台102基于变化的数据存储/检索需求以及变化的用户和系统需求进行动态改变。对动态变化的支持允许数据平台102响应于对数据平台102内的系统和组件的不断变化的需求而快速扩展。计算资源与数据存储设备的解耦支持大量数据的存储,而无需相应的大量计算资源。类似地,资源的这种解耦支持在特定时间使用的计算资源的显著增加,而无需相应增加可用数据存储资源。

[0034] 计算服务管理器104、元数据数据库114、执行平台110和数据库存储装置106在图1中被示为单独的分立组件。然而,计算服务管理器104、元数据数据库114、执行平台110和数据库存储装置106中的每一个都可以被实现为分布式系统(例如,分布在位于多个地理位置的多个系统/平台上)。另外,计算服务管理器104、元数据数据库114、执行平台110和数据库存储装置106中的每一个可以根据所接收的请求的变化和数据平台102的变化的需求而(彼此独立地)使其规模放大或缩小。因此,在所描述的示例中,数据平台102是动态的,并且支持定期变化以满足当前的数据处理需求。

[0035] 在操作期间,数据平台102处理由计算服务管理器104确定的多个作业。这些作业由计算服务管理器104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104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104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104可以确定处理任务需要什么数据,并进一步确定执行平台110内的哪些节点最适合处理任务。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114中的元数据帮助计算服务管理器104确定执行平台110中的哪些节点已经高速缓存了处理任务所需的数据的至少一部分。执行平台110中的一个或更多个节点使用由这些节点缓存的数据以及必要时使用从数据库存储装置106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数据库存储装置106中检索数据更快。

[0036] 如图1所示,计算环境100将执行平台110与数据库存储装置106分离。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数据库存储装置106中的数据存储设备(数据存储设备1 108a到数据存储设备N 108d)而操作。因此,计算资源和高速缓存资源不限于数据存储设备1 108a到数据存储设备N 108d中特定的一个。相反,所有计算资源和所有高速缓存资源都可以从数据库存储装置106中的任何数据存储资源中检索数据并将数据存储到其中。

[0037] 图2是示出根据本公开的一些示例的计算服务管理器104的组件的框图。如图2所示,计算服务管理器104包括耦合到数据存储设备206的访问管理器202和密钥管理器204。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。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,数据库存储装置106中的数据存储设备)中的数据的密钥。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”。

[0038] 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可以确定处理接收到的查询(例如,数据存储请求或数据检索请求)所需的数据。数据可以被存储在执行平台110内的高速缓存中,或者被存储在数据库存储装置106中的数据存储设备中。

[0039] 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可以接收执行作业并监视系统上的工作负荷的请求。

[0040] 计算服务管理器104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佳方法。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104确定的作业的执行代码。

[0041] 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,可以按该优先顺序对作业进行优先考虑和处理。在一些示例中,作业调度器和协调器218确定由计算服务管理器104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。在一些示例中,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。如下所述,每个虚拟仓库都包括多个执行节点,每个执行节点都包括一个高速缓存和一个处理器。

[0042] 另外,计算服务管理器104包括配置和元数据管理器222,配置和元数据管理器222管理与存储在远程数据存储设备和本地高速缓存(例如,执行平台110中的高速缓存)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微分区来检索用于处理特定任务或作业的数据。监视器和工作负荷分析器224监督由计算服务管理器104执行的进程,并管理跨执行平台110中的虚拟仓库和执行节点的任务(例如,工作负荷)的分配。监视器和工作负荷分析器224还根据需要,基于整个数据平台102中变化的工作负荷来重新分配任务,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”)查询工作负荷来重新分配任务。配置和元数据管理器222以及监视器和工作负荷分析器224耦

合到数据存储设备226。图2中的数据存储设备226表示数据平台102内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中的高速缓存、数据库存储装置106中的存储设备或任何其他存储设备。

[0043] 计算服务管理器104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被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1 304a)可能需要与另一个执行节点(例如,执行节点2 304b)通信,并且应该被禁止与第三执行节点(例如,执行节点1 316a)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。

[0044] 图3是示出根据本公开的一些示例的执行平台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 302a和虚拟仓库2 302b至虚拟仓库N 302c。每个虚拟仓库都包括多个执行节点,每个执行节点都包括一个数据高速缓存和一个处理器。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(drop)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。所有虚拟仓库可以从任何数据存储设备(例如,数据库存储装置106中的任何存储设备)访问数据。

[0045] 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。

[0046] 每个虚拟仓库能够访问图1中所示的数据存储设备1至N中的任何一个。因此,虚拟仓库不一定被分配给具体的数据存储设备1至N,而是可以访问来自数据库存储装置106内的数据存储设备1至N中的任何一个的数据。类似地,图3中所示的每个执行节点可以访问来自数据存储设备1至N中的任何一个的数据。在一些示例中,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稍后可以访问来自任何其他数据存储设备的数据。

[0047] 在图3的示例中,虚拟仓库1 302a包括多个执行节点,如执行节点1 304a、执行节点2 304b和执行节点N 304c所示。执行节点1 304a包括高速缓存1 306a和处理器1 308a。执行节点2 304b包括高速缓存2 306b和处理器2 308b。执行节点N 304c包括高速缓存N 306c和处理器N 308c。每个执行节点1至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(例如,集群服务、物化视图刷新服务、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。

[0048] 类似于上面讨论的虚拟仓库1 302a,虚拟仓库2 302b包括多个执行节点,如执行节点1 310a、执行节点2 310b和执行节点N 310c所示。执行节点1 310a包括高速缓存1 312a和处理器1 314a。执行节点2 310b包括高速缓存2 312b和处理器2 314b。执行节点N

310c包括高速缓存N 312c和处理器N 314c。另外,虚拟仓库N 302c包括多个执行节点,如执行节点1 316a、执行节点2 316b和执行节点N 316c所示。执行节点1 316a包括高速缓存1 318a和处理器1 320a。执行节点2 316b包括高速缓存2 318b和处理器2 320b。执行节点N 316c包括高速缓存N 318c和处理器N 320c。

[0049] 在一些示例中,相对于执行节点正在缓存的数据而言,图3所示的执行节点是无状态的。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或由特定执行节点缓存的数据的状态信息。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。

[0050] 尽管图3中所示的执行节点各自包括一个数据高速缓存和一个处理器,但是替代示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数据库存储装置106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。在一些示例中,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。每个高速缓存可以储存来自数据库存储装置106中的任何存储设备的数据。

[0051] 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。在一些示例中,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。

[0052] 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,则可以以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,则可以以为执行节点分配更多的高速缓存资源。

[0053] 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和虚拟仓库N由第二地理位置处的另一计算系统来实现。在一些示例中,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系统。

[0054] 另外,如图3中示出的每个虚拟仓库具有多个执行节点。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 302a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1 304a和执行节点2 304b,并在另一地理位置处的不同计算平台处实现执行节点N 304c。选择特定的计算系统以实现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。

[0055] 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资源和/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。

[0056] 在一些示例中,虚拟仓库可以对数据库存储装置106中的同一数据进行操作,但是每个虚拟仓库具有其自己的执行节点,其自己的执行节点具有独立的处理资源和高速缓存资源。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。

[0057] 图4A是根据一些示例的用于提供Web应用作为第一类数据库对象的计算环境400的部署图。数据平台102利用计算环境400为要由数据平台102的执行平台110执行的用户应用410提供安全框架。用户应用410和支持用户应用410的所有组件(例如但不限于Web应用引擎408和用户定义函数(UDF)服务器406,在本文统称为“Web应用”)被数据平台102视为第一类数据库对象,因为可以使用数据库查询内的一个或更多个命令来实例化,如以下代码段所示:

```
CREATE [ OR REPLACE ] WEBAPP [ IF NOT EXISTS ] <Webapp_name>

VERSION = (

    id = <version_name>

    root_location = <app_root>
[0058]    file_path = <file_name>

)

[ WAREHOUSE = <warehouse_name> ]

[ COMMENT = '<comment_string_literal>' ]
```

[0059] 其中:

[0060] <Webapp_name>指定Web应用的标识符,该标识符对于创建该Web应用的模式是唯一的。

[0061] <version_name>指定Web应用版本的标识符。

[0062] <app_root>是对指向用户应用410的根的暂存区URL(stage URL)的引用。当用户应用运行时,该app_root下方的文件将对Web应用引擎408可用。尽管各版本可以在数据平台102内的相同暂存区中,仅由前缀分隔,但每个版本具有不同的暂存区以更好地管理特权和清理可能是有用的。

[0063] <file_name>是要作为Web应用引擎408的一部分运行的用户文件的路径。这是相对于<app_root>的。

[0064] <warehouse_name>是虚拟仓库的名称,例如要运行用户应用410的数据平台102的虚拟仓库1 302a。

[0065] <comment_string_literal>是描述Web用户应用410的注释。

[0066] 由安全管理器策略420和/或沙箱策略422针对用户应用410及其支持组件实施的特权的部分列表在表1和表2中进行了描述：

	特权	使用
[0067]	CREATE WEBAPP	在模式中创建用户应用 410 及其相关联的组件的能力。

[0068] 表1

	特权	使用
[0069]	USAGE	使得能够在默认版本上命中用户应用 410 的 HTTPS 端点。 使得能够使用 DESCRIBE 或 SHOW 命令看到 Web 应用
	MODIFY	在用户应用 410 上实现 ALTER 命令
	ALL [PRIVILEGES]	授予除 OWNERSHIP 之外的所有特权
	OWNERSHIP	授予对 Web 应用的完全控制；需要丢弃用户应用 410。 一次只能有一个角色对特定对象拥有此特权
	VIEW_ANY_VERSION	使用户能够命中与用户应用 410 相关联的任何非默认版本。这可能用于开发新版本，并让一些用户尝试预发布的版本。

[0070] 表2

[0071] 在一些示例中，存在用户应用410所依赖的数据平台102的对象，例如但不限于，用于存储文件的存储位置或暂存区，以及用户应用410被加载到其中的虚拟仓库，例如虚拟仓库1 302a。当创建用户应用410及其引用暂存区的相关联的组件时，用户应用410继承对该暂存区的READ（读取）特权和对虚拟仓库的USAGE（使用）特权。如果在创建了用户应用410之后，暂存区的特权被改变，使得用户应用410的所有者不再具有对该暂存区的特权，则对用户应用410的请求将失败，并带有声明用户应用410不存在的错误。如果在创建了用户应用410之后仓库的特权被改变，则使得仓库能够进行使用的逻辑将表现为好像没有设置仓库。在一些示例中，暂存区或数据位置被嵌入到用户应用410或用户应用的关联组件之一，并且对用户应用410的特权和暂存区的特权被关联在一起。在一些示例中，用户应用410及其相关组件可以根据存储在安全管理器策略420和/或沙箱策略422中的特权与其他所有者或用户共享。

[0072] 因此,当被实例化时,用户应用410及其所有支持组件继承由数据平台102提供的数据库内的第一类对象的所有属性,包括可以被数据平台102用来管理数据库对象的特权 and 限制。在一些示例中,用户应用410由UDF服务器406利用Web应用引擎408作为服务来提供,并且可以由Web应用浏览器运行时(runtime)组件404利用用于访问万维网上的文档和文件的协议通过网络(例如互联网)来访问,该Web应用浏览器运行时组件404包括在由客户端设备112托管的浏览器402中。

[0073] 如参考图2所描述的,计算服务管理器104实现安全策略,该安全策略验证来自执行平台110的所有通信,以验证该通信的内容和上下文与已知被分配给执行平台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不相关的特定数据源(例如,数据存储设备226或数据库存储装置106中的任何一个存储设备)。在一些示例中,执行节点424可能需要与第二执行节点通信,但是本文描述的安全机制可能不允许与第三执行节点通信。此外,可以(例如,在日志418或其他位置)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。

[0074] 在一些示例中,UDF服务器406及其组件(例如Web应用引擎408和用户应用410)以特定的编程语言(例如Python等)来实现。在一些示例中,Web应用浏览器运行时组件404以与UDF服务器406不同的编程语言(例如,C或C++)来实现,这可以通过使用不同的代码库(例如,没有相同潜在安全漏洞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。

[0075] UDF服务器406经由数据平台102的全局服务进程444接收来自Web应用浏览器运行时组件404的通信。全局服务进程444负责从Web应用浏览器运行时组件404接收请求。全局服务进程444使用计算服务管理器104的组件来执行各种认证任务,包括使用计算服务管理器104的访问管理器202来基于安全策略232进行第一级授权。UDF服务器406执行任务,包括分配处理线程以执行用户应用410的用户代码以及经由全局服务进程444将由用户应用410生成的结果返回到Web应用浏览器运行时组件404。

[0076] 在一些示例中,UDF服务器406在沙箱进程414内执行,如下文更全面描述的。在一些示例中,UDF服务器406用由解释器进程解释的Python来实现。在一些示例中,UDF服务器406用由虚拟机(JVM)执行的另一种语言(例如Java)来实现。由于UDF服务器406有利地相对于浏览器402在单独的进程中执行,因此对用户应用410的恶意操纵的风险较低。

[0077] 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418中以供查看和检索。在一些示例中,日志418可以本地存储在执行节点424处的存储器中,或者存储在诸如数据库存储装置106的单独位置。

[0078] 在一些示例中,安全管理器416可以通过抛出异常(例如,如果操作不被许可),或者如果操作被允许则返回(例如,不做任何事情)来阻止来自用户应用410的操作的完成。在一个实现中,安全管理器416被实现为安全管理器对象,其允许应用实现诸如安全管理器策略420的安全策略,并且使得应用能够在执行可能不安全或敏感的操作之前,确定该操作是什么以及是否正在允许该操作被执行的安全上下文中尝试该操作。安全管理器策略420可以被实现为具有UDF服务器406被授予的特权的文件。因此,UDF服务器406可以至少部分地基于安全策略来允许或不允许该操作。

[0079] 在一些示例中,沙箱进程414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)限制不可信应用的运行环境,来降低安全漏洞的风险。此外,在一些示例中,沙箱进程414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)以在沙箱环境内以安全的方式处理数据库查询或其他服务请求。

[0080] 在一些示例中,沙箱进程414可以利用虚拟网络连接,以便与计算环境400内的其他组件通信。可以针对计算环境400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服务器406配置虚拟网络连接的这种规则,以限制由UDF服务器406执行的操作可访问的位置(例如,互联网上的特定站点或UDF服务器406可以通信的组件)。因此,在该示例中,UDF服务器406可以被拒绝访问互联网上的特定网络位置或站点。

[0081] 沙箱进程414可以被理解成向沙箱内的一个进程(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。

[0082] 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414作为子进程执行,在一些示例中,与本身可以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。

[0083] 如进一步所示,沙箱进程414可以利用沙箱策略422来实施给定的安全策略。沙箱策略422可以是具有与沙箱进程414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)和特权的细节的文件。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)。在一些示例中,沙箱进程414限制UDF服务器406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。

[0084] Web应用浏览器运行时组件404为用户应用410提供前端。Web应用浏览器运行时组件404执行与用户应用410的数据平台102的浏览器交互。计算环境400的组件使用网络应用协议412进行通信,网络应用协议412提供用于用户应用410和浏览器402之间的交互的一组命令。网络应用协议412在逻辑上与用户应用410交互,并且在物理上穿过数据平台102的层,以确保在每一层上实施安全限制和策略。这些可以包括来自计算服务管理器104的许可或运行时要求。Web应用浏览器运行时组件404将由执行平台110处理并用一系列转发消息响应的消息发送回来。

[0085] Web应用引擎408包括可以由第三方定义但作为应用在执行平台110内运行的指令。Web应用引擎408提供编程框架,用户可以构建诸如用户应用410的应用。在一些示例中,Web应用引擎408用Python编写,并且被执行平台110视为特殊的Python存储过程。在一些示例中,Web应用引擎408以另一种语言(例如但不限于Java)编写,并且由执行平台110内的虚拟机托管。在一些示例中,第三方可以构建他们自己的Web应用引擎。

[0086] 用户应用410包括由最终用户编写并且由Web应用引擎408评估的应用。在一些示例中,用户应用410包括由专有Python解释器评估的Python文件。

[0087] UDF服务器406负责在诸如沙箱进程414的受控执行环境中运行UDF。在一些示例

中,UDF服务器406包括Python UDF服务器。在一些示例中,UDF服务器406利用其他语言,例如Java。

[0088] 在一些示例中,分配给UDF服务器406的统一资源定位器(URL)标识是在UDF服务器406的复制中稳定的唯一值。例如,URL标识是随机生成的字符串,该字符串在所有者的帐户中是唯一的。URL标识可以通过使用UUID4和Base64对它进行编码来创建,以给它一个更简洁的表示。

[0089] 在一些示例中,数据平台102的模式对象用于定义这样的组件,该组件定义诸如但不限于UDF服务器406、Web应用引擎408、用户应用410和Web应用浏览器运行时组件404的web应用的组件。名称、网络端点、特权和策略都基于该对象。在一些示例中,模式对象包括要使用的Web应用引擎408的特定版本以及任何资源约束。

[0090] 在一些示例中,用户代码的版本被指定,并且将与web应用的命名版本相关联,该命名版本指代数据平台102用来运行用户代码的暂存区上的一个地方。

[0091] 图4B、图4C和图4D是根据一些示例的,用于提供Web应用作为第一类数据库对象的计算环境400的交互和数据流图。

[0092] 计算环境400利用行集操作符(Row Set Operators,RSO),行集操作符在执行平台110中作为程序的一部分运行。RSOI是在执行平台110的处理线程上操作的RSO的实例。RSOI扩展函数是运行存储过程的RSOI。在一些示例中,存储过程是用Python编写的。在一些示例中,存储过程是用Java编写的。

[0093] RSOI扩展函数436的所有者设置对RSOI扩展函数436本身的许可,定义谁可以使用它,以及设置用于运行它的不同角色。设置这些许可是在用户定义RSOI扩展函数436时发生,并且实施是在浏览器访问与Web应用相关联的URL时,在Web应用资源442中发生。Web应用资源442还将起动作业,该作业设置其许可上下文。RSOI扩展函数436在该许可/会话上下文中操作。在一些示例中,所有者设置对将被实例化的每个对象的许可,这些对象例如是但不限于,UDF服务器406、Web应用引擎408和用户应用410。

[0094] 在操作1中,由客户端设备112托管的浏览器402使用到Web应用资源442的web套接字连接来与由数据平台102托管的用户应用410通信。数据平台102的计算服务管理器104的访问管理器202基于存储在数据存储设备206上的一组安全策略232授权对用户应用410的访问。当被授权时,数据平台102向使用浏览器402的用户提供用户应用410。Web应用浏览器运行时组件404从web套接字拉回消息并发出适当的命令。

[0095] 如果不存在具有Web应用引擎408的实例的当前会话,则Web应用浏览器运行时组件404使用为用户应用410定义的安全上下文来验证请求具有使用应用的许可,然后请求UDF服务器406开始作业。初始执行计划为作业启动Web应用引擎408的实例。Web应用引擎408将基于分别由安全管理器416和沙箱进程414实施的安全管理器策略420和沙箱策略422来实例化,并且事件被记录到日志418中。在Web应用引擎408的会话起动之后,Web应用引擎408向查询协调器430发送命令。

[0096] 在操作2中,作业具有与作业相关联的查询协调器430。从这一点开始,与Web应用引擎408的通信通过向查询协调器430添加“应用交互”类型的查询协调器事件来发生。这将导致运行请求排队。查询协调器事件还具有对流的引用,它可以通过该流发送转发消息以到达浏览器402。这样,查询协调器430可以获得响应事件并将它们直接传回浏览器402。

[0097] 在操作3中,提供应用请求队列426。应用请求队列426是返回消息被推送到其上的存储器内队列。在一些示例中,应用请求队列426在存储器中,以确保操作4处的连接总是返回到与查询协调器430所在的相同的全局服务实例。在一些示例中,在全局服务失败的情况下,允许丢失消息并使Web应用浏览器运行时组件404通过新的运行请求重新建立状态。

[0098] 在操作4中,RSOI扩展函数436启动长时间运行的特殊存储过程。该存储过程在安全上下文中运行,该安全上下文是基于安全管理器策略420和沙箱策略422为将在执行节点424上实例化的对象配置的。存储过程创建UDF服务器406,该UDF服务器406在执行平台110的执行节点424的锁定环境中安全地运行脚本。RSOI扩展函数436使用RPC调用来通过UDF服务器406基于存储过程启动Web应用引擎408。RSOI扩展函数436用函数信息来调用“执行过程”,该函数信息将告知存储过程Web应用引擎408不会终止。基于分别由安全管理器416和沙箱进程414实施的安全管理器策略420和沙箱策略422在验证过程期间实例化Web应用引擎408,并且事件被记录到日志418中。Web应用引擎408的过程连接到UDF服务器406上的流应用请求RPC端点,并发出初始化应用消息。该消息将用于用适当的策略和文件信息引导Web应用引擎408,以运行用户应用410。该信息作为启动用户应用410的操作1的执行计划的一部分传过来。RSOI扩展函数436连接到执行平台资源432中的Web应用交互信道434端点,并处理通过该信道进入的消息。然后,UDF服务器406能够使用特殊存储过程来运行Web应用引擎408和用户应用410,来以低等待时间和高效的方式代理来自浏览器的通信。

[0099] 在一些示例中,网络端点基于基于帐户定位器的URL,从而为每个所有者的帐户提供域以充当浏览器安全边界。在其他示例中,URL标识符的组件是唯一的且不可猜测的数字。URL标识在重命名和复制到其他帐户时是稳定的。浏览器402的用户可以直接访问URL。为此,他们将被要求登录到数据平台102,并且他们将需要对用户应用410的使用特权。

[0100] 来自执行平台资源432的消息是通过使用或编辑应用得到的用户驱动的交互。去往UDF服务器406的消息在定义UDF应用请求的文档中被定义。对于应用用户请求,RSOI扩展函数436:1.将消息请求代理回到适当的UDF服务器406,并让它们由Web应用引擎408处理;2.更新文件消息请求命令RSOI扩展函数436:a.查找需要更新的文件;b.发出更新文件开始命令;c.遵循更新由Web应用引擎408所使用的适当文件所需的更新文件命令;以及d.生成更新文件结束消息。安全管理器416和沙箱进程414分别使用安全管理器策略420和沙箱策略422来验证用户应用410使用Web应用引擎408对执行平台110的所有访问,例如但不限于存储在数据库存储装置106中的数据以及由执行平台110执行的附加函数和过程。这允许执行平台110通过用户应用410向浏览器402提供服务,而不需要数据与执行平台110一起在安全位置和不安全位置之间移动,从而增强了可扩展性和安全性。在一些示例中,对于“执行平台资源(execution platform resource)”命令的HTTP信道,使用ContentType=application/octet-stream,在两个方向上使用Protobuf协议。在一些示例中,Protobuf协议以编码的方式在两个方向上使用,这将允许沿着流向下推送多个消息,而不必关闭TCP连接并重新发出请求。

[0101] 在操作5中,参考图4D,UDF服务器406管理Web应用引擎408的生命周期。UDF服务器406启动并随后管理通过流应用请求端点传入的请求。参考图4D,在一些示例中,存在两种基本类型的消息:应用用户请求440和应用控制平面请求438。应用用户请求440被直接传递到Web应用引擎408的运行方法中。应用控制平面请求438被定向到UDF服务器406以进行一

些系统操作,例如但不限于更新文件,初始化应用或关闭事物。

[0102] 在操作6中,Web应用过程包括被调用的附加生命周期函数。代码段中部分地示出了示例web应用过程:

```
//当 InitializeApp 消息传入时调用

def start(webApp: WebApp, config: ConfigurationParameters)

//用于处理 app 请求的主函数。

//针对每个传入的 BackMsg 调用

def run(webApp: WebApp, message: BackMsg, responseQueue

[0103] queue<ForwardMsg>, sessionCtx: SessionCtx)

//文件被更新时调用

def beforeFilesChanged(webApp: WebApp, files: List<Files>)

def afterFilesChanged(webApp: WebApp, files: List<Files>)

//在 app 结束之前调用

def stop(webApp: WebApp)
```

[0104] 在一些示例中,通过在存储过程数据持久性对象中进行指定,并作为起动Web应用引擎408的一部分而传递,UDF服务器406知道哪些函数与哪些操作相关联。在一些示例中,UDF服务器406知道哪些函数与哪些操作相关联,因为存储过程DP0具有作为起始函数的处理程序(handler),并且起始函数的返回类型返回指定其他函数的函数表。在一些示例中,UDF服务器406知道哪些函数与哪些操作相关联,因为存储过程DP0具有作为起始函数的处理程序,并且提供了UDF服务器406可以进行寻找以发现其他函数的注释。在一些示例中,存储过程的新属性将存储过程标记为Web应用引擎408。当设置了该项时,处理程序调用一个函数,该函数返回映射到上面不同应用生命周期事件(例如,运行、文件更改等)的函数表。

[0105] 在操作7中,来自UDF服务器406的响应通过执行平台资源返回。当用户应用410将消息添加到应用响应队列428时,UDF服务器406将获取这些响应并将它们通过RPC端点传递回RSOI扩展函数436,然后RSOI扩展函数436将它们沿着长轮询HTTP连接发送到执行平台资源中的web应用交互信道(web-app-interaction-channel)。执行平台资源将响应放入应用响应队列428中,并通知查询协调器430。

[0106] 在操作8中,查询协调器430将响应发送回浏览器402。查询协调器430拾取事件,过滤出违反策略的任何消息(例如,无限制的JS或HTML)。因为当查询协调器430获得查询协调器事件时,它已经被提供了响应信道,所以它使用该响应信道来发回响应。在一些示例中,当确保web套接字与查询协调器430处于相同的全局服务中时,执行平台资源发送请求本

身。在一些示例中,当不能确保web套接字与查询协调器430在相同的全局服务中时,查询协调器430执行发回响应的操作,因为已经存在为查询协调器430找到正确的全局服务实例的功能。

[0107] 图5是根据本公开的一些示例的数据平台102的计算环境500的部署图,该计算环境500为用户应用410的开发和使用提供统一安全上下文502。计算环境500支持用于数据平台102的组件的统一安全上下文502,数据平台102的组件提供用于开发源代码的软件开发环境,源代码诸如是用户应用(例如用户应用410)的用户应用源代码512。作为统一安全上下文502的一部分,计算环境500还支持用于提供用户应用410作为Web应用的一部分的生产环境,该Web应用允许用户访问数据平台102的资源,诸如存储在数据库存储装置106中的数据库对象510。

[0108] 访问管理器202基于安全策略506中指定的特权授权对统一安全上下文502的组件和对象的访问,安全策略506包括用于统一安全上下文502内的组件和对象的一个或多个安全策略。安全策略506包括但不限于:应用于存储在数据平台102的暂存区上的用户应用源代码512和存储在数据库存储装置106中的数据库对象510的安全策略,以及应用于用户应用410的使用和共享的安全策略。安全策略506还包括由安全管理器416实施的安全管理器策略420和由沙箱进程414实施的沙箱策略422,如参考图4A、图4B、图4C和图4D所描述的。

[0109] 计算环境500包括用户用来访问存储在数据库存储装置106中的数据库对象510的数据资源管理器(data explorer)228。数据资源管理器228包括可视化器508,用户可以使用可视化器508来访问和显示数据库对象510的数据。数据资源管理器228还包括编辑器,用户可以使用该编辑器来创建、编辑和/或存储用于访问和修改数据库对象510的数据的数据库查询和软件源代码,诸如用户应用410的用户应用源代码512。访问管理器202使用安全策略506中的一个或多个安全策略来确定用户是否被授权使用数据资源管理器228来访问统一安全上下文502所涉及的数据平台102的组件或数据库对象。这些用户包括但不限于:作为用户应用410和/或数据库对象510的所有者的数据平台102的用户、所有者的同一帐户内的用户、不同于所有者帐户的另一帐户的用户、数据平台102之外的外部系统等。

[0110] 计算环境500还包括应用和数据管理器230,该应用和数据管理器230允许用户指定哪些其他实体可以共享用户应用410的使用并且可以访问和/或修改数据库对象510的数据。访问管理器202使用安全策略506中的一个或多个安全策略来确定用户是否被授权使用应用和数据管理器230来访问安全策略506,以便设置安全策略,该安全策略允许另一用户访问统一安全上下文502所涉及的数据平台102的组件或数据库对象。这些用户包括但不限于:作为用户应用410和/或数据库对象510的所有者的数据平台102的用户、所有者的同一帐户内的用户、不同于所有者帐户的另一帐户的用户、数据平台102之外的外部系统等。

[0111] 沙箱进程414、安全管理器416、UDF服务器406和Web应用引擎408进行操作来为用户应用410提供安全的执行环境,如本文参考图4A、图4B、图4C和图4D所描述的。沙箱进程414和安全管理器416使用安全策略506中包括的它们各自的安全策略来提供用户使用浏览器402对用户应用410的访问,该浏览器402具有Web应用浏览器运行时组件404,并且由客户端设备112托管,如本文参考图4A、图4B、图4C和图4D所描述的。

[0112] 在一些示例中,访问管理器202使用安全策略506中的一个或多个安全策略来确定是否允许用户使用由提供外部编辑器(未示出)的Web应用提供的编辑器来访问用户应用

源代码512。外部编辑器是根据图4A、图4B、图4C和图4D中描述的方法提供的。在一些示例中,Web应用的用户应用包括由执行平台110执行的编辑器,编辑器的UI由被客户端设备托管的浏览器的Web应用浏览器运行时组件提供。在附加示例中,Web应用的用户应用包括由执行平台110执行的代理服务器,并且客户端设备的浏览器的Web应用浏览器运行时组件包括通过充当代理服务器的用户应用访问用户应用源代码512的编辑器。

[0113] 图6是示出根据本公开的一些示例的由数据平台102在统一安全上下文502中开发和部署用户应用410的方法600的活动图。

[0114] 在操作602中,数据平台102向第一用户614提供编辑器504,第一用户614基于安全策略506使用该编辑器504来创建、编辑和/或存储用户应用410的用户应用源代码512。第一用户614使用编辑器504对用户应用源代码512的访问由访问管理器202确定,访问管理器202基于安全策略506授权第一用户614使用编辑器在数据平台102的数据库存储装置106上创建、编辑和/或存储用户应用源代码512。

[0115] 在操作604中,第一用户614使用应用和数据管理器230通过设置存储在安全策略506中的特权来设置数据库对象510和用户应用410的共享和使用特权。用户使用应用和数据管理器230对安全策略506的访问由访问管理器202确定,访问管理器202基于安全策略506授权第一用户614创建、编辑和存储共享和使用特权。这些特权允许授权附加用户和系统,例如第二用户616,使用用户应用410来访问数据库对象510的数据和数据平台102的其他组件。共享和使用特权存储在安全策略506中。

[0116] 在操作606中,数据平台102部署用户应用410,从而使得用户应用410可用于一个或多个用户,例如第二用户616。

[0117] 在操作608中,数据平台102从第二用户616接收请求,以访问位于数据平台102上的组件和/或数据库对象的数据。作为响应,数据平台102基于安全策略506向第二用户616提供用户应用410。向第二用户616提供用户应用410由访问管理器202确定,访问管理器202基于安全策略506授权第二用户616使用用户应用410来访问数据平台102的组件和数据库对象510。

[0118] 在操作610中,用户应用410基于安全策略506访问数据平台102的组件和数据库对象510的数据。用户应用410与数据平台102的组件和数据库对象510的交互部分地由安全管理器416基于安全策略506中的安全策略420授权。此外,用户应用410与数据平台102的组件和数据库对象的交互部分地由沙箱进程414基于安全策略506中包括的沙箱策略422授权。

[0119] 在操作612中,用户应用410向第二用户616提供数据库对象510的数据。经由用户应用410向第二用户616提供数据库对象510的数据由访问管理器202确定,访问管理器202基于安全策略506授权第二用户616使用用户应用410来访问数据平台102的组件和数据库对象510。

[0120] 在一些示例中,使用由Web应用提供的编辑器(未示出)来访问和编辑用户应用源代码512。

[0121] 在一些示例中,用户应用410允许外部系统访问数据平台102。例如,数据平台102的用户应用充当代理服务器,允许外部系统访问数据平台的组件和数据库对象510的数据。

[0122] 图7示出了根据示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702(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702可以使机器700执行本文描述的方法中的任何一种或更多种方法的任何一个或更多个操作。以这种方式,指令702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104、执行平台110和数据库存储装置106的数据存储设备1至N),该特定的机器700被具体配置成以本文描述的方式执行所描述和示出的功能中的任何一个。

[0123] 在替代示例中,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-客户端网络环境中作为服务器机器或客户端机器来操作,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、网络桥接器或能够顺序地或以其它方式执行指令702的任何机器,指令702指定由机器700采取的动作(action)。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702以实施本文所讨论方法中的任何一种或更多种的机器的集合。

[0124] 机器700包括处理器704、存储器706和I/O组件708,它们被配置成例如经由总线710彼此通信。在一些示例中,处理器704(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,如由可以执行指令702的处理器712和处理器714所示的多个处理器。术语“处理器”旨在包括多核处理器,多核处理器可以包括可以同时执行指令702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704,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。

[0125] 存储器706可以包括主存储器732、静态存储器716和包括机器存储介质734的存储单元718,所有这些都可由处理器704例如经由总线710访问。主存储器732、静态存储器716和存储单元718存储指令702,指令702体现本文描述的方法或功能中任何一种或更多种。在由机器700执行期间,指令702也可以全部或部分地驻留在主存储器732内、静态存储器716内、存储单元718内、至少一个处理器704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。

[0126] 输入/输出(I/O)组件708包括用于接收输入、提供输出、产生输出、传输信息、交换信息、捕获测量值等的组件。包含在特定机器700中的具体的I/O组件708将取决于机器的类型。例如,诸如移动电话的便携式机器将可能包括触摸输入设备或其他这样的输入机制,而无头服务器机器可能不包括这样的触摸输入设备。将认识到的是,I/O组件708可以包括图7中未示出的许多其他组件。I/O组件708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。在各种示例中,I/O组件708可以包括输出组件720和输入组件722。输出组件720可以包括视觉组件(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)、声学组件(例如,扬

声器)、其他信号发生器等。输入组件722可以包括字母数字输入组件(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组件)、基于点的输入组件(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)、触觉输入组件(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组件)、音频输入组件(例如,麦克风)等。

[0127] 通信可以使用多种技术来实现。I/O组件708可以包括通信组件724,通信组件724可操作来分别经由耦合730和耦合728将机器700耦合到网络736或设备726。例如,通信组件724可以包括网络接口组件或与网络736接口的另一合适的设备。在进一步的示例中,通信组件724可以包括有线通信组件、无线通信组件、蜂窝通信组件和经由其他模态提供通信的其他通信组件。设备726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104、执行平台110中的任何一个,并且设备726可以包括数据存储设备226或本文描述的与数据平台102或数据库存储装置106通信的任何其他计算设备。

[0128] 各种存储器(例如,706、716、732和/或处理器704和/或存储单元718的存储器)可以存储指令702和数据结构(例如,软件)的一个或更多个集合,这些指令和数据结构体现本文所描述方法或功能中的任何一种或更多种或被本文所描述方法或功能中的任何一种或更多种所利用。当由处理器704执行时,这些指令702促使各种操作来实现所公开的示例。

[0129] 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包括处理器内部或外部的存储器的固态存储器以及光和磁介质。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,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、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)。

[0130] 在各种示例中,网络736的一个或更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736或网络736的一部分可以包括无线或蜂窝网络,并且耦合730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730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(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、第五代无线(5G)网络、通用移动通信系统(UMTS)、高速分组接入(HSPA)、微波接入全球互操作性(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。

[0131] 指令702可以使用传输介质经由网络接口设备(例如,包括在通信组件724中的网络接口组件)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736上传输或接收。类似地,指令702可以使用传输介质经由耦合728(例如,对等耦合)传输到设备726或被接收。术语“传输介质”和“信号介质”意思相同,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、编码或携带由机器700执行的指令702的任何无形介质,并且包括数字或模拟通信信号或便于这种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调制的数据信号、载波等。术语“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其特性中的一个或更多个特性。

[0132] 本文所述的示例方法的各种操作可以至少部分地由(例如,通过软件)临时配置或永久地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,本文公开的方法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分布在一个或更多个处理器中,不仅驻留在单个机器内,而且部署在多个机器上。在一些示例中,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(server farm)内)中,而在其他示例中,处理器可以分布在多个位置上。

[0133] 尽管已经参考具体示例描述了本公开的示例,但明显的是,可以对这些示例做出各种修改和改变,而不偏离本发明主题的更宽的范围。相应地,说明书和附图被认为是说明性的而不是限制性的。形成其一部分的附图通过说明方式并且不以限制方式示出了其中可以实施主题的具体示例。所示出的示例以充分细节描述,以便使本领域那些技术人员能够实施本文公开的教导。可以使用其他示例和从其中衍生的示例,使得可以作出结构性或逻辑性替换和变化而不偏离本公开的范围。因此不以限制性意义理解这一详细说明,并且各种示例的范围仅由所附权利要求连同授权的此类权利要求的完整范围的等同物限定。

[0134] 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,除非另有说明,否则术语“或”用于指非排他性的或,因此“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“包含(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”和“包含”是开放式的;也就是说,包括除了权利要求中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。

[0135] 本发明主题的此类示例可以在本文中单独地和/或共同地由术语“示例”指代,该术语仅出于便利目的并且不意在自愿将本申请的范围限于任何单一发明或发明构思,如果实际上公开的多于一种的话。因此,尽管本文示出并描述了具体示例,但是应当理解,可以用经计算以实现相同目的的任何布置替换用于所示的具体示例。本公开旨在覆盖各种示例的任何和所有修改或变型。对于本领域的技术人员来说,在阅读了以上描述之后,上述示例的组合以及本文没有具体描述的其他示例将是明显的。

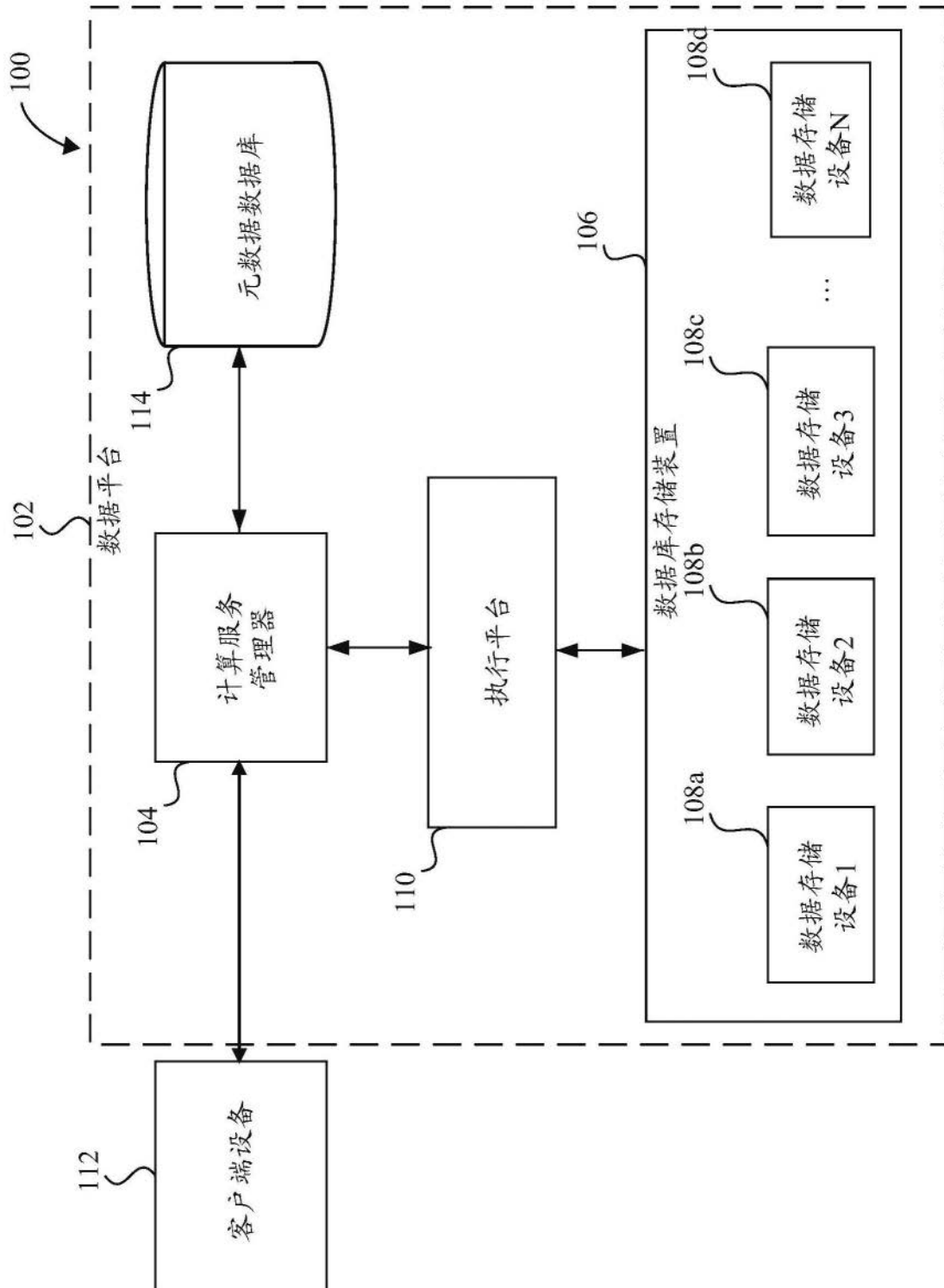


图1

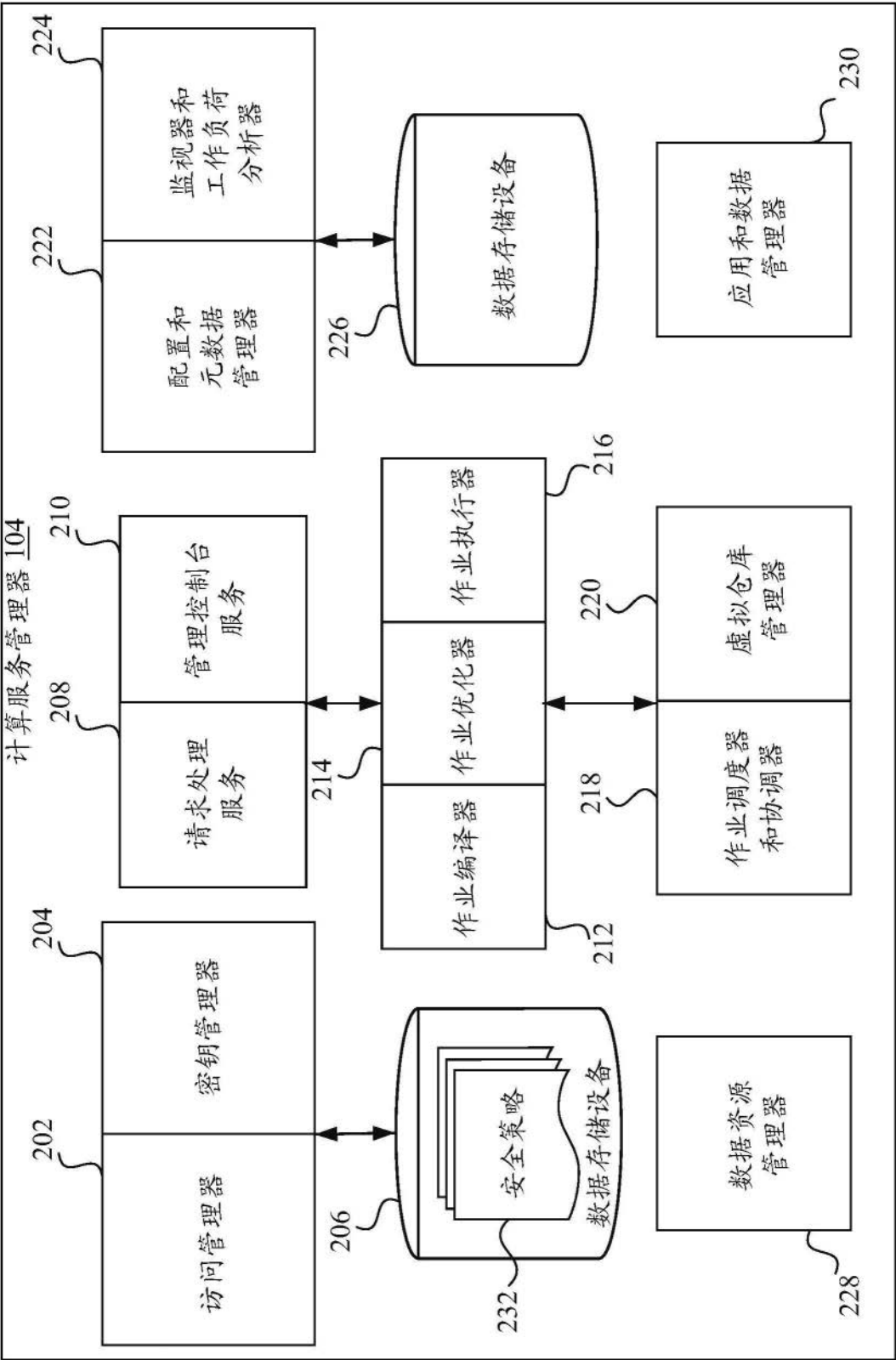


图2

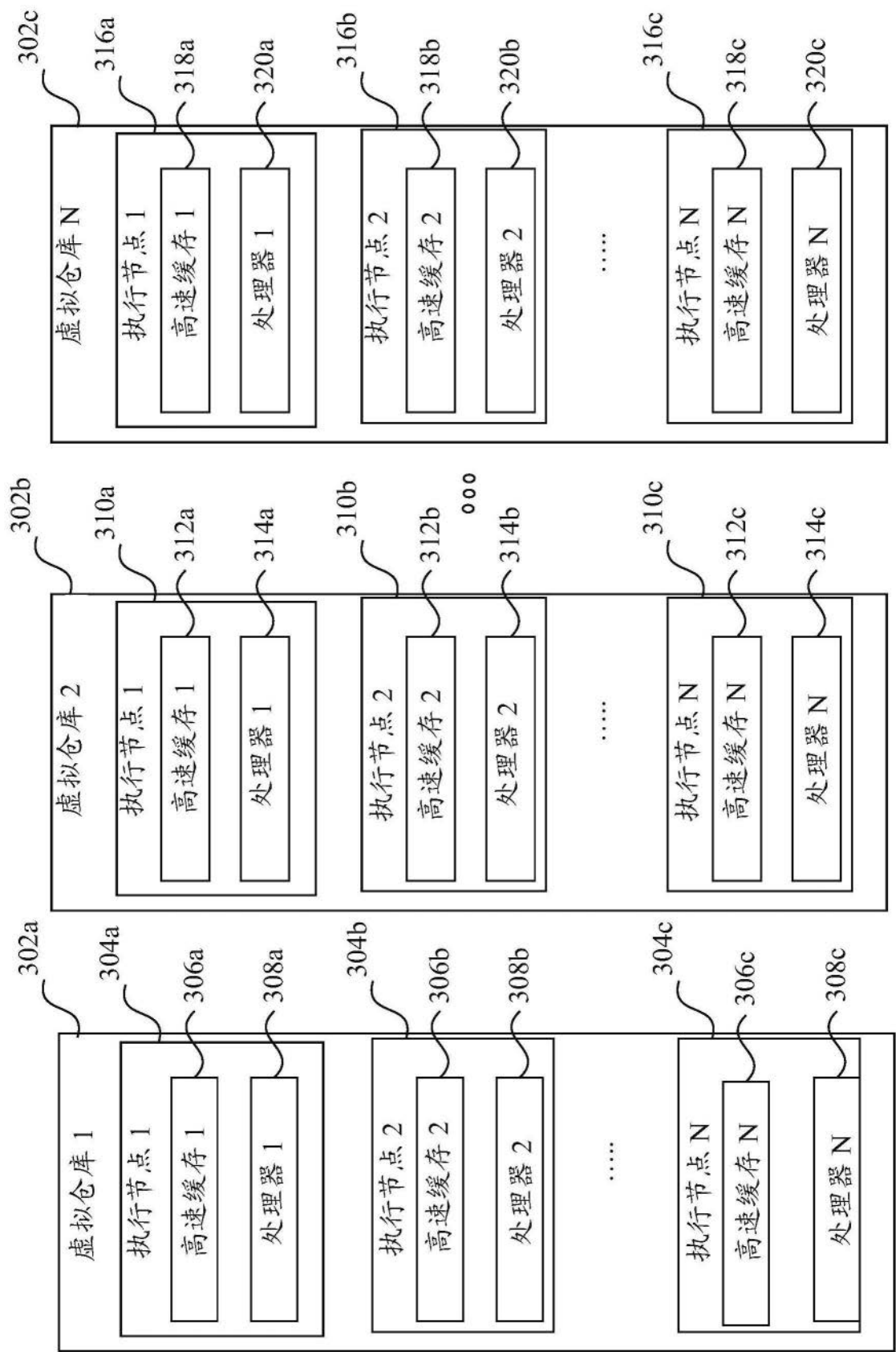


图3

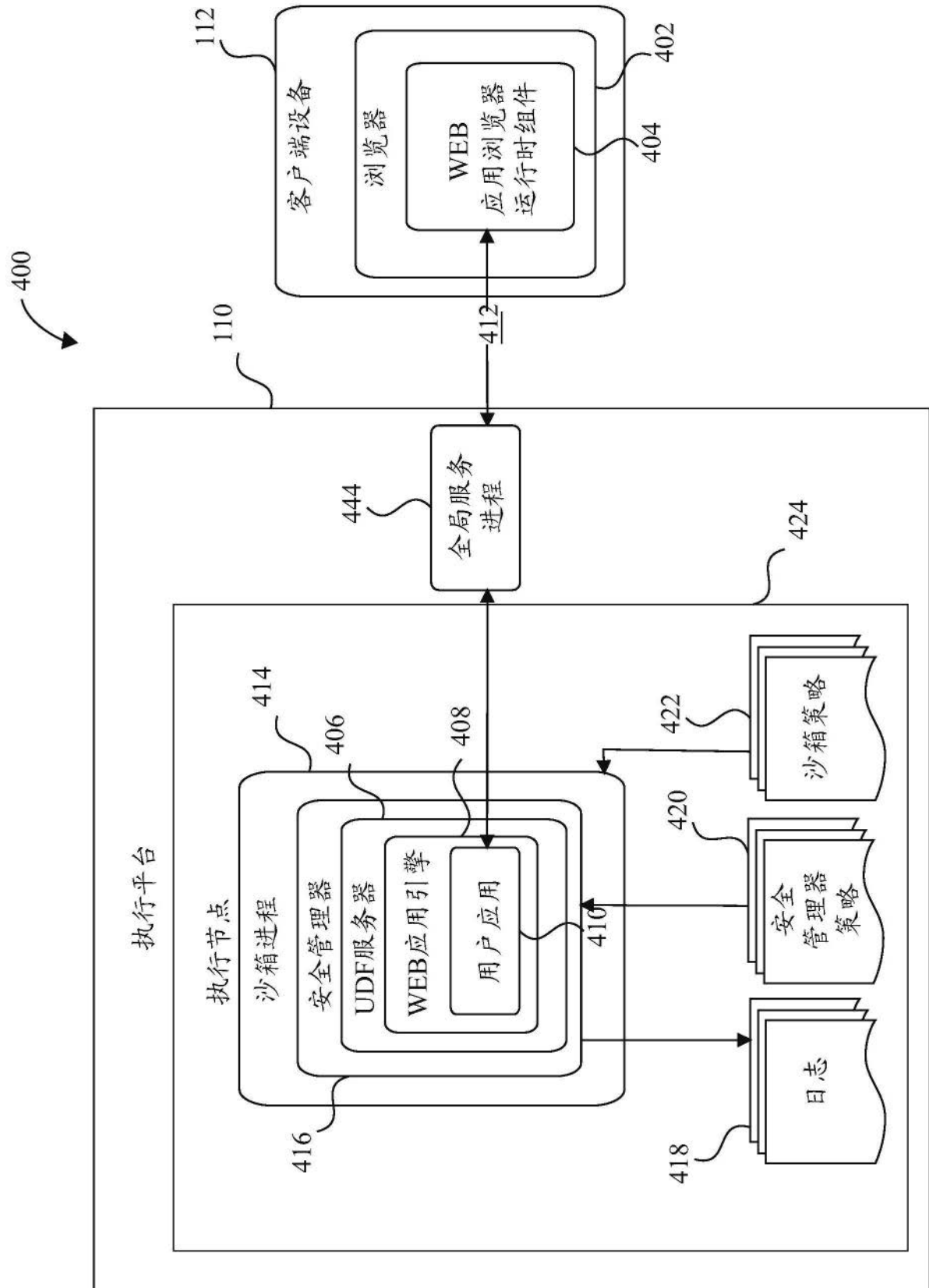


图4A

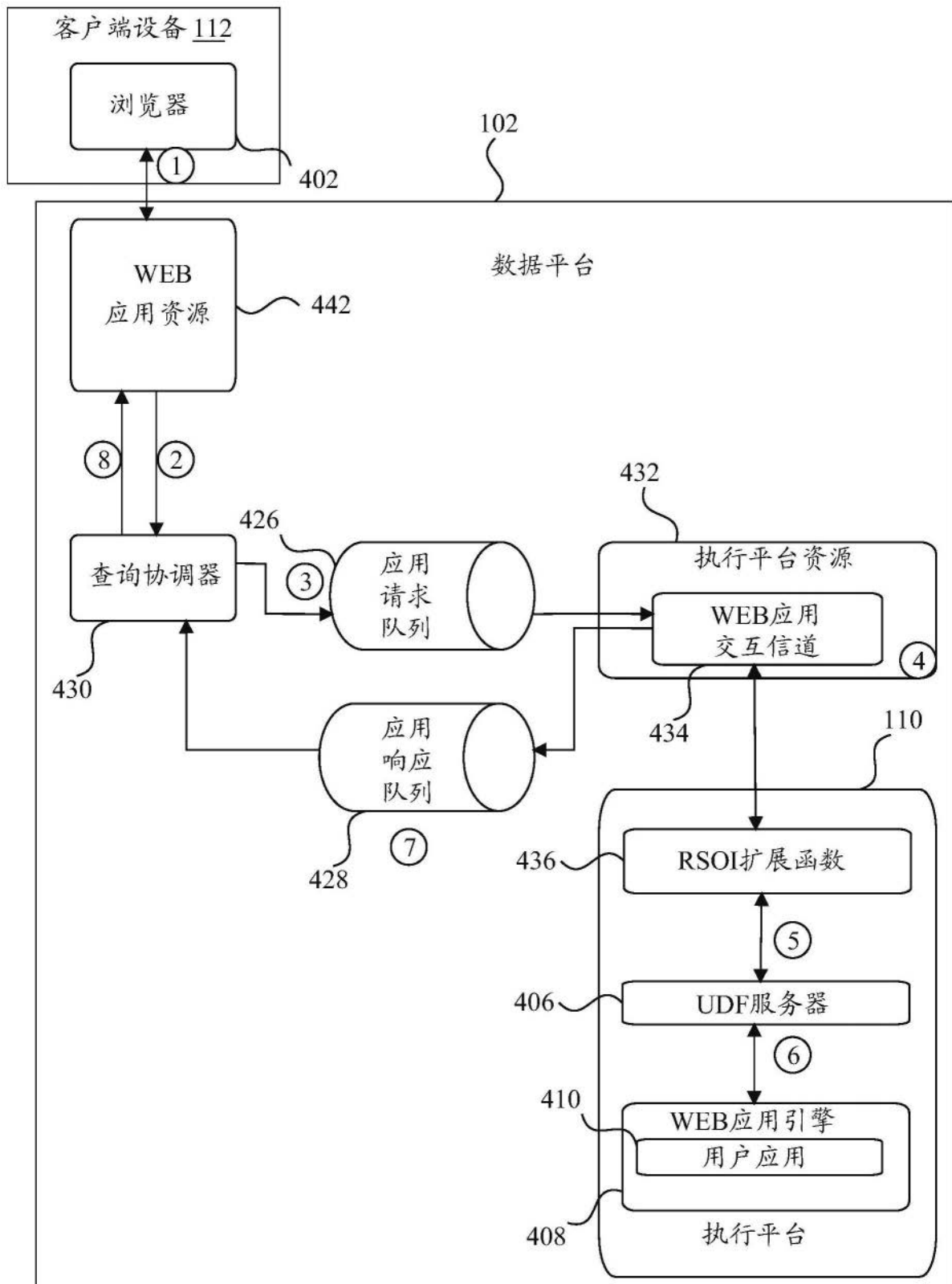


图4B

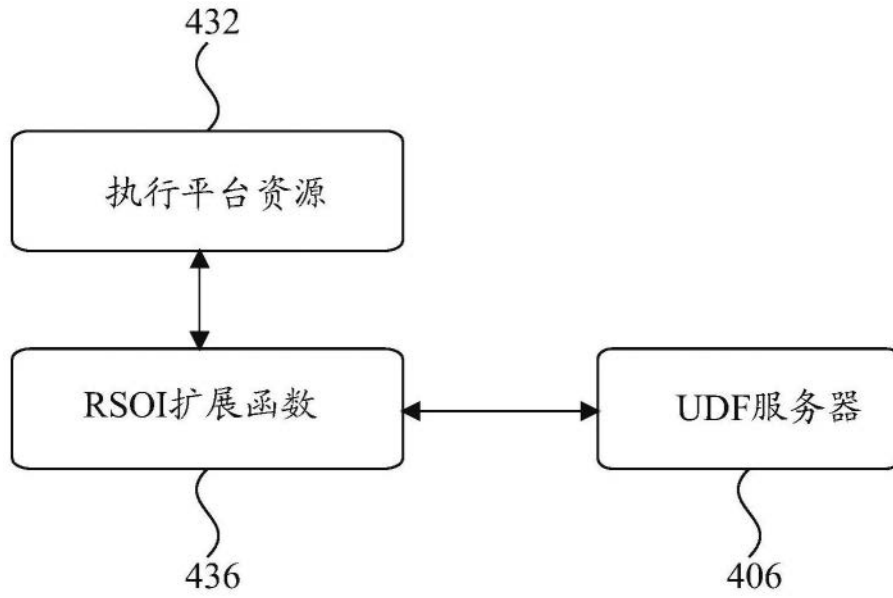


图4C

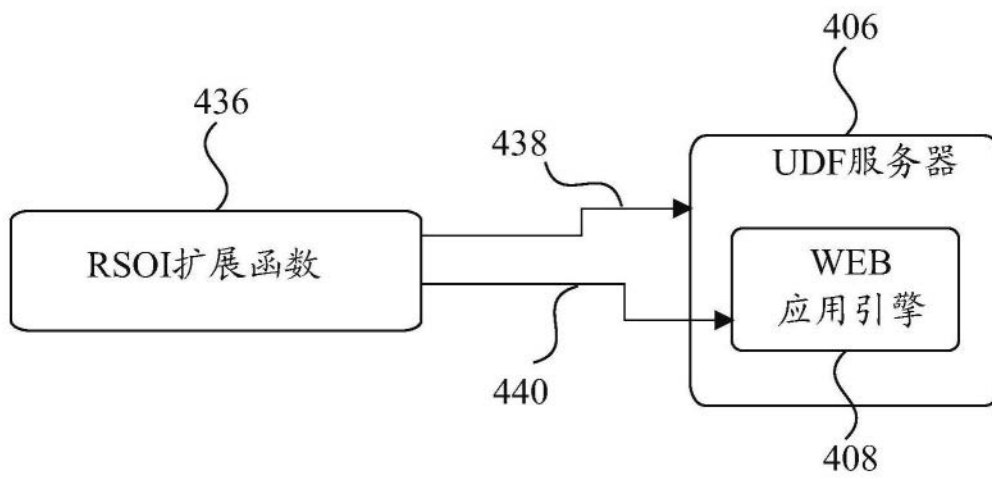


图4D

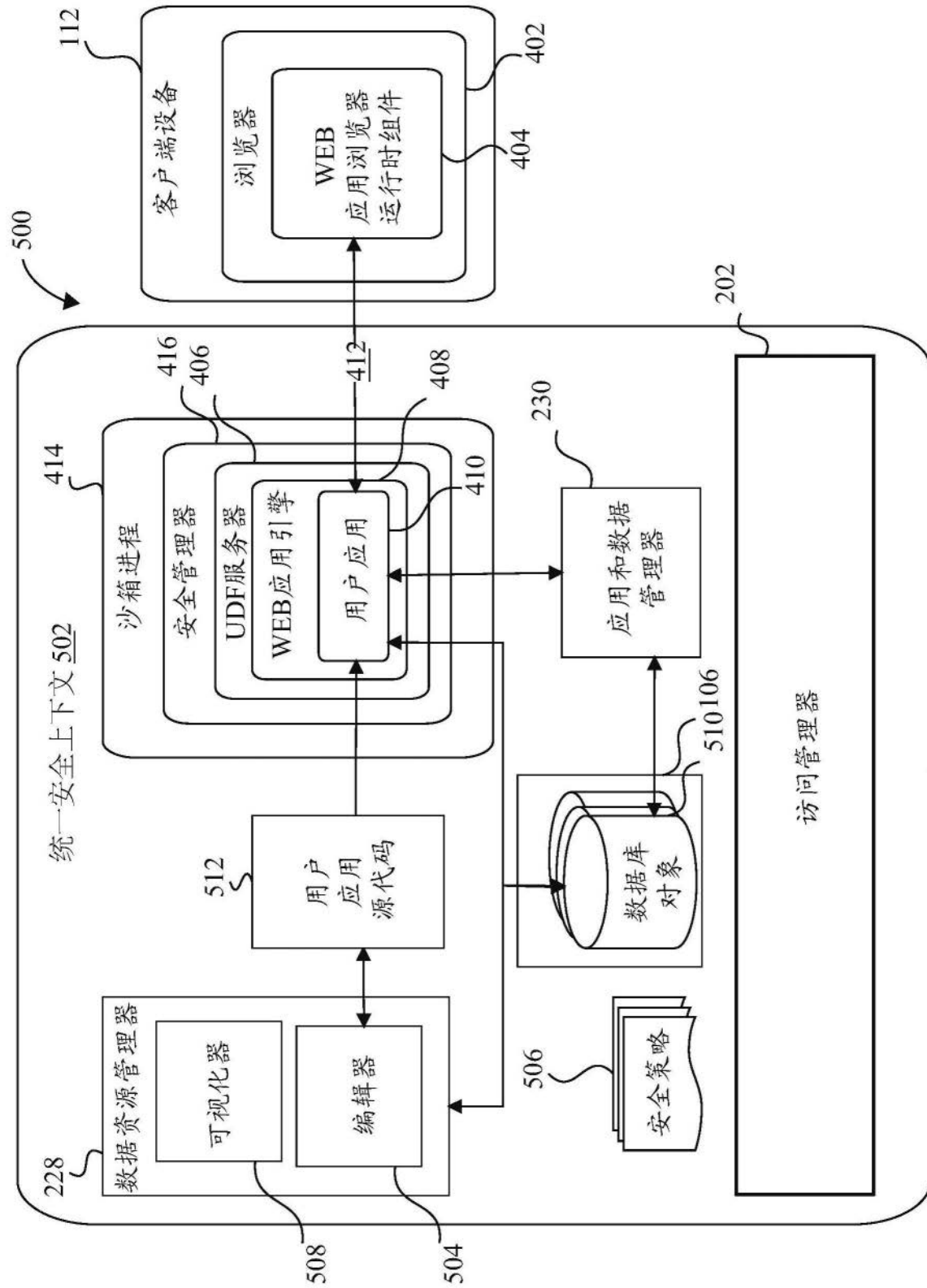


图5

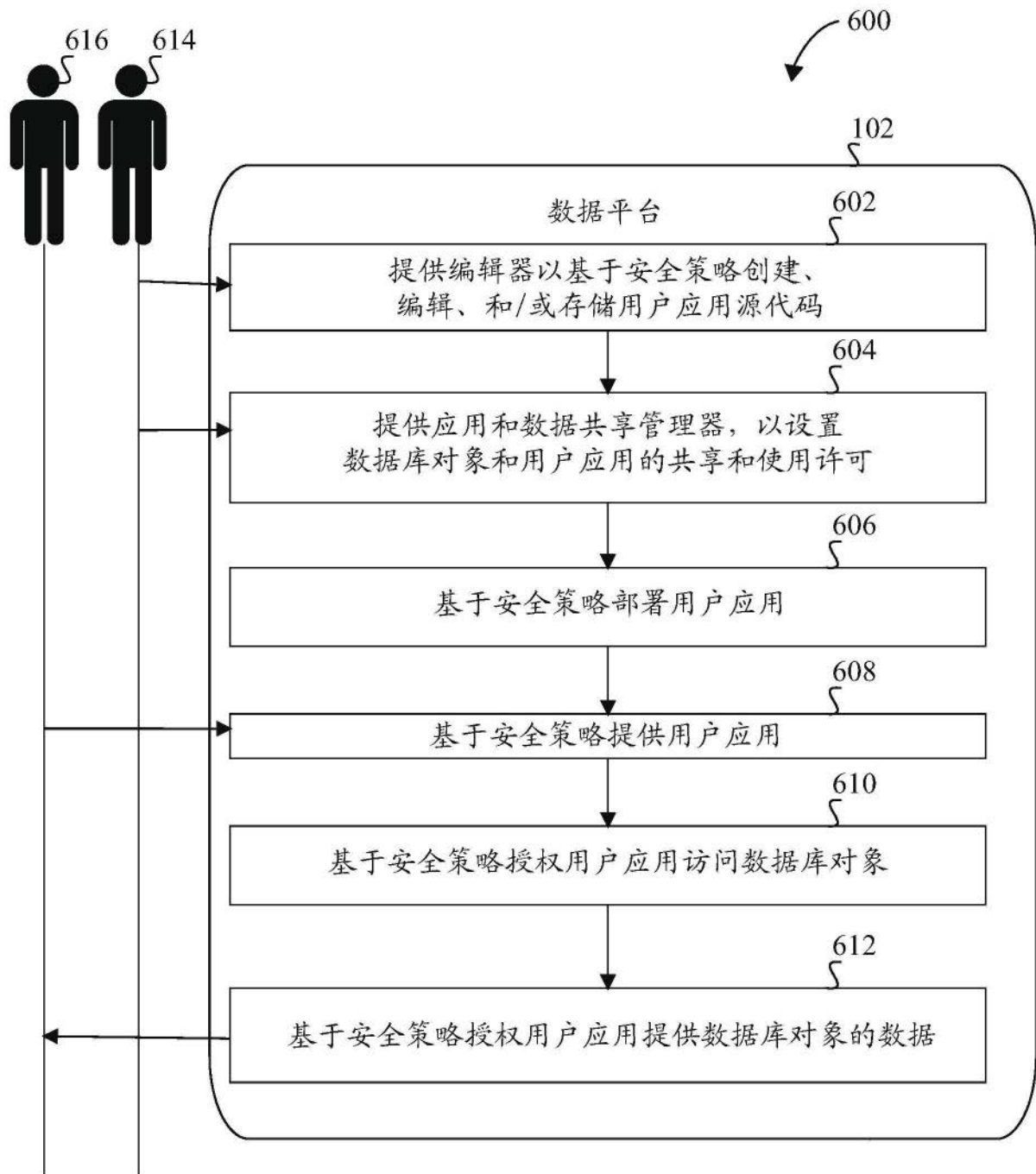


图6

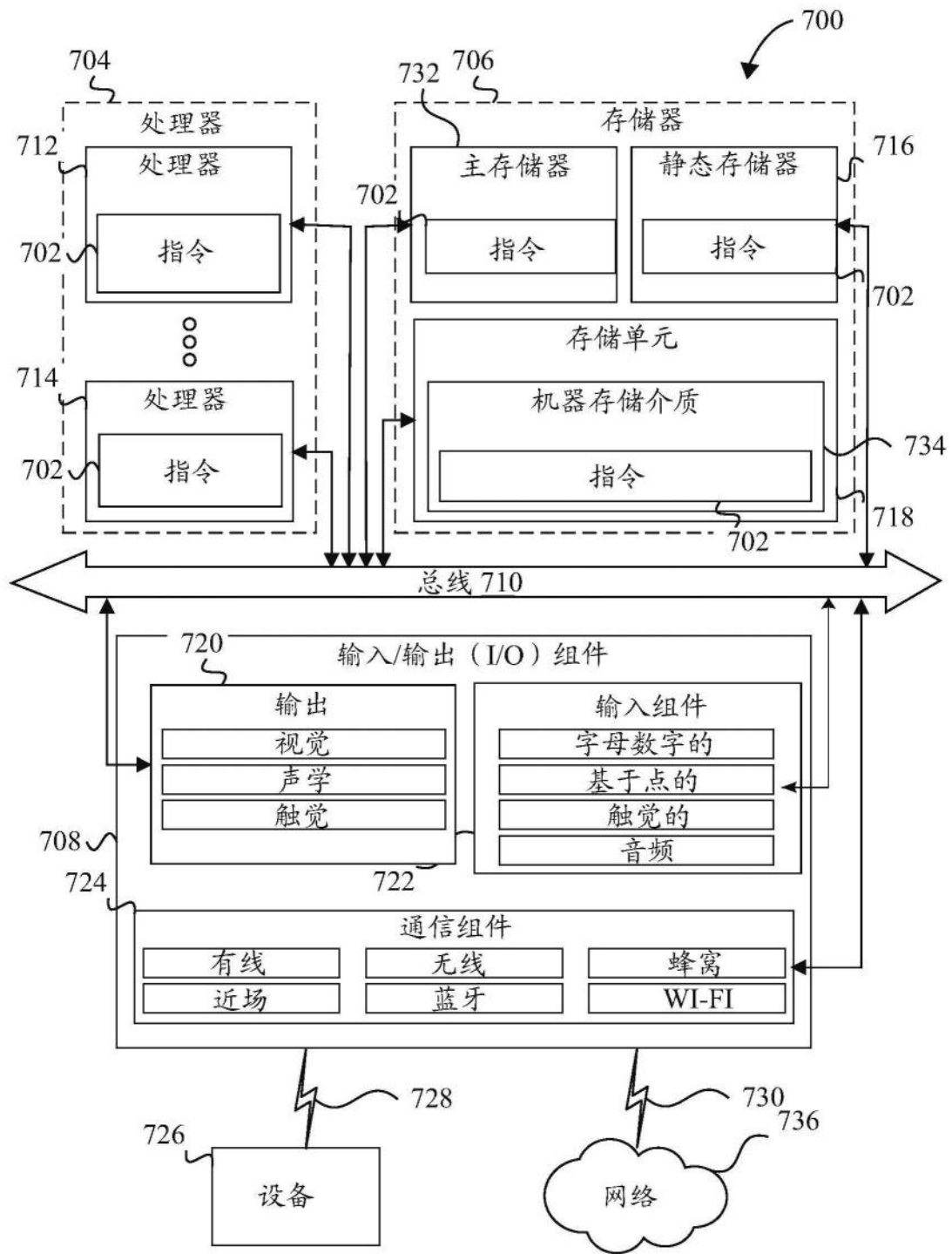


图7