



(12) 发明专利

(10) 授权公告号 CN 110023901 B

(45) 授权公告日 2023.08.11

(21) 申请号 201880004673.0

D·娜拉亚楠

(22) 申请日 2018.09.21

(74) 专利代理机构 中国贸促会专利商标事务所  
有限公司 11038

(65) 同一申请的已公布的文献号  
申请公布号 CN 110023901 A

专利代理师 邹丹

(43) 申请公布日 2019.07.16

(51) Int.Cl.

(30) 优先权数据

G06F 8/656 (2006.01)

62/561,599 2017.09.21 US

G06F 8/658 (2006.01)

(85) PCT国际申请进入国家阶段日  
2019.05.30

(56) 对比文件

(86) PCT国际申请的申请数据

CN 106815071 A, 2017.06.09

PCT/US2018/052131 2018.09.21

CN 104487948 A, 2015.04.01

(87) PCT国际申请的公布数据

W02019/060663 EN 2019.03.28

CN 105516233 A, 2016.04.20

CN 104321745 A, 2015.01.28

CN 103617269 A, 2014.03.05

CN 101425017 A, 2009.05.06

WO 2006125392 A1, 2006.11.30

(73) 专利权人 甲骨文国际公司

何俊等. 数据中心内网核心业务运行环境升级改造实践.《国土资源信息化》.2014,(第01期),

地址 美国加利福尼亚

审查员 彭凤鸣

(72) 发明人 N·科里斯纳帕 B·纳拉亚南

A·A·苏嘉达

M·X·C·阿尔东卡尔

权利要求书3页 说明书22页 附图15页

(54) 发明名称

用于更新基于云的多层应用栈的系统和方法

(57) 摘要

本公开涉及在云环境中自动且周期性地更新多层应用栈。更具体地,本公开涉及访问中央服务器以确定是否已经发布新更新以及何时已经发布新更新,以最小的网络负担和服务中断自动将新更新应用于应用环境的系统和方法。

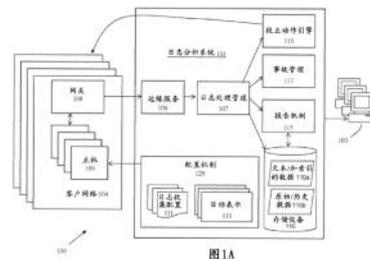


图1A

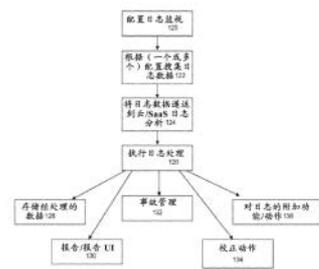


图1B

CN 110023901 B

1. 一种计算机实现的方法,包括:  
运行应用环境的实例,其中应用环境包括对应用环境所依赖的一个或多个组件的一个或多个依赖性;  
检测被配置为修改所述一个或多个组件中的至少一个组件的一个或多个更新;  
响应于检测到所述一个或多个更新,访问更新服务器以检索所述一个或多个更新;  
自动生成应用环境的新实例;  
在应用环境的所述新实例上安装所述一个或多个更新;  
识别与所述实例相关联的元数据;  
识别与所述新实例相关联的新元数据;  
将与所述实例相关联的元数据和与所述新实例相关联的新元数据进行比较;  
基于该比较,确定与所述实例相关联的元数据和与所述新实例相关联的新元数据之间的一个或多个差异;  
生成包括识别出的所述一个或多个差异中的每一个差异的更新数据;以及  
通过将识别出的所述一个或多个差异结合到所述实例中,使用更新数据来更新所述实例。
2. 如权利要求1所述的计算机实现的方法,其中更新所述实例还包括:  
发起锁定期,在此期间停止与在应用环境中运行的应用相关联的所有事务;  
在锁定期期间用更新数据来更新所述实例;以及  
释放锁定期,锁定期的释放使得与所述应用相关联的事务能够继续。
3. 如权利要求1所述的计算机实现的方法,其中所述元数据包括与所述实例相关联的第一错误号集合,其中所述新元数据包括与所述新实例相关联的第二错误号集合,其中当第一错误号集合与第二错误号集合不同时,第一集合与第二集合之间的差异与所述一个或多个更新中包括的一个或多个错误修复对应。
4. 如权利要求3所述的计算机实现的方法,其中更新数据包括与在所述一个或多个更新中包括的所述一个或多个错误修复相关联的对象。
5. 如权利要求1所述的计算机实现的方法,其中检测存储在中央服务器上的所述一个或多个更新包括周期性地轮询中央服务器以便以规律或不规律的间隔进行更新。
6. 如权利要求1所述的计算机实现的方法,其中所述元数据与没有所述一个或多个更新的源代码相关联,并且其中所述新元数据与具有所述一个或多个更新的源代码相关联。
7. 如权利要求1所述的计算机实现的方法,还包括:  
重新启动更新后的实例。
8. 一种计算机系统,包括:  
一个或多个数据处理器;以及  
包含指令的非瞬态计算机可读存储介质,当指令在所述一个或多个数据处理器上执行时,使所述一个或多个数据处理器执行包括以下操作的操作:  
运行应用环境的实例,其中应用环境包括对应用环境所依赖的一个或多个组件的一个或多个依赖性;  
检测被配置为修改所述一个或多个组件中的至少一个组件的一个或多个更新;  
响应于检测到所述一个或多个更新,访问更新服务器以检索所述一个或多个更新;

自动生成应用环境的新实例；  
在应用环境的所述新实例上安装所述一个或多个更新；  
识别与所述实例相关联的元数据；  
识别与所述新实例相关联的新元数据；  
将与所述实例相关联的元数据和与所述新实例相关联的新元数据进行比较；  
基于该比较，确定与所述实例相关联的元数据和与所述新实例相关联的新元数据之间的一个或多个差异；  
生成包括识别出的所述一个或多个差异中的每一个差异的更新数据；以及  
通过将识别出的所述一个或多个差异结合到所述实例中，使用更新数据来更新所述实例。

9. 如权利要求8所述的计算机系统，其中更新所述实例还包括：  
发起锁定期，在此期间停止与在应用环境中运行的应用相关联的所有事务；  
在锁定期期间用更新数据来更新所述实例；以及  
释放锁定期，锁定期的释放使得与所述应用相关联的事务能够继续。

10. 如权利要求8所述的计算机系统，其中所述元数据包括与所述实例相关联的第一错误号集合，其中所述新元数据包括与所述新实例相关联的第二错误号集合，其中当第一错误号集合与第二错误号集合不同时，第一集合与第二集合之间的差异与在所述一个或多个更新中包括的一个或多个错误修复对应。

11. 如权利要求10所述的计算机系统，其中更新数据包括与在所述一个或多个更新中包括的所述一个或多个错误修复相关联的对象。

12. 如权利要求8所述的计算机系统，其中检测存储在中央服务器上的所述一个或多个更新包括周期性地轮询中央服务器以便以规律或不规律的间隔进行更新。

13. 如权利要求8所述的计算机系统，其中所述元数据与没有所述一个或多个更新的源代码相关联，并且其中所述新元数据与具有所述一个或多个更新的源代码相关联。

14. 如权利要求8所述的计算机系统，所述操作还包括：  
重新启动更新后的实例。

15. 一种非瞬态机器可读存储介质，包括被配置为使数据处理装置执行操作的指令，所述操作包括：

运行应用环境的实例，其中应用环境包括对应用环境所依赖的一个或多个组件的一个或多个依赖性；

检测被配置为修改所述一个或多个组件中的至少一个组件的一个或多个更新；  
响应于检测到所述一个或多个更新，访问更新服务器以检索所述一个或多个更新；  
自动生成应用环境的新实例；  
在应用环境的所述新实例上安装所述一个或多个更新；  
识别与所述实例相关联的元数据；  
识别与所述新实例相关联的新元数据；  
将与所述实例相关联的元数据和与所述新实例相关联的新元数据进行比较；  
基于该比较，确定与所述实例相关联的元数据和与所述新实例相关联的新元数据之间的一个或多个差异；

生成包括识别出的所述一个或多个差异中的每一个差异的更新数据;以及  
通过将识别出的所述一个或多个差异结合到所述实例中,使用更新数据来更新所述实例。

16. 如权利要求15所述的非瞬态机器可读存储介质,其中更新所述实例还包括:  
发起锁定期,在此期间停止与在应用环境中运行的应用相关联的所有事务;  
在锁定期期间用更新数据来更新所述实例;以及  
释放锁定期,锁定期的释放使得与应用相关联的事务能够继续。

17. 如权利要求15所述的非瞬态机器可读存储介质,其中所述元数据包括与所述实例相关联的第一错误号集合,其中所述新元数据包括与所述新实例相关联的第二错误号集合,其中当第一错误号集合与第二错误号集合不同时,第一集合与第二集合之间的差异与  
所述一个或多个更新中包括的一个或多个错误修复对应。

18. 如权利要求17所述的非瞬态机器可读存储介质,其中更新数据包括与在所述一个或多个更新中包括的所述一个或多个错误修复相关联的对象。

19. 如权利要求15所述的非瞬态机器可读存储介质,其中检测存储在中央服务器上的  
所述一个或多个更新包括周期性地轮询中央服务器以便以规律或不规律的间隔进行更新。

20. 如权利要求15所述的非瞬态机器可读存储介质,其中所述元数据与没有所述一个或多个更新的源代码相关联,并且其中所述新元数据与具有所述一个或多个更新的源代码  
相关联。

21. 如权利要求15所述的非瞬态机器可读存储介质,所述操作还包括:  
重新启动更新后的实例。

22. 一种用于更新应用环境的实例的系统,包括用于执行如权利要求1-7中任一所述的  
计算机实现的方法的装置。

## 用于更新基于云的多层应用栈的系统和方法

[0001] 相关申请

[0002] 本申请援引35U.S.C 119(e)要求于2017年9月21日提交的标题为“SYSTEMS AND METHODS FOR UPDATING MULTI-TIER CLOUD-BASED APPLICATION STACKS”的美国临时申请序列No.62/561,599的权益和优先权,其内容通过引用整体并入本文,用于所有目的。

### 技术领域

[0003] 本公开涉及在云网络环境中自动和周期性地更新多层应用栈。更具体而言,本公开涉及周期性地(例如,以规律或不规律的间隔)访问中央服务器以确定是否已经释放对应用的新更新的系统和方法。当已发布对应用的新更新时,系统和方法自动将新更新应用于与应用相关联的应用环境,同时将网络负担和服务中断降至最低。

### 背景技术

[0004] 一般而言,应用栈可以使用多个层来实现,包括web层(例如,web服务器)、应用层(例如,应用服务器)和数据库层(例如,数据库服务器)。软件组件可以在这些层中的每一层上独立运行。各层的软件组件可以被配置为在同一服务器内或跨多个服务器进行通信。另外,这些服务器可以是虚拟机或物理机,在同一数据中心内或跨不同区域中的数据中心运行。随着时间的推移,可能需要更新应用栈,例如,修复错误或向使用应用栈提供的服务添加功能。更新应用栈通常涉及雇用供应商更新和测试每一层的不同软件组件,以测试安装软件组件时出现的问题。但是,这种更新处理对网络资源来说是繁重的,因为复杂的应用栈可以非常大(例如,1TB),并且服务器通常需要长时间脱机以便能够测试服务器。

[0005] 即使在没有传统应用栈的应用环境中,应用也可以依赖于云网络环境中的各种资源或组件。通过重写应用的映像来更新应用可能会破坏这些依赖性,从而导致进一步的错误和/或失败。在重写应用映像之后,应用环境的管理人员需要通过重新连接或重新建立所有这些依赖性来跟进(follow up)更新处理。这样的跟进处理一般是乏味的,但在具有大量这些依赖性的较大应用环境中甚至更乏味。

### 发明内容

[0006] 本公开的某些方面和特征涉及用于在云环境中自动更新多层应用栈的系统和方法,其中处理资源的负担最小并且服务器停机时间有限。特别地,某些实施例涉及在云环境中的应用栈上运行的应用(例如,应用的3层模型,包括托管web代码的web服务器、托管应用代码的应用服务器和提供数据库服务的数据库服务器)。应用可以被配置为周期性地与中央服务器(例如,更新服务器、中央服务器等)通信,以检查对应用栈的新更新。可以在中央服务器处周期性地(例如,以规律或不规律的间隔)存储对应用栈的更新。更新的非限制性示例可以包括更新图像和补丁集,其被配置为修复应用错误和/或添加或修改应用的特征。更新图像(例如,每季度发布一次的主要版本)可以是包括所有错误修复和特定版本的应用的新特征的数据包。补丁集(例如,主要版本之间发布的小错误修复)可以是包含自上次更

新映像发布以来报告的各个错误修复的数据包。在一些实施例中，更新可以是对应用的应用代码中包括的对象的二进制改变(例如，错误修复)的形式。如果可以从中央服务器下载新更新，那么应用可以将新更新下载到与应用栈相关联的网络位置(例如，将新更新下载到与应用栈相关联的储存库)。

[0007] 在一些实施例中，应用与生产环境中的当前运行的实例相关联。当前运行的实例可以与不包括存储在中央服务器中的新更新的应用的版本(例如，尚未更新的应用的旧版本和/或新更新尚未在其上安装的应用的最新版本)对应。可以将应用被配置为生成(spawn)虚拟机并在虚拟机上运行应用栈的新实例。另外，应用可以将下载的新更新安装在应用的新实例上。例如，安装了新更新后的新实例将包含新更新中包含的任何错误修复或新特征。此外，应用可以将当前运行的实例与新实例进行比较，以确定是否存在任何差异。例如，当前运行的实例与新实例之间的差异可以表示包括在新实例中的新更新(例如，新添加到应用代码中的对象、添加到应用的新特征、应用的二进制代码中的固定错误等)。比较期间检测到的差异可以被序列化为更新数据包，并导出到当前运行的实例以进行安装。更新数据包可以安装在当前运行的实例中，因此，将使用新的错误修复和/或新特征更新当前实例。例如，在当前运行的实例中安装更新数据包可以包括识别与检测到的差异相关联的对象(例如，由新更新修改、删除或添加的文件引用对象)并将修改、删除或添加的对象结合到存储在当前运行的实例的应用服务器处的应用代码处。新更新可以更新在多个层中的任何层上运行的软件组件。为了说明，并且作为非限制性示例，新更新可以修复web服务器的软件组件中的错误，而无需更新数据库服务器的软件组件。在一些实施例中，应用环境可以包括对应用环境所依赖的一个或多个组件(例如，软件组件)的依赖性(例如，由于这些依赖性，应用环境的可执行映像是不同的)。

[0008] 在一些实施例中，可以通过将与当前运行的实例相关联的元数据和与新实例相关联的元数据(其包括新的更新)进行比较来实现当前运行的实例与新实例的比较。在一些实施例中，与当前运行的实例和新实例中的任一个或两者相关联的元数据可以包括软件特征标志、包括在代码中的文件引用对象的标识符、安装或执行设置或者其它元数据参数。在这些实施例中，比较当前运行的实例与新实例之间的元数据可以包括将旧实例中启用/禁用的特征标记与新实例中启用/禁用的特征标记进行比较，比较旧实例中的安装或执行设置或参数值与新实例中的安装或执行设置或参数值，将当前运行的实例的文件引用对象的标识符与新实例的文件引用对象的标识符进行比较，以识别哪些文件引用对象已被修改、删除或添加到代码。在一些实施例中，与当前运行的实例相关联的元数据可以包括一个或多个错误号。错误号可以唯一地识别错误报告，该报告描述针对当前实例修复或修改的错误。类似地，与新实例(其包括新更新)相关联的元数据也可以包括一个或多个错误号，其表示为新实例修复的错误。当与当前运行的实例相关联的错误号的集合和与新实例关联的错误号的集合不同时，应用可以确定对新实例执行了更新。应用可以查询和检索针对为新实例但不为当前正在运行的实例修复的错误的错误报告(例如，错误号之间的增量)。每个错误号和/或错误报告与促进、实现或执行错误修复的一个或多个对象(例如，文件引用对象)相关联。可以识别并导出当前运行的实例的错误号的集合与新实例的错误号的集合之间不同的错误号，并且可以在当前运行的实例上执行或实施识别出的错误号，这导致更新当前运行的实例以包括新实例中的更新。

[0009] 在一些实施例中,新更新可以包括具有对涉及数据库服务器的一个或多个对象的改变的源代码。可以将对一个或多个对象的改变添加到与新更新相关联的元数据,作为表示改变的对象的文件引用对象,并存储在源目录中。例如,文件引用对象可以具有存储在元数据表中的标识符,以供将来参考和提取。作为另一个示例,可以将对代码、文件或脚本的二进制改变(例如,诸如Java Archive (JAR) 文件之类的Java文件中的改变)添加到代码、文件或脚本的元数据,作为文件引用对象。文件引用对象表示指向部署文件的网络位置的路径。在这些实施例中,当将当前运行的实例与新实例进行比较时,可以存储、序列化当前运行的实例的文件引用对象与新实例之间的差异,并将其应用于当前运行的实例。

[0010] 有利地,当应用在生产环境中操作时,本发明的某些实施例可以自动更新多层应用栈。另外,在应用更新期间负担和/或中断网络资源的技术问题通过上文和本文描述的技术新颖性解决。例如,将应用的新虚拟实例的元数据与原始或当前运行的应用实例(即,在生产环境中运行的应用)的元数据进行比较并安装表示基于该比较检测到的差异的元数据改善了云网络环境中处理资源的使用并减少了应用栈的服务中断。此外,无论在当前运行的实例的版本和新实例的版本之间发布的更新的尺寸或应用的迭代或版本的数量是多少,都可以更新多层应用栈。

## 附图说明

[0011] 说明书参考以下附图,其中在不同附图中使用相同的标号旨在示出相同或类似的组件。

[0012] 图1A图示了根据本公开一些实施例的用于配置、收集和分析日志数据的示例系统。

[0013] 图1B图示了使用系统来配置、收集和分析日志数据的方案的流程图。

[0014] 图2图示了示例报告用户界面。

[0015] 图3A-3C是图示主机环境处的日志分析系统的内部结构的流程图。

[0016] 图4是图示示例网络环境的框图。

[0017] 图5是图示用于自动更新多层应用栈的处理流程的框图。

[0018] 图6-8是根据某些实施例的示例界面。

[0019] 图9是图示用于创建模板的处理的流程图。

[0020] 图10是图示用于实现实施例之一的分布式系统的简化图。

[0021] 图11是图示系统环境的一个或多个组件的简化框图。

[0022] 图12图示了示例性计算机系统,其中可以实现本发明的各种实施例。

## 具体实施方式

[0023] 在以下描述中,出于解释的目的,阐述了具体细节以便提供对本发明实施例的透彻理解。但是,显而易见的是,可以在没有这些具体细节的情况下实践各种实施例。附图和描述不是限制性的。

[0024] 许多类型的计算系统和应用生成大量与那个计算系统或应用的操作有关或由此产生的数据。然后,这些大量数据经常被存储到收集的位置,诸如日志文件/记录,如果需要分析系统或应用的行为或操作,那么可以在稍后的时间段查看这些数据。

[0025] 虽然以下描述可以通过关于“日志”数据的说明来描述实施例,但是还可以预期其它类型数据的处理。因此,实施例在其应用方面不仅限于日志数据。此外,以下描述也可以互换地指被处理为“记录”或“消息”的数据,而无意将本发明的范围限制到数据的任何特定格式。

[0026] 图1A图示了根据本发明一些实施例的用于配置、收集和分析日志数据的示例系统100。系统100包括日志分析系统101,在一些实施例中,日志分析系统101被实施为基于云和/或基于SaaS(软件即服务)的体系架构。这意味着日志分析系统101能够将日志分析功能作为被托管平台上的服务提供,使得需要该服务的每个客户不需要在客户自己的网络上单独安装和配置服务组件。日志分析系统101能够向多个单独的客户id提供日志分析服务,并且可以被扩展以服务于任何数量的客户。

[0027] 每个客户网络104可以包括任何数量的主机109。主机109是客户网络104内的计算平台,其生成作为一个或多个日志文件的日志数据。在主机109内产生的原始日志数据可以源自任何日志产生源。例如,原始日志数据可以源自数据库管理系统(DBMS)、数据库应用(DB App)、中间件、操作系统、硬件组件或任何其它创建日志的应用、组件或系统。在每个客户网络中提供一个或多个网关108以与日志分析系统101通信。

[0028] 系统100可以包括在一个或多个用户站103处的一个或多个用户,其使用系统100来操作日志分析系统101并与之交互。用户站103包括可以用于操作系统100中的日志分析系统101或与之交互的任何类型的计算站。这种用户站的示例包括例如工作站、个人计算机、移动设备或远程计算终端。用户站包括显示设备,诸如显示监视器,用于向用户站处的用户显示用户界面。用户站还包括一个或多个输入设备,以供用户提供对系统100的活动的操作控制,诸如鼠标或键盘,以操纵图形用户界面中的指向对象以生成用户输入。在一些实施例中,用户站103可以(但不是必须)位于客户网络104内。

[0029] 日志分析系统101包括用户站101处的用户可访问的功能,其中日志分析系统101被实现为引擎、机制和/或模块的集合(无论是硬件、软件还是硬件和软件的混合)以执行日志数据的配置、收集和分析。用户界面(UI)机制生成UI以显示分类和分析结果,并允许用户与日志分析系统交互。

[0030] 图1B示出了使用系统100来配置、收集和分析日志数据的方案的流程图。图1B的这个讨论将参考图1A中针对系统100所示的组件。

[0031] 在120处,在系统内配置日志监视。例如,这可以通过用户/客户配置用户/客户期望的日志监视/数据搜集的类型而发生。在系统101内,包括UI控件的配置机制129可由用户操作,以选择和配置日志收集配置111和用于日志收集配置的目标表示113。

[0032] 日志收集配置111包括识别要收集哪些数据(例如,哪些日志文件)、要收集数据的位置(例如,目录位置)、如何访问数据(例如,日志的格式和/或要获取的日志中的具体字段)和/或何时收集数据(例如,定期地)的信息集(例如,日志规则、日志源信息和日志类型信息)。日志收集配置111可以包括由服务提供者包括的开箱即用规则。日志收集配置111还可以包括客户定义/客户定制的规则。

[0033] 目标表示113识别“目标”,它是包含和/或产生日志的客户环境内的各个组件。这些目标与客户环境中的具体组件/主机相关联。示例目标可以是具体数据库应用,它与一个或多个日志和/或一个或多个主机相关联。

[0034] 122处的下一个动作是根据用户配置捕获日志数据。日志数据可以源自任何日志产生源位置,诸如数据库管理系统、数据库应用、中间件、硬件日志、操作系统日志、应用日志、应用服务器日志、数据库服务器日志以及任何其它类型的监视系统或应用的行为的日志。

[0035] 在一些实例中,日志规则111与目标表示之间的关联被发送到客户网络104以进行处理。日志分析系统的代理存在于每个主机109上,以从主机109上的适当日志收集数据。

[0036] 在一些实施例中,可以对捕获的数据执行数据遮蔽。遮蔽是在收集时执行的,它在客户数据离开客户网络之前保护客户数据。例如,所收集的日志数据中的各种类型的信息(诸如用户名和其它个人信息)可以足够敏感,以在被发送到服务器之前被遮蔽。为这样的数据识别模式,其可以在为服务器收集之前被移除和/或改变为代理服务器数据。这允许数据仍然用于分析目的,同时隐藏敏感数据。一些实施例永久地移除敏感数据(例如,将所有此类数据改变为“\*\*\*”符号),或者改变为被映射的数据,以便可以恢复原始数据。

[0037] 在124处,将收集的日志数据从客户网络104递送到日志分析系统101。客户网络104中的多个主机109将收集的数据提供给较少数量的一个或多个网关108,然后网关108将日志数据发送到日志分析系统101处的边缘服务106。边缘服务106接收通过一个或多个客户网络收集的数据,执行任何摄取处理(例如,应用语法规则以将每个消息变换为缺少集群间消息可变性的分量的标准化的消息或概略消息,并将每个经变换的消息指派给使用经变换的息的散列识别出的初始集群),并且可以将数据放入入站数据存储装置中以供日志处理管道107进一步处理。

[0038] 在126处,日志处理管道107对收集的日志数据执行一系列数据处理和分析操作。在各种情况下,处理和分析操作可以包括在存储数据之前执行的动作和/或通过对从数据存储装置检索到的数据执行动作。例如,可以在摄入时间(例如,在从源接收到(一个或多个)日志消息时)将一个或多个日志消息指派给初始集群,并且随后可以响应于查询而检索(一个或多个)日志消息,以修改或补充初始聚类并基于聚类生成统计和/或呈现。

[0039] 在128处,然后将经处理的数据存储到数据存储设备110中。计算机可读存储设备110包括硬件和软件的任意组合,其允许随时访问位于计算机可读存储设备110的数据。例如,计算机可读存储设备110可以被实现为由操作系统可操作地管理的计算机存储器。计算机可读存储设备110中的数据还可以被实现为数据库对象、云对象和/或文件系统中的文件。在一些实施例中,经处理的数据存储在文本/加索引的数据存储装置110a(例如,作为SOLR集群)和原始/历史数据存储装置110b(例如,作为HDFS集群)内。

[0040] SOLR集群与Apache™开源本地搜索平台对应。SOLR集群可以使用搜索库来执行全文加索引以及存储在HDFS集群中的数据的搜索。SOLR集群可以提供与各种语言兼容的API,以将搜索功能与其它程序和应用通过接口接合。加索引可以近乎实时地执行。集群可以在服务器集合上运行,以促进容错和可用性。加索引和搜索任务可以分布在服务器集合上。

[0041] HDFS集群与Hadoop分布式文件系统集群对应。HDFS集群可以包括许多(例如,数千个)服务器以托管存储装置(例如,直接附连的存储装置)并执行任务,诸如由用户应用定义的任务。HDFS集群可以包括主/从体系架构,其具有用于管理集群的命名空间的单个主服务器。可以将文件划分为块以存储在HDFS集群的多个DataNode上。主服务器可以执行文件操作(例如,打开、关闭等)并确定哪些块将存储在哪些数据节点上。主服务器可以与数据节点

交流响应于接收到对应的文件操作而读取或写入数据的请求。

[0042] 在130处,可以使用报告机制/UI 115对经处理的数据执行报告。如图2中所示,报告UI 200可以包括日志搜索工具202、一个或多个仪表盘204,和/或用于分析/查看经处理的日志数据的任何合适的应用206。下面更详细地描述这种报告组件的示例。

[0043] 在132处,可以对经处理的数据执行事故(incident)管理。可以在日志分析系统内配置一个或多个警报条件,使得在检测到警报条件时,事故管理机制117向指定的用户集合提供事故/警报的通知。

[0044] 在134处,校正动作引擎119可以执行在客户网络104内采取的任何必要动作。例如,可以接收数据库系统停机的日志条目。当识别出这种日志条目时,可能的自动校正动作是尝试重新启动数据库系统。客户可以创建校正动作脚本来解决这种情况。可以执行触发器以运行执行校正动作的脚本(例如,触发器使得指令被发送到客户网络上的代理以运行脚本)。在替代实施例中,将用于该情况的适当脚本从服务器下推到客户网络以便被执行。此外,在136处,可以至少基于经处理的数据适当地采取任何其它附加功能和/或动作。

[0045] 图3A提供了主机环境340处的日志分析系统的内部结构和客户环境342内与日志分析系统交互的组件的更详细的图示。这个体系架构300被配置为提供能够处置大量日志数据摄入的日志监视流。

[0046] 在单个客户主机/服务器344内的客户环境342中,LA(日志分析)代理333取得日志监视配置数据332(例如,嗅探器配置或目标侧配置素材),并调用日志文件336嗅探器(本文也称为“日志收集器”),以从一个或多个日志文件338搜集日志数据。

[0047] 守护进程管理器334可以被用来与日志文件嗅探器336通过接口接合。日志文件嗅探器336从主机机器344上的一个或多个日志文件338中读取。守护进程管理器334取得日志内容并将其打包,以便可以将其传递回LA代理333。要注意的是,系统可以包括任何数量的不同种类的嗅探器,并且日志嗅探器336仅仅是可以在系统中使用的单一类型的嗅探器的示例。因此,可以在本发明的各种实施例中采用其它类型的嗅探器,例如,监视注册表、数据库、窗口事件日志等的嗅探器。此外,在一些实施例中,日志嗅探器被配置为处置集体/压缩文件,例如,Zip文件。

[0048] LA代理333将搜集的日志数据发送到网关代理330。网关代理330打包从多个客户主机/服务器收集的日志数据,基本上充当聚合器以聚合来自多个主机的日志内容。然后,打包的内容从网关代理330发送到边缘服务306。边缘服务306来自任意数量的不同客户环境342的多个网关代理330接收大量数据。

[0049] 给定可以在边缘服务306处接收的潜在大量数据,可以立即处理数据以将每个日志消息指派给初始集群并存储到进站数据存储设备304(“平台进站聚类存储装置”)中。在一些实例中,可以在摄入时间执行初始或初步处理,这可以包括与数据存储对应的时间(例如,之前、之后很短时间或之后立即,或与之并发)。初始或初步处理可以包括(例如)检测数据的哪些部分是不可变分量,并基于在消息中检测到的不可变分量确定每个日志消息的初始集群。例如,可以将散列技术应用每个不可变分量的值以生成初始集群的标识符。然后将日志消息与初始集群的标识符相关联地存储,或者可以存储其它集群数据以指示日志消息与初始集群相关联。可以在后续处理期间进一步细化、增强和/或使用集群指派,诸如在后续资源可用性的时间期间发生的处理期间和/或响应于接收对与相关联的日志消息

对应或可能对应的数据的查询。

[0050] 因此,在一些实例中,管理和维护队列,其中队列元素与集群指派要对其进行细化、增强和/或使用的一个或多个日志消息对应。在队列元素的初始存储之后和/或响应于接收与一个或多个相关联的日志消息对应或可能对应的数据的查询,可以将元素添加到队列(例如)。该队列可以用于日志处理管道308。

[0051] 提供数据结构以管理进站数据存储装置内要处理的项。在一些实施例中,消息传递平台302(例如,使用Kafka产品实现)可以用于跟踪队列内的待处理项。在日志处理管道308内,队列消费者310识别要处理的队列中的下一个项,然后从平台进站存储装置中检索该项。队列消费者310包括能够处理系统内离开队列的工作的任何实体,诸如进程、线程、节点或任务。

[0052] 检索到的日志数据经历“解析”阶段312,其中日志条目被解析并分解成具体的字段或分量。为日志配置的“日志类型”指定如何将日志条目分解为期望的字段。

[0053] 在“聚类”阶段313,进一步分析日志数据以将各个日志消息指派给集群。具体而言,可以评估在摄取处理期间(例如,在304处)向其指派日志消息的多个初始集群,以确定是否将其中一些初始集群合并在一起。评估可以包括识别每个集群的一个或多个代表性样品并进行成对定量比较评估。经由成对比较评估被评估的集群对可以包括具有相同或相似数量的分量(或词)的日志消息的集群。在一些实例中,每对集群包括与多个分量相关联的集群,所述多个分量相同或彼此相差小于阈值数量(例如,预定义、默认数量或由用户识别),使用评估被评价。比较评估可以迭代地和/或以结构化方式执行(例如,使得在评价具有不同数量的分量的对之前评价具有相同数量的分量的对)。

[0054] 成对定量比较评估可以包括,例如,使用代表消息生成相似度度量,并确定度量是否超过阈值度量(例如,预定义的、由用户识别出的默认数量)。相似度度量可以基于(例如)代表性消息是否包括相同(或相似)数量的分量、可变(或不可变)分量的数量、一个或多个不可变分量中的每一个的内容、一个或多个可变分量的特点(例如,格式、字符类型或长度)等等。相似度度量可以基于生成集群间消息之间的相关系数,或者通过使用更大的消息集执行聚类技术,达到集群的代表性消息被指派给相同集群或共享分量的程度(例如,如果技术包括使用分量分析,诸如主分量分析或独立分量分析)。

[0055] 在“标准化”阶段314中,识别出的字段被标准化。例如,“time(时间)”字段可以在不同日志中以任何数量的不同方式表示。这个时间字段可以被标准化为单个可识别的格式(例如,UTC格式)。作为另一个示例,词“error(错误)”可以在不同系统上以不同方式表示(例如,全部大写“ERROR”、全部小写“error”、第一个字母大写“Error”或缩写“err”)。这种情况可以要求将不同的词形式/类型标准化为单一格式(例如,所有小写的非缩写术语“error”)。

[0056] “变换”阶段316可以用于从日志数据合成新内容。作为示例,可以将“标签”添加到日志数据中,以提供关于日志条目的其它信息。作为另一个示例,标签可以识别向其指派了日志消息的集群。

[0057] “条件评价”阶段318用于评价日志数据上的指定条件。可以执行这个阶段以识别日志数据内的模式,并创建/识别日志中的警报条件。可以在这个阶段执行任何类型的通知,包括例如发送给管理员/客户的电子邮件/文本消息/呼叫或者到另一个系统或机制的

警告。作为一个示例,条件可以定义与集群指派的改变对应的事件,诸如检测指派给给定集群的日志消息的数量(例如,数量或百分比)已经超过阈值(例如,是固定的并且由用户、客户端或规则预定义),诸如低于下阈值或高于上阈值。作为另一个示例,条件可以定义与指派给给定阈值的日志消息的数量正在改变的程度对应的事件,诸如通过识别时间序列的斜率的阈值或者在两个时间区间(bin)之间指派给集群的日志消息的计数或百分比的差的阈值。作为又一个示例,条件可以定义与多个集群指派对应的事件,诸如指示多个集群中的每个集群的时间序列具有相似形状的事件(例如,通过确定曲线拟合系数是否在阈值量内足够相似、通过确定时间序列中的一个或多个峰值的时间是否在既定的阈值时间内、确定集群的时间序列之间的相关系数是否超过阈值,和/或确定每个个体集群的时间序列的可变性与时间序列之和的可变性之间的差异是否超过阈值)。

[0058] 然后,日志写入器320将经处理的日志数据写入一个或多个数据存储装置324。在一些实施例中,经处理的数据被存储在文本/加索引的数据存储装置(例如,作为SOLR集群)和原始和/或历史数据存储装置(例如,作为HDFS集群)内。日志写入器还可以将日志数据发送到另一个处理阶段322和/或下游处理引擎。

[0059] 如图3B中所示,一些实施例提供了侧加载机制350,用于收集日志数据而无需通过客户端侧的代理333。在这种方案中,用户登录到服务器以选择本地系统上的一个或多个文件。系统将在服务器上加载那个文件,并将嗅探那个文件(例如,通过让用户提供日志类型、尝试可能的日志类型、滚动不同的日志类型,或通过对日志类型进行有根据的“猜测”)。然后,如前所述,将嗅探结果传递给边缘服务并进行处理。在图3C的实施例中,仅存在侧加载机制350以搜集日志文件-其中在客户端服务器344上未安装和/或不需代理/嗅探器实体。

[0060] 图4示出了用于将内部部署应用导出到云系统的示例网络环境400。网络环境400可以包括本地系统410和云系统450。本地系统410可以包括一个或多个内部子系统,诸如数据库420、应用服务器430和web服务器440。本地系统410的每个内部子系统可以不包括在云系统中。在一些实现中,数据库420、应用服务器430和web服务器440的物理组件可以在与公司相关联的设施处内部部署。数据库420、应用服务器430和web服务器440中的每一个可以至少部分地执行一个或多个应用。

[0061] 将认识到的是,虽然术语“内部部署”用于指代位于公司设施处的系统,但是本公开不限于此。例如,本地系统410及其内部子系统可以或可以不在公司的设施处物理地内部部署,而是内部子系统可以分布在不与公司相关的区域(例如,租用的服务器)上。此外,本地系统410可以用于运行由公司管理的一个或多个应用。

[0062] 在一些实现中,云系统450可以包括储存库460、云管理器470、云管理器应用栈475和中央服务器480。对多层应用栈的更新可以周期性地存储在中央服务器480中。例如,更新490可以包括更新一个或多个应用的代码。更新490的非限制性示例可以包括错误修复、向应用添加的特征、向应用添加的功能、修改后的特征、修改后的功能、删除的特征和/或删除的功能。可以由一个或多个程序员定义更新490(例如,代码被编写),或者在一些情况下,可以基于一个或多个规则和/或机器学习技术自动创建更新490。由于周期性地定义新更新,因此更新可以作为更新490存储在中央服务器480中。中央服务器480可以存储用于各种应用的多个更新。储存库460可以包括一个或多个存储系统,用于存储从本地系统导出到云系

统450的传入应用的部署包。例如,部署包可以是可跨系统导出的应用的序列化版本。云管理器470可以是配置为管理云管理器应用栈475的一个或多个方面的一个或多个应用、系统或引擎。例如,云管理器管理员可以使用云管理器应用栈475来操作或访问云管理器470以运行云管理器应用。云管理器470可以是在云系统450中提供环境部署的应用。例如,云管理器470可以以自动方式在云系统450上部署各种应用。有关云管理器应用的其它详细信息,请参阅2017年5月16日提交的标题为“Distributed Versioning of Applications Using Cloud-Based Systems”的美国序列No.62/507,086,其公开内容通过引用整体并入本文,用于所有目的。

[0063] 在一些示例中,本地系统410的网络管理员可能需要将他们的企业应用导出到云(例如,以增加企业应用的可伸缩性)。本公开的实施例使得本地系统的网络管理员能够将他们的企业应用从内部子系统导出到云系统450。在一些示例中,在应用的生命周期期间,应用管理员可以将他们的内部部署的应用迁移到云系统470。在迁移过程中,应用管理员可以使用部署包导出内部部署环境,并将部署包导入云系统450。云管理器470可以读取企业应用的部署包并在云系统450上部署应用。

[0064] 将认识到的是,新特征集或新映像可以在中央服务器上发布、可用或存储。新特征集或新映像可以包含最新版本应用的所有新特征。可以将新特征集或新映像与应用的过时版本进行比较,并且可以通过自动的一步处理将新特征集或新映像移动到应用的过时版本。例如,使用来自用户的单个输入,新特征集或新映像可以应用于应用的过时版本。

[0065] 还将认识到的是,云系统450可以被实现为元数据驱动的体系架构。例如,软件组件、Java代码、python代码、任何其它合适的源代码可以被单独地表示为运行时环境内的数据库对象。继续这个示例,包括在应用下面的源代码中的软件组件可以被称为文件引用对象。每个文件引用对象可以与存储在元数据表中的元数据相关联。例如,文件引用对象的元数据可以包括文件引用对象的内部版本标识符。内部版本号可以是唯一地识别文件引用对象和/或文件引用对象的版本的唯一标识符。如果创建了文件引用对象的新版本(例如,如果文件引用对象被修改或更新),那么新版本的文件引用对象可以与新的内部版本号相关联。在一些实现中,文件引用对象可以被表示为二进制对象,并且那个二进制对象的引用(例如,唯一标识符)可以存储在数据库中,以将引用与二进制对象和/或二进制对象的网络位置相关联。

[0066] 在一些实现中,代码块可以与数据库对象对应、被称为数据库对象或可以是数据库对象。数据库对象可以与描述数据库对象的属性的内部唯一标识符和/或附加内部元数据对应。例如,作为与数据库对象相关联的元数据存储的数据库对象的属性可以包括如何执行代码块、如何在运行时表示代码块或数据库对象等。将认识到的是,本公开不限于文件引用对象和数据库对象;代替地,可以在本公开的实施例中,使用任何类型的对象。

[0067] 此外,当创建对象(例如,数据库对象)时,生成对象的版本,并且定义或描述对象的属性的元数据存储于开发表或表示表中。例如,通过针对两个对象查询存储在表示表中的元数据,应用和/或改变辅助工具可以被配置为识别两个对象之间的(一个或多个)差异和/或相似性。作为非限制性示例,应用和/或改变辅助工具可以被配置为将目标(例如,当前运行的应用实例)与源(例如,应用的新实例)进行比较以识别目标与源之间的差异。比较的结果可以是确定源包括文件引用对象ABC,版本12(与唯一标识符ABC12相关联),而目标

包括文件引用对象ABC,版本11(与唯一标识符ABC11相关联)。因而,应用可以访问存储与每个唯一标识符ABC12和ABC11相关联的元数据的表示表。然后,应用可以将与标识符ABC12相关联的文件引用对象序列化为更新数据包,并将更新数据包的应用(例如,安装)应用于当前运行的实例,以便当前运行的实例的文件引用对象ABC版本11可以更新至12版。在一些实现中,可以识别ABC12与ABC11之间不同的任何其它元数据并将其安装在当前运行的实例中,以便更新当前运行的实例以包括安装在新实例中的所有更新。此外,更新数据包可以是可在系统之间导出的文件。更新数据包使存储在表示表中的元数据的特征能够导出到当前运行的实例,而不是将整个新实例导出到当前运行的实例。有利地,更新后的对象的导出仅通过减小发送或移动到当前运行实例的导出文件的尺寸来改善网络资源使用。

[0068] 图5是图示用于自动更新多层应用栈的处理流程的框图。可以将对多层应用栈的定期更新存储在中央服务器480中。应用环境530可以包括储存库460,其被配置为从中央服务器480检索和存储更新(例如,更新490)。更新的非限制性示例可以包括更新映像和补丁集。例如,更新映像(例如,大约每季度发布一次的主要应用更新)可以是一个或多个数据包,其包括应用的特定版本的所有错误修复和新特征。补丁集(例如,在主要应用更新之间发布的次要应用更新,例如修复错误)可以是包含自上次更新映像被释放以来报告的各个错误修复的一个或多个数据包。每当释放更新映像时,更新映像的源文件可以存储在中央服务器480中。例如,在图5中,中央服务器480可以将更新490存储为更新映像或补丁集。中央服务器480可由云系统450中的所有应用栈访问,因此,中央服务器480可以存储针对任何数量的应用栈的任何数量的更新。参考图5,更新490可以是针对特定多层应用栈的更新。

[0069] 在一些实现中,与更新490相关联的应用栈可以被配置为周期性地(以规律或不规律的间隔)查询中央服务器480以进行更新。在一些实现中,中央服务器480可以向一个或多个应用栈发送通知。该通知可以用于向应用的用户或应用本身通知例如更新490当前可用于从中央服务器480下载。图6图示了示例界面,其示出了中央服务器480中的新更新可用的通知。当中央服务器480中有新更新可用时,应用栈的当前运行实例的版本可能已过时。当中央服务器480中有新的更新可用时,本发明的某些实施例使应用能够自动将其自身更新为最新版本。

[0070] 在一些实施例中,储存库460可以存储从中央服务器480检索到的更新。例如,储存库460可以向中央服务器480查询任何更新,并且如果可用,那么可以将更新490下载到储存库460。另外,应用环境530可以包括与多层应用栈相关联的当前运行的实例500。例如,当前运行的实例500可以在应用环境530中执行,并且可以在生产环境中的应用栈上运行。例如,真实用户设备可以与生产环境中当前运行的实例500通信,以使应用能够向用户提供服务。因为应用在生产环境中运行,所以为了更新的任何服务器停机都会产生负面影响,因为用户可能无法访问应用及其对应的服务。有利地,并且作为新颖性的技术要点,本发明的实施例使得能够以最小的服务器停机时间更新应用,而不管更新的尺寸(例如,1个错误修复或10000个错误修复)。

[0071] 如上所述,当前实例500可以是应用的过时版本。例如,当前实例500可以是不包括更新490的应用的版本。为了用更新490更新当前实例500,应用环境530可以生成新实例510。当前实例500和新实例510中的每一个可以在具有预定义配置设置(例如,CPU参数、RAM分配等)的相同或不同虚拟机上运行。另外,应用环境530可以被配置为在新实例510中安装

更新490。因而,新实例510可以是应用的最新版本(例如,安装有更新490并且可供用户使用的应用的版本)。

[0072] 应用环境530可以被配置为将当前实例500与新实例510进行比较,以识别当前实例500与新实例510之间是否存在任何增量520(例如,差异)。在一些实施例中,比较当前实例500与新实例510以识别增量520可以包括比较与每个实例相关联的错误号。例如,错误号可以是唯一标识符,其识别与在应用的源代码内修复的一个或多个错误(或源代码被修改以解决错误)对应的特定错误报告。例如,实例的错误号可以被存储为与实例相关联的元数据。在这个示例中,当前实例500的元数据可以存储指示错误号1和3先前已被固定或应用于当前实例500的数据。另外,新实例510的元数据可以存储指示错误号1、3和6被应用于新实例510的数据。在这个示例中,错误号1和3已经应用于新实例510,因为新实例510是当前实例500的新生成的实例,因此,错误号6将被检测为更新。例如,应用环境530可以将当前实例500的元数据与新实例510的元数据进行比较,以确定这两个实例之间的增量520是错误号6。

[0073] 因此,在更新490是在当前实例500中运行的应用的版本之后的应用的下一版本或迭代的情况下,更新490将与错误号6对应,指示更新490修复与错误号6相关联的一个或多个错误。当应用环境530识别出增量520与错误号6对应时,应用环境530可以访问与由错误号6的错误报告所解决的错误相关联的一个或多个对象。应用环境530可以导出该一个或多个对象,并将与错误号6相关联的错误修复结合到应用环境530的源代码中。由于可以在web服务器、应用服务器和/或数据库服务器上运行的特定软件组件中修复错误,因此可以将一个或多个对象应用于相关软件组件(例如,如果错误在web服务器的软件组件中被修复,那么该一个或多个对象将应用于与当前实例500相关联的web服务器的软件组件)。有利地,可以以最小的停机时间自动更新在生产环境中运行并且用户可访问的当前实例500。

[0074] 将认识到的是,当执行比较时,当前实例500仍然可以向用户提供服务(例如,没有服务器的停机时间)。当基于增量520更新应用栈时,应用环境530可以发起锁定,该锁定在既定的时间段期间阻止事务。在锁定期间,可以使用与增量520相关联的一个或多个对象来更新当前实例500。在更新当前实例500之后,可以重新启动当前实例500,并且可以释放锁定,使得更新490可以是可用的或结合到当前实例500中。

[0075] 将认识到的是,当发布新版本的应用时,将应用的新映像发布到中央服务器(例如,更新存储在中央服务器中)。新映像包括新版本的应用的所有特征。应用可以被配置为提供带有新版本的特征的应用的新实例。可以将改变辅助工具配置为将新实例的元数据与当前运行的应用实例(这将是应用的比新版本更为过时的版本)进行比较。然后,应用可以安装更新后的源代码或更新后的二进制代码的元数据,其可以包括数据库对象和二进制运行时。另外,更新后的源代码的元数据可以作为更新数据包以序列化形式移动到当前运行的实例,并安装或应用于当前运行的实例。

[0076] 图6-8图示了当用户访问在多层应用栈上运行的应用时在用户设备上呈现的示例界面。例如,界面600呈现“新修复可用于云管理器并准备好应用”的通知。在一些实现中,当用户访问应用的设置配置页面时,可以在用户设备上呈现界面600。作为非限制性示例,可以在设置配置页面上定义若干配置,诸如文件服务器配置、虚拟机配置和更新管理配置。虽然界面600可以在中央服务器(例如,中央服务器480)中存储和获得新的更新时呈现通知,

但是在一些实施例中,中央服务器中可用的新更新被自动下载并应用于应用,因此,不需要通知。在一些实施例中,可以在可用更新从中央服务器被下载并且被安装在当前运行的实例中之后(例如,在执行比较并且确定增量之后)通知用户。图7是图示在中央服务器(例如,中央服务器480)可用(例如,存储在其中)的更新的示例界面。如从界面700可以看出的,更新的示例可以是表示作为更新的一部分被修复的一个或多个错误的错误号。为了说明,界面700中示出的项目号10与错误号22093199对应。错误号22093199的描述是“当使用设计表单向导时没有显示OK按钮。”在这个图示中,错误号22093199被配置为修复应用中的错误-具体而言,“OK”按钮在设计表单向导中未显示的错误。在这个示例中,当前运行的实例尚未安装错误号22093199,因此,当前运行的实例将是应用的过时版本。但是,与错误号22093199对应的错误修复将安装在应用的新实例上。将当前运行的实例的元数据与新实例的元数据进行比较将导致错误号22093199被识别为尚未应用于当前运行的实例的错误号。在一些实施例中,存储在中央服务器中的一些或全部可用更新被自动下载到与应用相关联的储存库(例如,储存库460)。在一些实施例中,界面700使用户能够选择用户想要应用于当前运行的应用实例的更新。在一些实施例中,与当前运行的实例相关联的更新将自动应用于当前运行的实例,而无需接收来自用户的输入或指令。在这些实施例中,当前运行的实例始终以最新版本运行。

[0077] 图8是示例界面800,其图示了为自动更新当前运行的应用实例而执行的示例性步骤。可以在界面800上单独呈现每个步骤的状态。步骤1可以包括创建新的环境模板,其中可以部署应用的新实例。步骤2可以包括在新环境模板中部署新实例。步骤3包括安装“改变助手”,这是将源数据库(例如,已安装了更新的新实例)与目标数据库(例如,未安装更新的当前运行的实例)进行比较的工具。在步骤4处,可以将更新(例如,可以包括源代码内的错误修复的补丁集)应用于源数据库(例如,现在安装有更新的新实例)。在步骤5处,可以定义更新数据包(例如,“make me current”改变包)。更新数据包可以包括源数据库与目标数据库之间不同的对象的标识符。例如,源数据库与目标数据库之间不同的对象的标识符可以表示源数据库上安装的更新。在一些示例中,一旦识别出源数据库与目标数据库之间的差异,改变助理就可以生成更新数据包,其包括由更新改变的所有对象。如果源数据库与目标数据库之间的对象不同,那么那个对象可以由唯一标识符表示。一旦被识别出,那个对象就可以被下载到文件中,然后以序列化形式导出。在步骤6处,创建更新数据包(例如,包括检测到在源数据库和目标数据库之间已经改变的对象的标识符的数据打包)。在步骤7处,可以将更新数据包应用于目标数据库,以便可以更新目标数据库的文件并将其带到最新版本的应用。

[0078] 图9是图示用于自动更新多层应用的示例处理900的流程图。如上所述,可以在规律或不规律的基础上周期性地提供主要版本,并且可以在主要版本之间周期性地提供次要版本。例如,在主要版本之间,可以使用次要更新来修复主要版本中发现的错误。某些实施例使得应用栈能够以最小的停机时间和最小的网络负担自动更新,而不管最近的更新是主要版本还是次要更新。

[0079] 处理流程900开始于方框910,其中应用周期性地访问中央服务器以确定更新是否存储在中央服务器处。例如,应用可以被配置为随机地或规律地向中央服务器查询可用或存储在中央服务器的任何新更新。在方框920处,应用可以确定更新是否可用。如果没有可

用的更新,那么应用继续在另一个时间向中央服务器查询更新。但是,如果在中央服务器处存储了一个或多个更新,那么可以将这一个或多个更新下载到与应用相关联的储存库,如在方框930中。一旦将更新从中央服务器下载到储存库,就可以从中央服务器删除更新或将其标记为已下载。在方框940处,应用环境然后可以在云中提供其自身的新实例,并在新实例中安装该一个或多个更新。例如,提供新实例可以包括生成具有预定义或在运行中定义的配置设置(例如,CPU和RAM分配)的新虚拟机,并在安装了一个或多个更新的新虚拟机上运行应用栈。此时,应用环境可以与当前运行的应用实例(例如,当前与用户交互的应用的实例)和应用的新实例相关联,该应用安装了下载的更新。但是,在一些实现中,当前运行的实例位于生产环境中,而应用的新实例不在生产环境中。在这些实现中,最终用户不能与应用的新实例通信和/或应用的新实例不基于新实例和最终用户设备之间的交互来处理事务。代替地,最终用户继续与当前运行的实例进行通信。

[0080] 在方框950处,应用将当前运行的实例与新实例进行比较以确定两个实例之间是否存在任何差异(例如,增量)。例如,并且如上所述,比较这两个实例可以包括将当前运行的实例的元数据与新实例的元数据进行比较。在一些实现中,元数据可以包括已经或先前应用于实例的源代码的错误号。如果新实例的元数据中包含的错误号的集合与当前实例的元数据中包含的错误号的集合不同,那么应用可以确定更新与错误号中的差异(例如,增量)对应。例如,错误号1和3先前已应用于当前实例,并且如果错误号1、3和5先前已应用于新实例,那么错误号5的增量被确定为应用于当前实例以将当前实例更新为最新版本的应用。将认识到的是,元数据可以包括除错误号之外的数据。例如,表示修改后的源代码的文件引用对象和/或文件引用对象的标识符可以被存储为元数据。在这个示例中,将新实例的元数据与当前实例的元数据进行比较可以识别文件引用对象,这些对象由安装在新实例中的更新修改。在方框960处,应用可以创建更新数据包,其包括当前实例的元数据与新实例的元数据之间的增量。在方框970处,当前实例可以将数据包应用于其自身,这实际上是更新当前实例。有利地,对当前实例的更新最小程度地中断由应用提供的服务并改善网络效率。

[0081] 图10描绘了用于实现实施例之一的分布式系统1000的简化图。在所示实施例中,分布式系统1000包括一个或多个客户端计算设备1002、1004、1006和1008,其被配置为通过一个或多个网络1010执行和操作客户端应用,诸如web浏览器、专有客户端(例如,Oracle Forms)。服务器1012可以经由网络1010与远程客户端计算设备1002、1004、1006和1008通信耦合。

[0082] 在各种实施例中,服务器1012可以适于运行由系统的一个或多个组件提供的一个或多个服务或软件应用。在一些实施例中,这些服务可以作为基于web的或云服务或者在软件即服务(SaaS)模型下提供给客户端计算设备1002、1004、1006和/或1008的用户。操作客户端计算设备1002、1004、1006和/或1008的用户进而可以利用一个或多个客户端应用与服务器1012交互以利用由这些组件提供的服务。

[0083] 在图中描绘的配置中,系统1000的软件组件1018、1020和1022被示为在服务器1012上实现。在其它实施例中,系统1000的一个或多个组件和/或由这些组件提供的服务也可以由客户端计算设备1002、1004、1006和/或1008中的一个或多个来实现。然后,操作客户端计算设备的用户可以利用一个或多个客户端应用来使用由这些组件提供的服务。这些组

件可以用硬件、固件、软件或其组合来实现。应当认识到的是,各种不同的系统配置是可能的,其可以与分布式系统1000不同。因此,图中所示的实施例是用于实现实施例系统的分布式系统的一个示例,并不旨在进行限制。

[0084] 客户端计算设备1002、1004、1006和/或1008可以是便携式手持设备(例如,**iPhone®**、蜂窝电话、**iPad®**、计算平板电脑、个人数字助理(PDA))或可穿戴设备(例如,**GoogleGlass®**头戴式显示器),运行诸如Microsoft Windows**Mobile®**之类的软件,和/或各种移动操作系统(诸如iOS、Windows Phone、Android、BlackBerry 10、Palm OS等),并且启用互联网、电子邮件、短信服务(SMS)、**Blackberry®**或其它通信协议。客户端计算设备可以是通用个人计算机,包括例如运行各种版本的Microsoft **Windows®**、Apple**Macintosh®**和/或Linux操作系统的个人计算机和/或膝上型计算机。客户端计算设备可以是运行各种商用**UNIX®**或类UNIX操作系统中的任何一种的工作站计算机,包括但不限于各种GNU/Linux操作系统,诸如例如Google Chrome OS。可替代地或附加地,客户端计算设备1002、1004、1006和1008可以是任何其它能够通过(一个或多个)网络1010进行通信的电子设备,诸如瘦客户端计算机、启用互联网的游戏系统(例如,具有或不具有**Kinect®**手势输入设备的Microsoft Xbox游戏控制台)和/或个人消息传递设备。

[0085] 虽然示出了具有四个客户端计算设备的示例性分布式系统1000,但是可以支持任何数量的客户端计算设备。其它设备(诸如具有传感器的设备等)可以与服务器1012交互。

[0086] 分布式系统1000中的一个或多个网络1010可以是本领域技术人员熟悉的任何类型的网络,其可以使用各种商业可用协议中的任何协议来支持数据通信,包括但不限于TCP/IP(传输控制协议/互联网协议)、SNA(系统网络体系架构)、IPX(互联网分组交换)、AppleTalk等。仅作为示例,(一个或多个)网络1010可以是局域网(LAN),诸如基于以太网、令牌环等的局域网。(一个或多个)网络1010可以是广域网和互联网。它可以包括虚拟网络,包括但不限于虚拟专用网络(VPN)、内联网、外联网、公共交换电话网(PSTN)、红外网络、无线网络(例如,在电气和电子学会(IEEE)802.11协议套件、**Bluetooth®**和/或任何其它无线协议中的任何一个下操作的网络);和/或这些和/或其它网络的任意组合。

[0087] 服务器1012可以由一个或多个通用计算机、专用服务器计算机(包括例如PC(个人计算机)服务器、**UNIX®**服务器、中端服务器、大型计算机、机架式服务器等)、服务器集群、服务器场或任何其它适当的布置和/或组合组成。在各种实施例中,服务器1012可以适于运行在前述公开中描述的一个或多个服务或软件应用。例如,服务器1012可以与用于执行根据本公开的实施例的上述处理的服务器对应。

[0088] 服务器1012可以运行包括上面讨论的任何操作系统的操作系统,以及任何商业上可用的服务器操作系统。服务器1012还可以运行各种附加服务器应用和/或中间层应用中的任何一种,包括HTTP(超文本传输协议)服务器、FTP(文件传输协议)服务器、CGI(公共网关接口)服务器、**JAVA®**服务器、数据库服务器等。示例性数据库服务器包括但不限于可从Oracle、Microsoft、Sybase、IBM(国际商业机器)等商业获得的数据库服务器。

[0089] 在一些实现中,服务器1012可以包括一个或多个应用,以分析和整合从客户端计算设备1002、1004、1006和1008的用户接收到的数据馈送和/或事件更新。作为示例,数据馈

送和/或事件更新可以包括但不限于, **Twitter®** 推送、**Facebook®** 更新或者从一个或多个第三方信息源和连续数据流接收到的实时更新, 实时更新可以包括与传感器数据应用、金融价格收报机 (financial ticker)、网络性能测量工具 (例如, 网络监视和流量管理应用)、点击流分析工具、汽车交通监视等相关的实时事件。服务器1012还可以包括一个或多个应用, 以经由客户端计算设备1002、1004、1006和1008的一个或多个显示设备来显示数据推送和/或实时事件。

[0090] 分布式系统1000还可以包括一个或多个数据库1014和1016。数据库1014和1016可以驻留在各种位置。举例来说, 数据库1014和1016中的一个或多个可以驻留在服务器1012本地 (和/或驻留在服务器1012中) 的非瞬态存储介质上。可替代地, 数据库1014和1016可以远离服务器1012并经由基于网络的连接或专用的连接与服务器1012通信。在一组实施例中, 数据库1014和1016可以驻留在存储区域网络 (SAN) 中。类似地, 用于执行服务器1012所具有的功能的任何必要文件都可以适当地本地存储在服务器1012上和/或远程存储。在一组实施例中, 数据库1014和1016可以包括适于响应SQL格式的命令而存储、更新和检索数据的关系数据库, 诸如由Oracle提供的数据库。

[0091] 图11是根据本公开的实施例的系统环境1100的一个或多个组件的简化框图, 通过该系统环境1100, 由实施例系统的一个或多个组件提供的服务可以作为云服务提供。在所示实施例中, 系统环境1100包括可以由用户使用以与提供云服务的云基础设施系统1102交互的一个或多个客户端计算设备1104、1106和1108。客户端计算设备可以被配置为操作客户端应用, 诸如web浏览器、专有客户端应用 (例如, Oracle Forms) 或某种其它应用, 这些应用可以由客户端计算设备的用户用来与云基础设施系统1102交互以使用由云基础设施系统1102提供的服务。

[0092] 应当认识到的是, 图中描绘的云基础设施系统1102可以具有除了所描绘的那些之外的其它组件。另外, 图中所示的实施例仅是可以结合本发明的实施例的云基础设施系统的一个示例。在一些其它实施例中, 云基础设施系统1102可以具有比图中所示更多或更少的组件、可以组合两个或更多个组件、或者可以具有不同的组件配置或布置。

[0093] 客户端计算设备1104、1106和1108可以是与上面针对1002、1004、1006和1008所描述的设备类似的设备。

[0094] 虽然示例性系统环境1100被示出具有三个客户端计算设备, 但是任何数量的客户端计算设备可以被支持。诸如具有传感器的设备等的其它设备可以与云基础设施系统1102交互。

[0095] (一个或多个) 网络1110可以促进客户端1104、1106和1108与云基础设施系统1102之间的通信和数据交换。每个网络可以是本领域技术人员所熟悉的可以使用各种商业上可获得的协议 (包括上面针对 (一个或多个) 网络1010所描述的那些协议) 中的任何一种支持数据通信的任何类型的网络。

[0096] 云基础设施系统1102可以包括一个或多个计算机和/或服务器, 其可以包括上面针对服务器1012所描述的那些计算机和/或服务器。

[0097] 在某些实施例中, 由云基础设施系统提供的服务可以包括按需对云基础设施系统的用户可用的托管服务, 诸如在线数据存储和备份解决方案、基于Web的电子邮件服务、被托管的办公室 (office) 套件和文档协作服务、数据库处理、受管理的技术支持服务等。由云

基础设施系统提供的服务可以动态扩展以满足云基础设施系统的用户的需要。由云基础设施系统提供的服务的具体实例化在本文中被称为“服务实例”。一般而言,从云服务提供商的系统经由通信网络(诸如互联网)对用户可用的任何服务被称为“云服务”。通常,在公共云环境中,构成云服务提供商的系统的服务器和系统与用户自己的内部部署服务器和系统不同。例如,云服务提供商的系统可以托管应用,并且用户可以经由诸如互联网的通信网络按需订购和使用应用。

[0098] 在一些示例中,计算机网络云基础设施中的服务可以包括对存储装置、被托管的数据库、被托管的Web服务器、软件应用或由云供应商向用户提供的其它服务的受保护的计算机网络访问,或者如本领域中另外已知的。例如,服务可以包括通过互联网对云上的远程存储装置进行密码保护的访问。作为另一个示例,服务可以包括基于Web服务的被托管的关系数据库和脚本语言中间件引擎,以供联网的开发人员私有使用。作为另一个示例,服务可以包括对在云供应商的网站上托管的电子邮件软件应用的访问。

[0099] 在某些实施例中,云基础设施系统1102可以包括以自助服务、基于订阅、弹性可扩展、可靠、高度可用和安全的方式递送给客户的应用、中间件和数据库服务供应的套件。这种云基础设施系统的示例是由本受让人提供的Oracle公共云。

[0100] 在各种实施例中,云基础设施系统1102可以适于自动供应、管理和跟踪客户对由云基础设施系统1102供给的服务的订阅。云基础设施系统1102可以经由不同的部署模型来提供云服务。例如,可以依据公共云模型提供服务,其中云基础设施系统1102被销售云服务的组织拥有(例如,被Oracle拥有),并且服务对一般公众或不同行业的企业可用。作为另一个示例,可以依据私有云模型来提供服务,其中云基础设施系统1102仅针对单个组织操作,并且可以为该组织内的一个或多个实体提供服务。还可以依据社区云模型来提供云服务,其中云基础设施系统1102和由云基础设施系统1102提供的服务由相关社区中的若干组织共享。云服务还可以依据混合云模型被提供,该混合云模型是两个或更多个不同模型的组合。

[0101] 在一些实施例中,由云基础设施系统802提供的服务可以包括在软件即服务(SaaS)类别、平台即服务(PaaS)类别、基础设施即服务(IaaS)类别或包括混合服务的其它服务类别下提供的一个或多个服务。客户经由订阅订单可以订购由云基础设施系统1102提供的一个或多个服务。云基础设施系统1102然后执行处理以提供客户的订阅订单中的服务。

[0102] 在一些实施例中,由云基础设施系统1102提供的服务可以包括但不限于应用服务、平台服务和基础设施服务。在一些示例中,应用服务可以由云基础设施系统经由SaaS平台提供。SaaS平台可以被配置为提供落入SaaS类别的云服务。例如,SaaS平台可以提供在集成开发和部署平台上构建和递送按需应用套件的能力。SaaS平台可以管理和控制用于提供SaaS服务的底层软件和基础设施。通过利用由SaaS平台提供的服务,客户可以利用在云基础设施系统上执行的应用。客户可以获取应用服务,而无需客户购买单独许可证和支持。可以提供各种不同的SaaS服务。示例包括但不限于为大型组织提供销售绩效管理、企业集成和商务灵活性的解决方案的服务。

[0103] 在一些实施例中,平台服务可以由云基础设施系统经由PaaS平台提供。PaaS平台可以被配置为提供落入PaaS类别的云服务。平台服务的示例可以包括但不限于使组织(诸

如Oracle)能够在共享的公共体系架构上整合现有应用以及充分利用平台提供的共享服务来构建新应用的能力的服务。PaaS平台可以管理和控制用于提供PaaS服务的底层软件和基础设施。客户可以获取由云基础架构系统提供的PaaS服务,而无需客户购买单独的许可证和支持。平台服务的示例包括但不限于Oracle Java云服务(JCS)、Oracle数据库云服务(DBCS)等。

[0104] 通过利用由PaaS平台提供的服务,客户可以采用由云基础设施系统支持的编程语言和工具,并且还控制所部署的服务。在一些实施例中,由云基础设施系统提供的平台服务可以包括数据库云服务、中间件云服务(例如,Oracle融合中间件服务)和Java云服务。在一个实施例中,数据库云服务可以支持共享服务部署模型,该模型使得组织能够汇集数据库资源并且以数据库云的形式向客户供应数据库即服务。中间件云服务可以为客户提供开发和部署各种商务应用的平台,并且Java云服务可以为客户提供在云基础设施系统中部署Java应用的平台。

[0105] 各种不同的基础设施服务可以由云基础设施系统中的IaaS平台提供。基础设施服务促进底层计算资源(诸如存储装置、网络和其它基础计算资源)的管理和控制,以供客户利用由SaaS平台和PaaS平台提供的服务。

[0106] 在某些实施例中,云基础设施系统1102还可以包括基础设施资源1130,用于向云基础设施系统的客户提供用于提供各种服务的资源。在一个实施例中,基础设施资源1130可以包括预先集成和优化的硬件(诸如服务器、存储装置和联网资源)的组合,以执行由PaaS平台和SaaS平台提供的服务。

[0107] 在一些实施例中,云基础设施系统1102中的资源可以由多个用户共享并且根据需要动态重新分配。此外,可以将资源分配给在不同时区的用户。例如,云基础设施系统1130可以使在第一时间区中的第一组用户能够在指定的小时数内利用云基础设施系统的资源,并且然后使相同资源能够被重新分配给位于不同时间区的另一组用户,从而使资源的利用率最大化。

[0108] 在某些实施例中,可以提供由云基础设施系统1102的不同组件或模块以及由云基础设施系统1102提供的服务共享的多个内部共享服务1132。这些内部共享服务可以包括但不限于:安全和身份服务、集成服务、企业储存库服务、企业管理器服务、病毒扫描和白名单服务、高可用性、备份和恢复服务、启用云支持的服务、电子邮件服务、通知服务、文件传输服务等。

[0109] 在某些实施例中,云基础设施系统1102可以提供云基础设施系统中的云服务(例如,SaaS、PaaS和IaaS服务)的综合管理。在一个实施例中,云管理功能可以包括用于供应、管理和跟踪由云基础设施系统1102接收到的客户订阅等的功能。

[0110] 在一个实施例中,如图中所绘出的,云管理功能可以由一个或多个模块提供,诸如订单管理模块1120、订单编排模块1122、订单供应模块1124、订单管理和监视模块1126以及身份管理模块1128。这些模块可以包括一个或多个计算机和/或服务器或者使用一个或多个计算机和/或服务器来提供,这些计算机和/或服务器可以是通用计算机、专用服务器计算机、服务器场、服务器集群或任何其它适当的布置/或组合。

[0111] 在示例性操作1134中,使用客户端设备(诸如客户端设备1104、1106或1108)的客户可以通过请求由云基础设施系统1102提供的一个或多个服务并且下订阅由云基础设施

系统1102供应的一个或多个服务来的订单来与云基础设施系统1102交互。在某些实施例中,客户可以访问云用户界面(UI)、云UI 1112、云UI 1114和/或云UI 1116并经由这些UI下订阅订单。云基础设施系统802响应于客户下订单而接收到的订单信息可以包括识别客户以及客户想要订阅的云基础设施系统1102供应的一个或多个服务的信息。

[0112] 在客户下订单之后,经由云UI 1112、1114和/或1116接收订单信息。

[0113] 在操作1136处,订单存储在订单数据库1118中。订单数据库1118可以是由云基础设施系统1118操作和与其它系统元件一起操作的若干数据库之一。

[0114] 在操作1138处,订单信息被转发到订单管理模块1120。在一些情况下,订单管理模块1120可以被配置为执行与订单相关的计费 and 记账功能,诸如验证订单,并且在验证后,预订订单。

[0115] 在操作1140处,将关于订单的信息传送到订单编排模块1122。订单编排模块1122可以利用订单信息为客户下的订单编排服务和资源的供应。在一些情况下,订单编排模块1122可以使用订单供应模块1124的服务来编排资源的供应以支持订阅的服务。

[0116] 在某些实施例中,订单编排模块1122使得能够管理与每个订单相关联的业务流程并应用业务逻辑来确定订单是否应该进行到供应。在操作1142处,在接收到新订阅的订单时,订单编排模块1122向订单供应模块1124发送请求以分配资源并配置履行订阅订单所需的那些资源。订单供应模块1124使得能够为客户订购的服务分配资源。订单供应模块1124提供由云基础设施系统1100提供的云服务和用于供应于提供所请求的服务的资源的物理实现层之间的抽象层。因此,订单编排模块1122可以与实现细节隔离,诸如服务和资源是否实际上即时供应或预先供应并仅在请求后才分配/指派。

[0117] 在操作1144处,一旦供应了服务和资源,就可以通过云基础设施系统1102的订单供应模块1124向客户端设备1104、1106和/或1108上的客户发送所提供服务的通知。

[0118] 在操作1146处,订单管理和监视模块1126可以管理和跟踪客户的订阅订单。在一些情况下,订单管理和监视模块1126可以被配置为收集订阅订单中的服务的使用统计,诸如,所使用的存储量、传输的数据量、用户的数量,以及系统运行时间和系统停机时间量。

[0119] 在某些实施例中,云基础设施系统1100可以包括身份管理模块1128。身份管理模块1128可以被配置为提供身份服务,诸如云基础设施系统1100中的访问管理和授权服务。在一些实施例中,身份管理模块1128可以控制关于希望利用由云基础设施系统1102提供的服务的客户的信息。这样的信息可以包括认证这些客户的身份的信息以及描述这些客户被授权相对于各种系统资源(例如,文件、目录、应用、通信端口、存储器段等)执行哪些动作的信息。身份管理模块1128还可以包括对关于每个客户的描述性信息以及关于如何和由谁来访问和修改这些描述性信息的管理。

[0120] 图12图示了其中可以实现本发明的各种实施例的示例性计算机系统1200。系统1200可以用于实现上述计算机系统中的一个。如图所示,计算机系统1200包括经由总线子系统1202与多个外围子系统通信的处理单元1204。这些外围子系统可以包括处理加速单元1206、I/O子系统1208、存储子系统1218和通信子系统1224。存储子系统1218包括有形计算机可读存储介质1222和系统存储器1210。

[0121] 总线子系统1202提供用于让计算机系统1200的各种组件和子系统按意图彼此进行通信的机制。虽然总线子系统1202被示意性地示出为单条总线,但是总线子系统的替代

实施例可以利用多条总线。总线子系统1202可以是若干种类型的总线结构中的任何类型，这些总线类型包括存储器总线或存储器控制器、外围总线、以及使用各种总线体系架构中的任何体系架构的局部总线。例如，这种体系架构可以包括工业标准体系架构 (ISA) 总线、微通道体系架构 (MCA) 总线、增强型ISA (EISA) 总线、视频电子标准协会 (VESA) 局部总线和外围组件互连 (PCI) 总线，这些总线可以被实现为按IEEE P1386.1标准制造的Mezzanine总线。

[0122] 可以被实现为一个或多个集成电路 (例如，常规微处理器或微控制器) 的处理单元1204控制计算机系统1200的操作。一个或多个处理器可以被包括在处理单元1204中。这些处理器可以包括单核处理器或多核处理器。在某些实施例中，处理单元1204可以被实现为一个或多个独立的处理单元1232和/或1234，其中在每个处理单元中包括单核处理器或多核处理器。在其它实施例中，处理单元1204也可以被实现为通过将两个双核处理器集成到单个芯片中形成的四核处理单元。

[0123] 在各种实施例中，处理单元1204可以响应于程序代码执行各种程序并且可以维护多个并发执行的程序或进程。在任何给定的时间，要被执行的程序代码中的一些或全部代码可以驻留在 (一个或多个) 处理器1204中和/或存储子系统1218中。通过适当的编程，(一个或多个) 处理器1204可以提供上述各种功能。计算机系统1200可以附加地包括处理加速单元1206，该处理加速单元1206可以包括数字信号处理器 (DSP)、专用处理器等。

[0124] I/O子系统1208可以包括用户接口输入设备和用户接口输出设备。用户接口输入设备可以包括键盘、诸如鼠标或轨迹球的指向设备、结合到显示器中的触摸板或触摸屏、滚动轮、点击轮、拨盘、按钮、开关、小键盘、具有语音命令辨识系统的音频输入设备、麦克风以及其它类型的输入设备。用户接口输入设备可以包括，例如，运动感测和/或手势辨识设备，诸如Microsoft **Kinect®**运动传感器，该运动传感器使得用户能够通过使用手势和语音命令的自然用户接口来控制诸如Microsoft **Xbox®**360游戏控制器的输入设备并与输入设备交互。用户接口输入设备也可以包括眼睛姿势辨识设备，诸如从用户检测眼睛活动 (例如，当拍摄照片和/或进行菜单选择时的“眨眼”) 并且将眼睛姿势变换为进入输入设备 (例如，Google **Glass®**) 中的输入的Google **Glass®**眨眼检测器。此外，用户接口输入设备可以包括使用户能够通过语音命令与语音辨识系统 (例如，**Siri®**导航器) 交互的语音辨识感测设备。

[0125] 用户接口输入设备也可以包括但不限于：三维 (3D) 鼠标、操纵杆或指向棒 (pointing stick)、游戏面板和绘图板，以及音频/视觉设备，诸如扬声器、数码相机、数码摄录机、便携式媒体播放器、网络摄像头、图像扫描仪、指纹扫描仪、条形码阅读器3D扫描仪、3D打印机、激光测距仪和视线跟踪设备。此外，用户接口输入设备可以包括，例如，医学成像输入设备，诸如计算机断层扫描、磁共振成像、正电子发射断层显像、医疗超声设备。用户接口输入设备也可以包括，例如，诸如MIDI键盘、数字乐器等的音频输入设备。

[0126] 用户接口输出设备可以包括显示子系统、指示灯，或者诸如音频输出设备的非可视显示器等。显示子系统可以是阴极射线管 (CRT)、诸如使用液晶显示器 (LCD) 或等离子显示器的平板设备、投影设备、触摸屏等。一般而言，术语“输出设备”的使用旨在包括用于从计算机系统1200向用户或其它计算机输出信息的所有可能类型的设备和机制。例如，用户

接口输出设备可以包括但不限于：可视地传达文本、图形和音频/视频信息各种显示设备，诸如监视器、打印机、扬声器、耳机、汽车导航系统、绘图仪、语音输出设备以及调制解调器。

[0127] 计算机系统1200可以包括存储子系统1218，存储子系统1218包括被示出为当前位于系统存储器1210内的软件元件。系统存储器1210可以存储可加载并且可在处理单元1204上执行的程序指令，以及在这些程序的执行期间所生成的数据。

[0128] 取决于计算机系统1200的配置和类型，系统存储器1210可以是易失性的（诸如随机存取存储器（RAM））和/或非易失性的（诸如只读存储器（ROM）、闪存存储器等）。RAM通常包含可被处理单元1204立即访问和/或目前正在被处理单元1204操作和执行的数据和/或程序模块。在一些实现中，系统存储器1210可以包括多种不同类型的存储器，例如静态随机存取存储器（SRAM）或动态随机存取存储器（DRAM）。在一些实现中，诸如在启动期间，包含有助于在计算机系统1200内的元件之间传送信息的基本例程的基本输入/输出系统（BIOS）通常可以被存储在ROM中。作为示例但不是限制，系统存储器1210也示出了可以包括客户端应用、Web浏览器、中间层应用、关系数据库管理系统（RDBMS）等的应用程序1212，程序数据1214、以及操作系统1216。作为示例，操作系统1216可以包括各种版本的Microsoft **Windows**<sup>®</sup>，Apple **Macintosh**<sup>®</sup> 和/或Linux操作系统、各种可商业获得的 **UNIX**<sup>®</sup> 或类UNIX操作系统（包括但不限于各种GNU/Linux操作系统、Google **Chrome**<sup>®</sup> OS等）和/或诸如iOS、**Windows**<sup>®</sup> Phone、**Android**<sup>®</sup> OS、**BlackBerry**<sup>®</sup> 10 OS和 **Palm**<sup>®</sup> OS操作系统的移动操作系统。

[0129] 存储子系统1218也可以提供用于存储提供一些实施例的功能的基本编程和数据结构的有形计算机可读存储介质。当被处理器执行时提供上述功能的软件（程序、代码模块、指令）可以被存储在存储子系统1218中。这些软件模块或指令可以被处理单元1204执行。存储子系统1218也可以提供用于存储根据本发明被使用的数据的储存库。

[0130] 存储子系统1200也可以包括可被进一步连接到计算机可读存储介质1222的计算机可读存储介质读取器1220。与系统存储器1210一起并且可选地与系统存储器1210相结合，计算机可读存储介质1222可以全面地表示用于临时和/或更持久地包含、存储、发送和检索计算机可读信息的远程、本地、固定和/或可移动的存储设备加存储介质。

[0131] 包含代码或代码的一部分的计算机可读存储介质1222也可以包括本领域已知或使用的任何适当的介质，包括存储介质和通信介质，诸如但不限于：以用于信息的存储和/或传输的任何方法或技术实现的易失性和非易失性、可移动和不可移动介质。这可以包括有形的计算机可读存储介质，诸如RAM、ROM、电可擦除可编程ROM（EEPROM）、闪存存储器或其它存储器技术、CD-ROM、数字多功能盘（DVD）或其它光学储存器、磁带盒、磁带、磁盘储存装置或其它磁存储设备、或者其它有形计算机可读介质。这也可以包括非有形的计算机可读介质，诸如数据信号、数据传输，或者可以用于发送期望信息并且可以被计算机系统1200访问的任何其它介质。

[0132] 作为示例，计算机可读存储介质1222可以包括从不可移动的非易失性磁介质读取或写入到不可移动的非易失性磁介质的硬盘驱动器、从可移动的非易失性磁介质读取或写入到可移动的非易失性磁介质的磁盘驱动器、以及从可移动的非易失性光盘（诸如CD ROM、DVD

和Blu-**Ray**® (蓝光) 盘或其它光学介质) 读取或写入到可移动的非易失性光盘的光盘驱动器。计算机可读存储介质1222可以包括但不限于:**Zip**®驱动器、闪存卡、通用串行总线(USB)闪存驱动器、安全数字(SD)卡、DVD盘、数字音频带等。计算机可读存储介质1222也可以包括:基于非易失性存储器的固态驱动器(SSD)(诸如基于闪存存储器的SSD、企业闪存驱动器、固态ROM等)、基于易失性存储器的SSD(诸如固态RAM、动态RAM、静态RAM,基于DRAM的SSD、磁阻RAM(MRAM)SSD)、以及使用基于DRAM和闪存存储器的SSD的组的混合SSD。盘驱动器及其关联的计算机可读介质可以为计算机系统1200提供计算机可读指令、数据结构、程序模块以及其它数据的非易失性存储。

[0133] 通信子系统1224提供到其它计算机系统和网络的接口。通信子系统1224用作用于从其它系统接收数据和从计算机系统1200向其它系统发送数据的接口。例如,通信子系统924可以使计算机系统1200能够经由互联网连接到一个或多个设备。在一些实施例中,通信子系统1224可以包括用于访问无线语音和/或数据网络的射频(RF)收发器组件(例如,使用蜂窝电话技术、先进数据网络技术,诸如3G、4G或EDGE(用于全球演进的增强型数据速率)、WiFi(IEEE 1202.11系列标准)或其它移动通信技术、或者其任何组合)、全球定位系统(GPS)接收器组件和/或其它组件。在一些实施例中,作为无线接口的附加或者替代,通信子系统1224可以提供有线网络连接(例如,以太网)。

[0134] 在一些实施例中,通信子系统1224也可以代表可以使用计算机系统1200的一个或多个用户来接收以结构化和/或非结构化数据馈送1226、事件流1228、事件更新1230等形式的输入通信。

[0135] 作为示例,通信子系统1224可以被配置为实时地从社交网络和/或其它通信服务的用户接收数据馈送1226,诸如**Twitter**®馈送、**Facebook**®更新、诸如丰富站点摘要(RSS)馈送的web馈送和/或来自一个或多个第三方信息源的实时更新。

[0136] 此外,通信子系统1224也可以被配置为接收以连续数据流形式的数据,该数据可以包括本质上可以是连续的或无界的没有明确终止的实时事件的事件流1228和/或事件更新1230。生成连续数据的应用的示例可以包括,例如,传感器数据应用、金融价格收报机、网络性能测量工具(例如,网络监视和流量管理应用)、点击流分析工具、汽车流量监视等。

[0137] 通信子系统1224也可以被配置为向一个或多个数据库输出结构化和/或非结构化的数据馈送1226、事件流1228、事件更新1230等,这一个或多个数据库可以与耦合到计算机系统1200的一个或多个流式数据源计算机进行通信。

[0138] 计算机系统1200可以是各种类型中的一种类型,包括手持便携式设备(例如,**iPhone**®蜂窝电话、**iPad**®计算平板电脑、PDA)、可穿戴设备(例如,Google**Glass**®头戴式显示器)、PC、工作站、大型机、信息站(kiosk)、服务器机架、或任何其它数据处理系统。

[0139] 由于计算机和网络的不断变化的本质,在图中绘出的计算机系统1200的描述仅旨在作为具体的示例。具有比图中绘出的系统更多或更少组件的许多其它配置是可能的。例如,定制的硬件也可以被使用和/或特定的元素可以在硬件、固件、软件(包括小程序应用(applets))或其组合中实现。另外,可以采用到诸如网络输入/输出设备的其它计算设备的连接。基于本文提供的公开内容和教导,本领域普通技术人员将认识到实现各种实施例的其它方式和/或方法。

[0140] 在前述说明书中,参考本发明的各方面的具体实施例对本发明的各方面进行了描述,但是本领域技术人员将认识到的是,本发明不限于此。上述发明的各个特征和方面可以被单独使用或联合使用。另外,在不脱离本说明书的更广泛精神和范围的情况下,实施例可以在除本文所述的那些环境和应用之外的任何数目的环境和应用中被使用。相应地,本说明书和附图应当被认为是说明性的而不是限制性的。

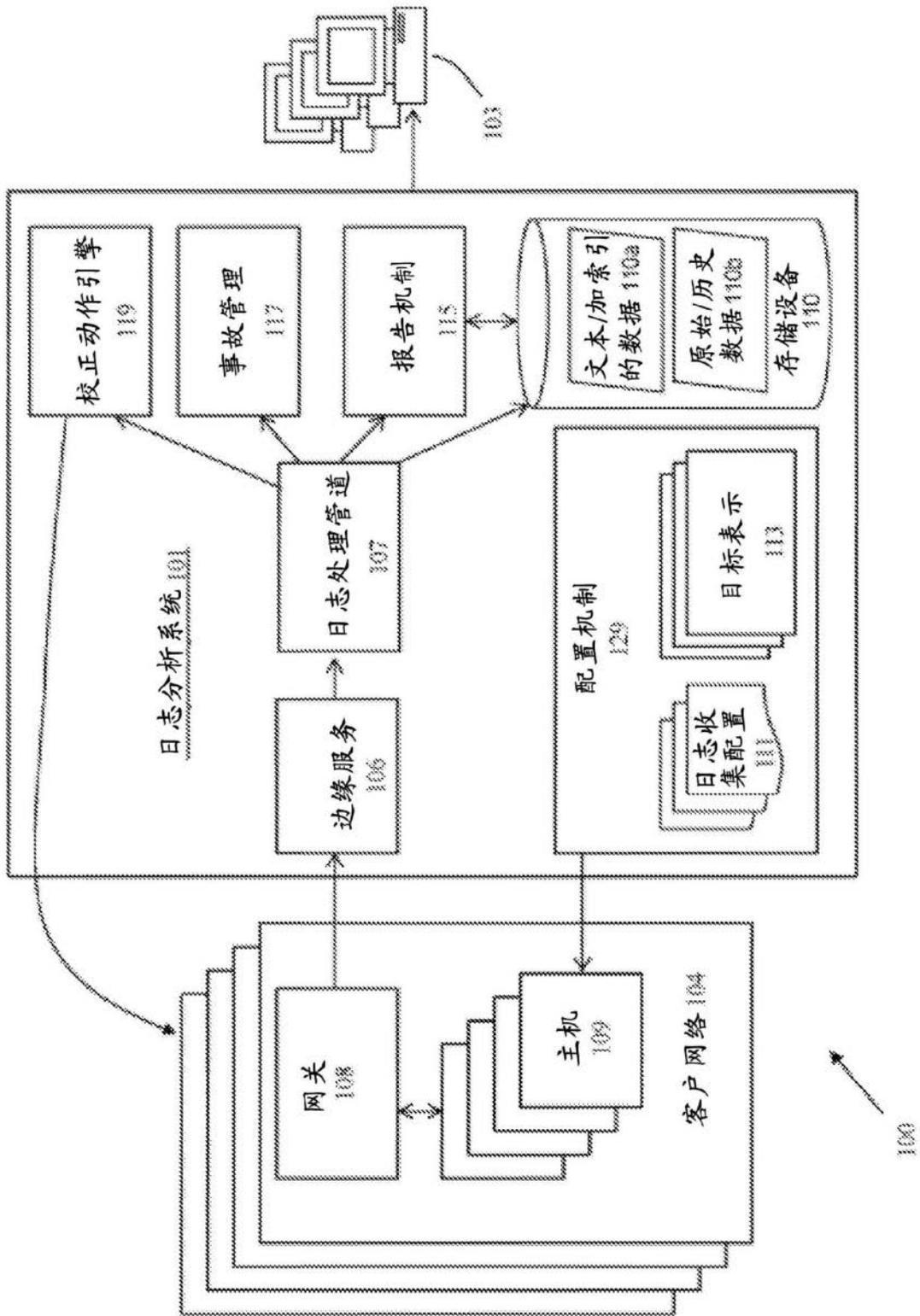


图1A

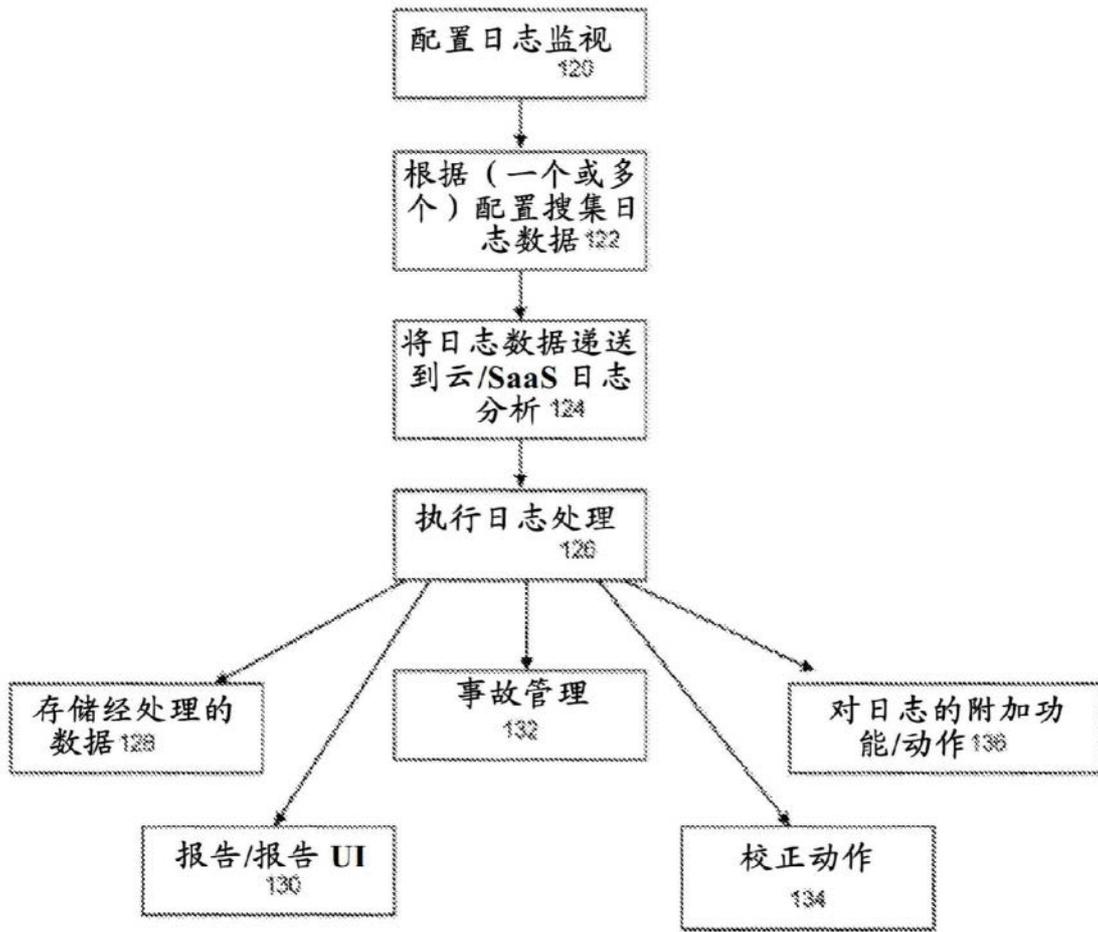


图1B

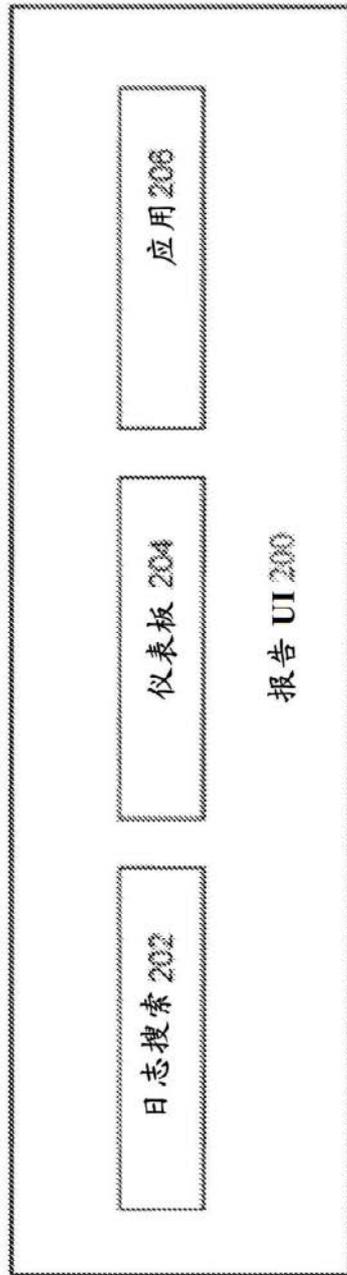


图2

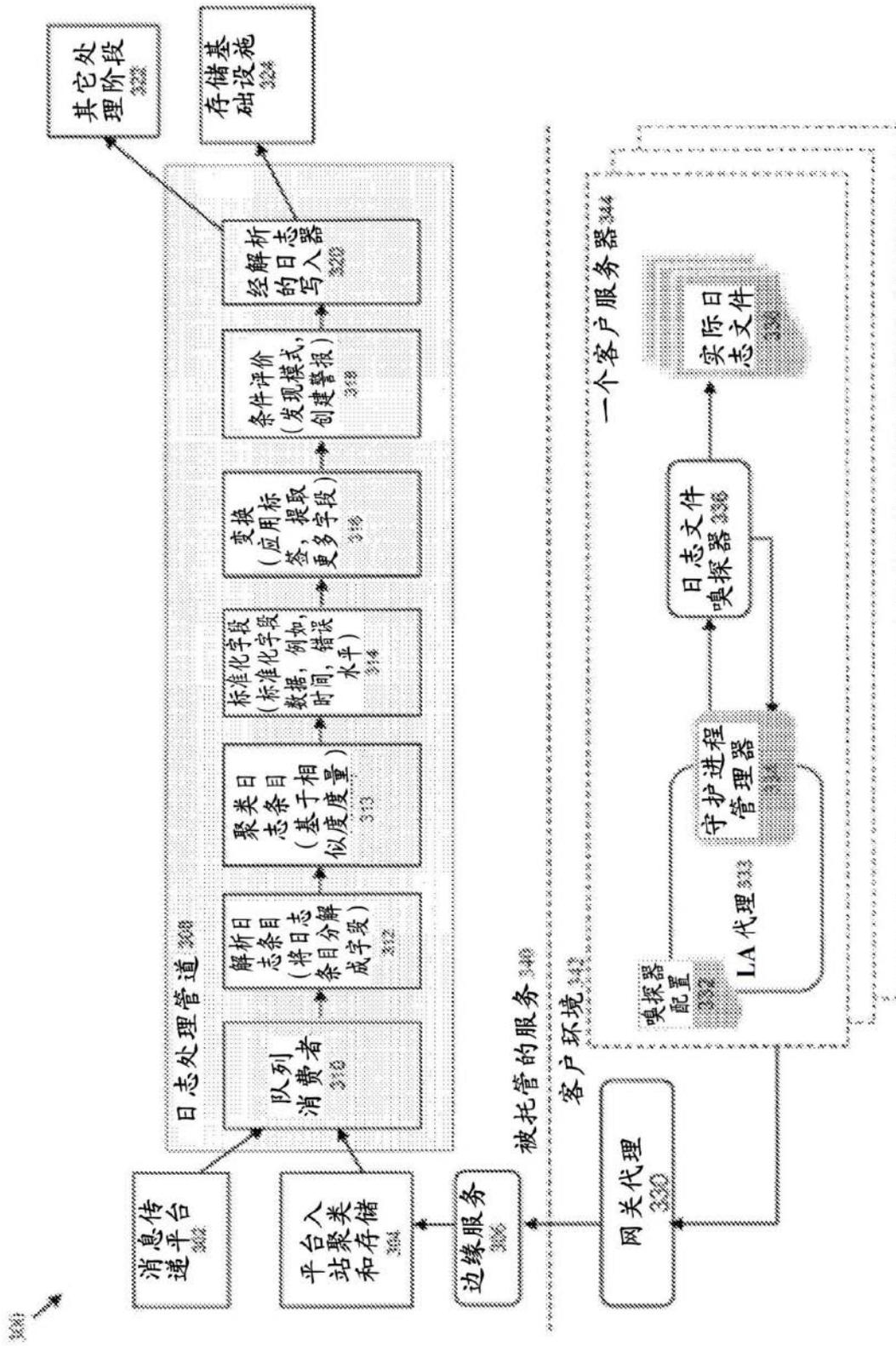


图3A

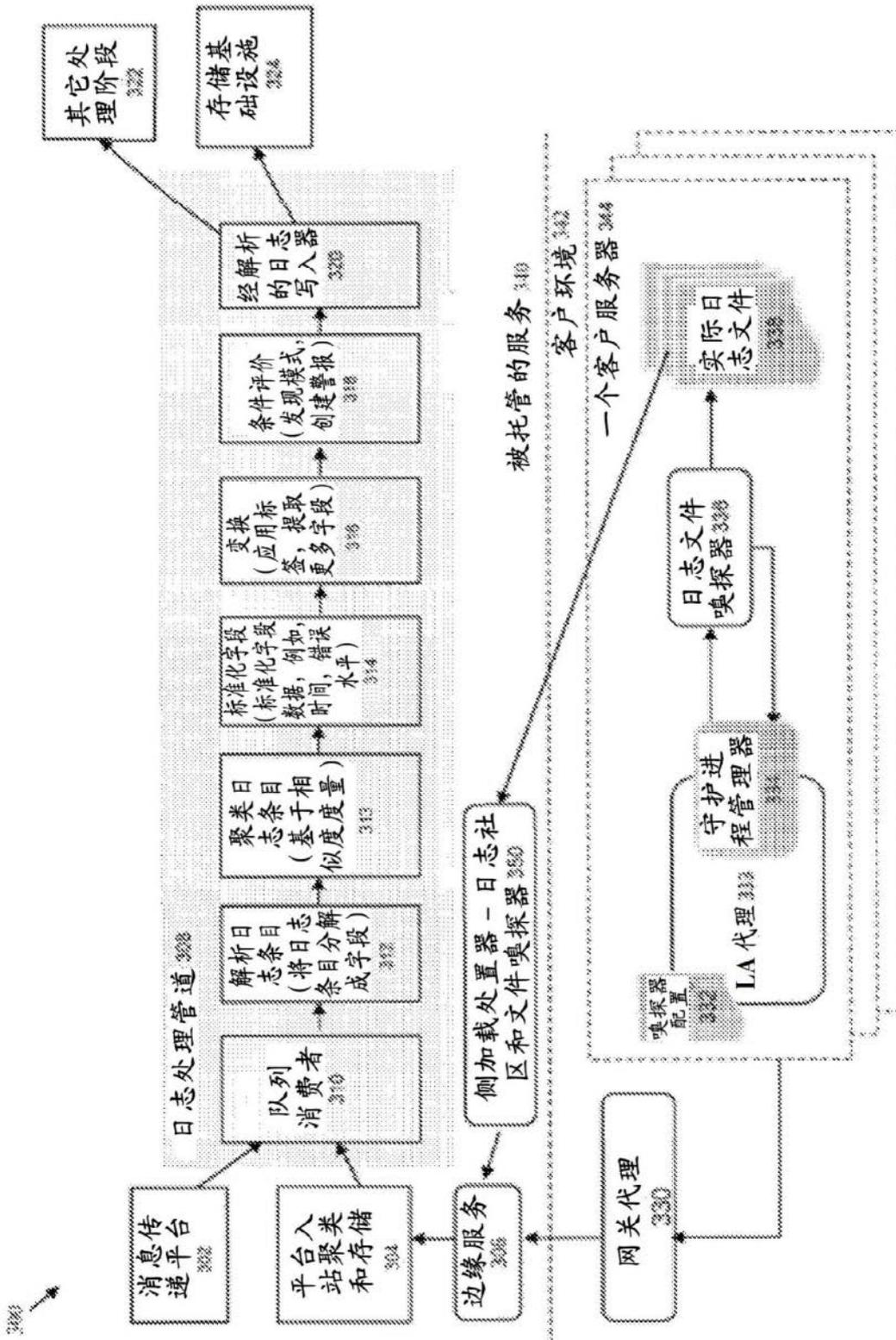


图3B

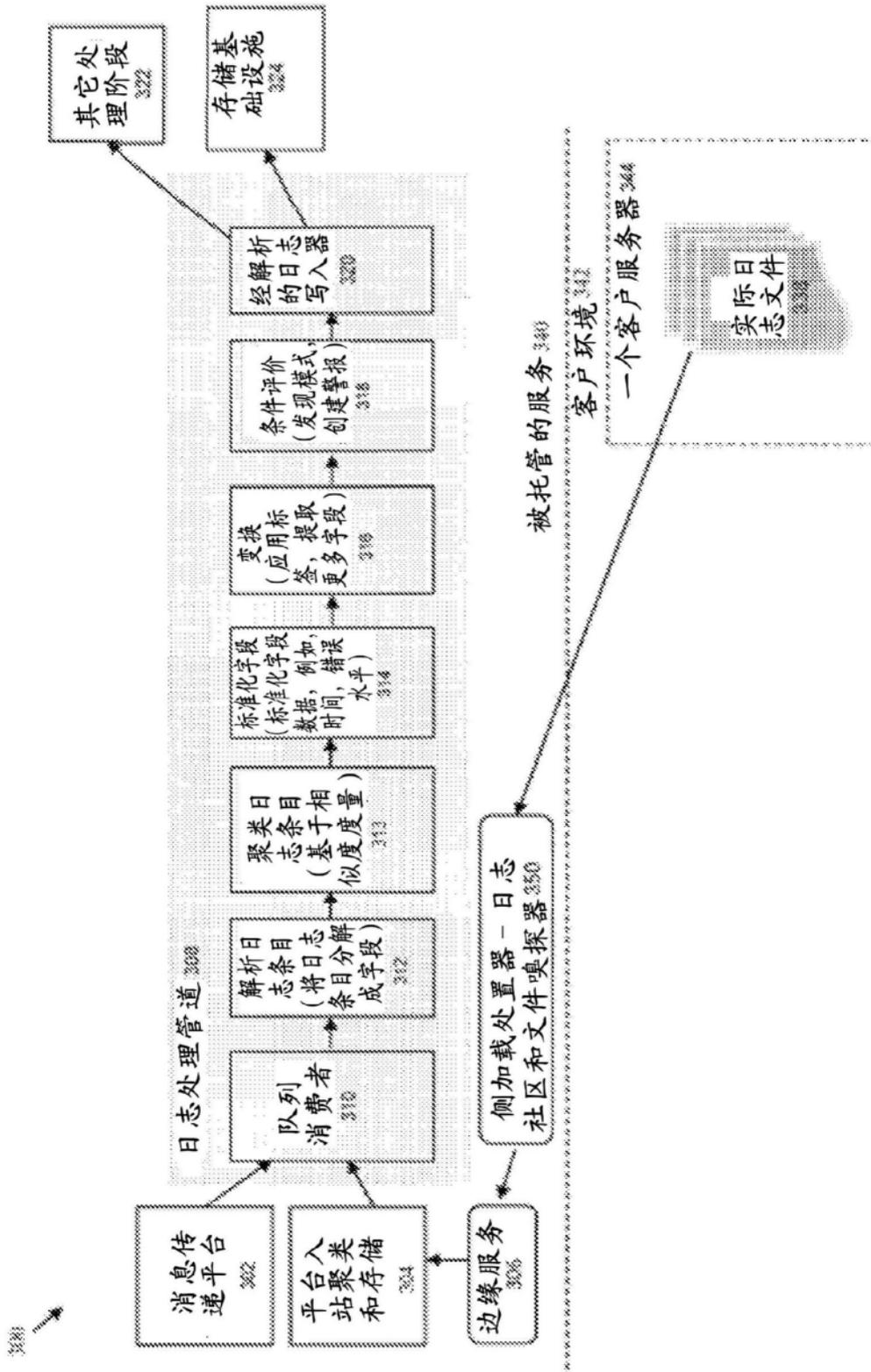


图3C

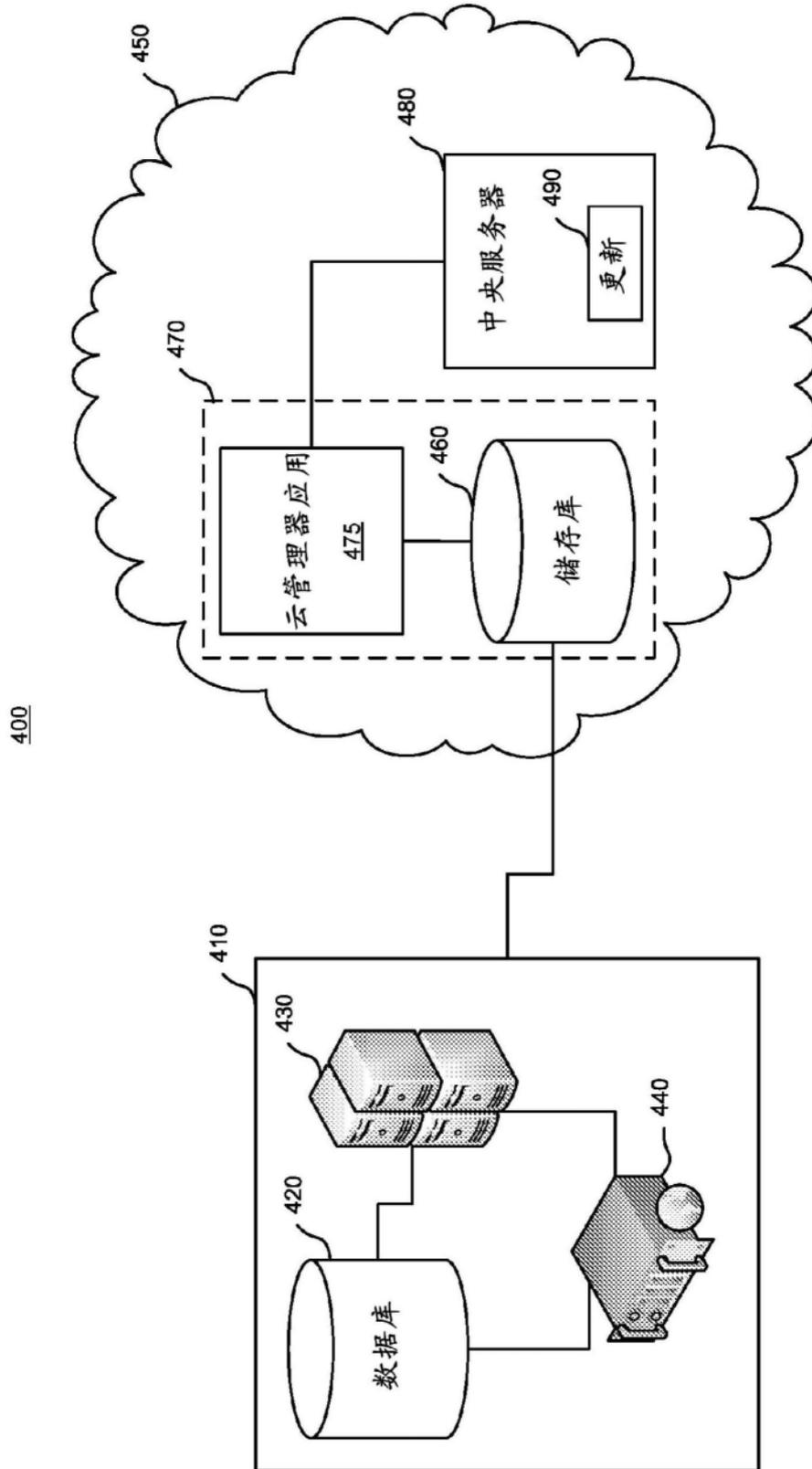


图4

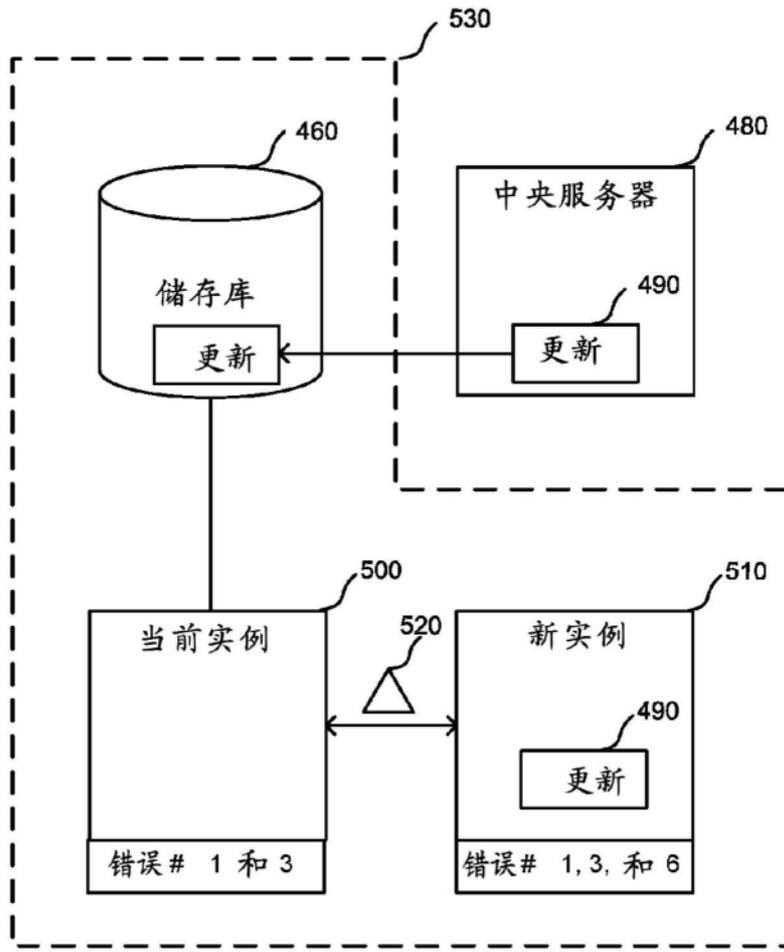
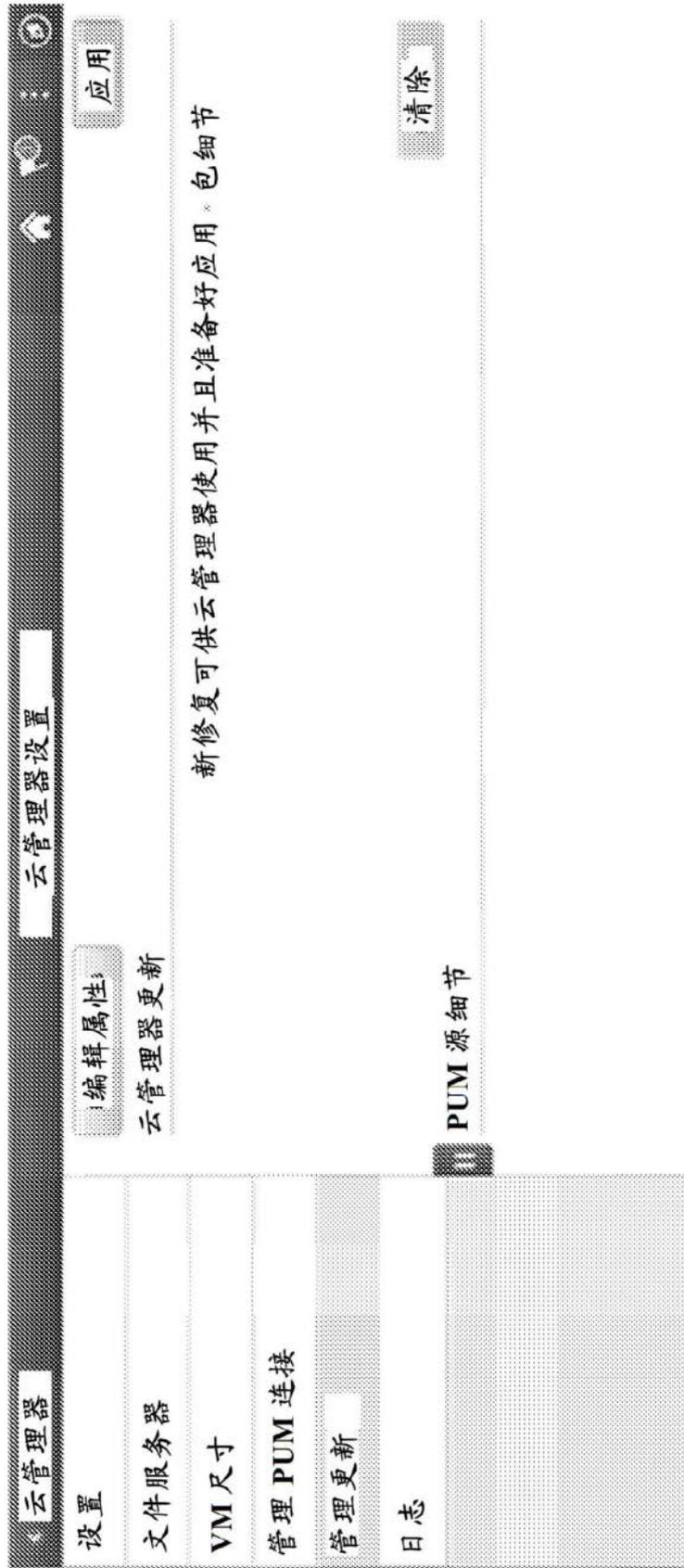


图5



600

图6

92 行

错误号	错误描述
1 280383347	在点击“作为小组发布”按钮后 peoplecode 错误
2 280417451	流内容 经典特征登记
3 280532225	在 FUNCLIB 记录中递交的公共函数
4 280537043	流内容-分开经典和流功能
5 130548001	WPTG_LEGAL:OSN:错误的版权串:PSFT/公共部件(EO):ICE/EOC
6 130548043	当批准处理建立定义 ID 改变, 对 PO SELF- APPR 产生错误
7 211677209	EC:凭单批准历史页面未显示最近的批准流程
8 211873402	在 PT8.54 应用程序版本 9.1 中工作流图标未正确地对准
9 211892525	在事件汇总表, 表格标题丢失, 提示未发现消息 ( 18111,3024 )
10 220003198	在使用设计表单向导时未显示 OK 按钮
11 220003198	在使用设计表单向导时未显示 OK 按钮

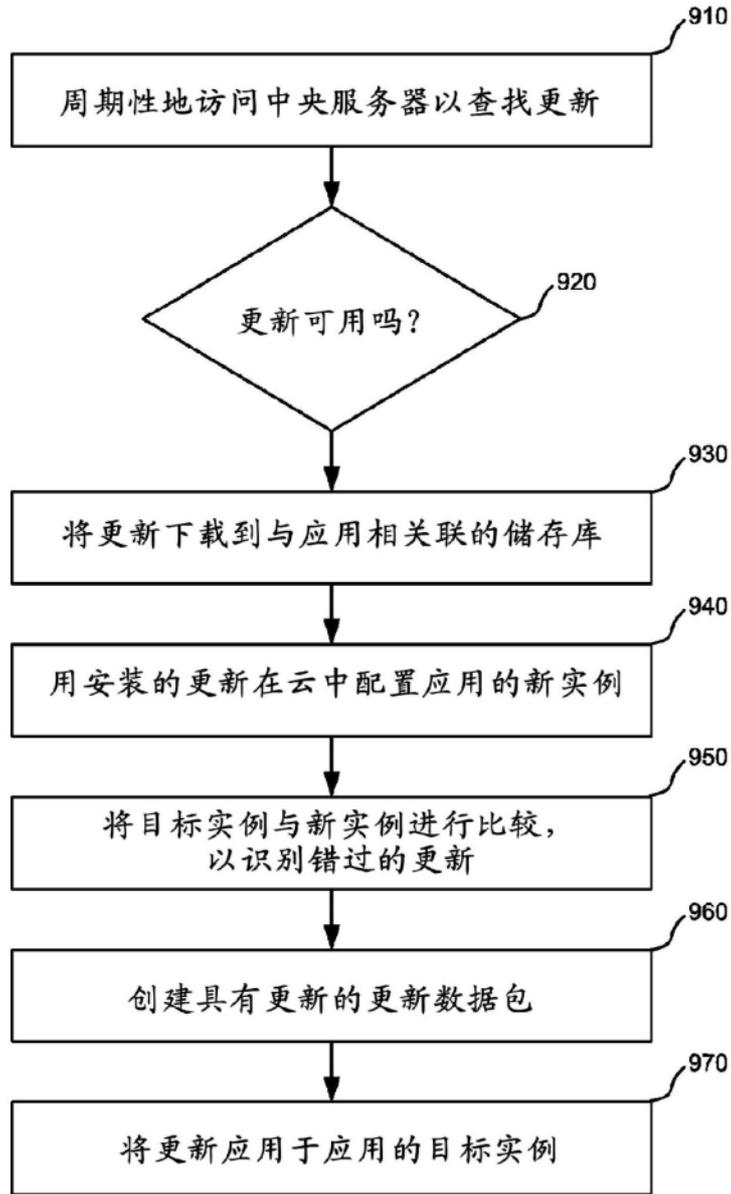
700

图7

云管理器设计		
编辑属性		
云管理器更新		
没有更新可供云管理器使用		
更新步骤	状态	步骤细节
1 创建新的环境模板以部署 PUM 源	跳过	
2 部署 PUM 源环境	成功	
3 安装改变助手并配置 PUM 源和目标数据库	成功	
4 向 PUM 源应用 PRP	成功	
5 定义“make me current”改变包	成功	
6 从定义中创建“make me current”改变包	成功	
7 向云管理器目标数据库应用改变包	成功	
8 更新云管理器目标上的文件	成功	

8行

图8



900

图9

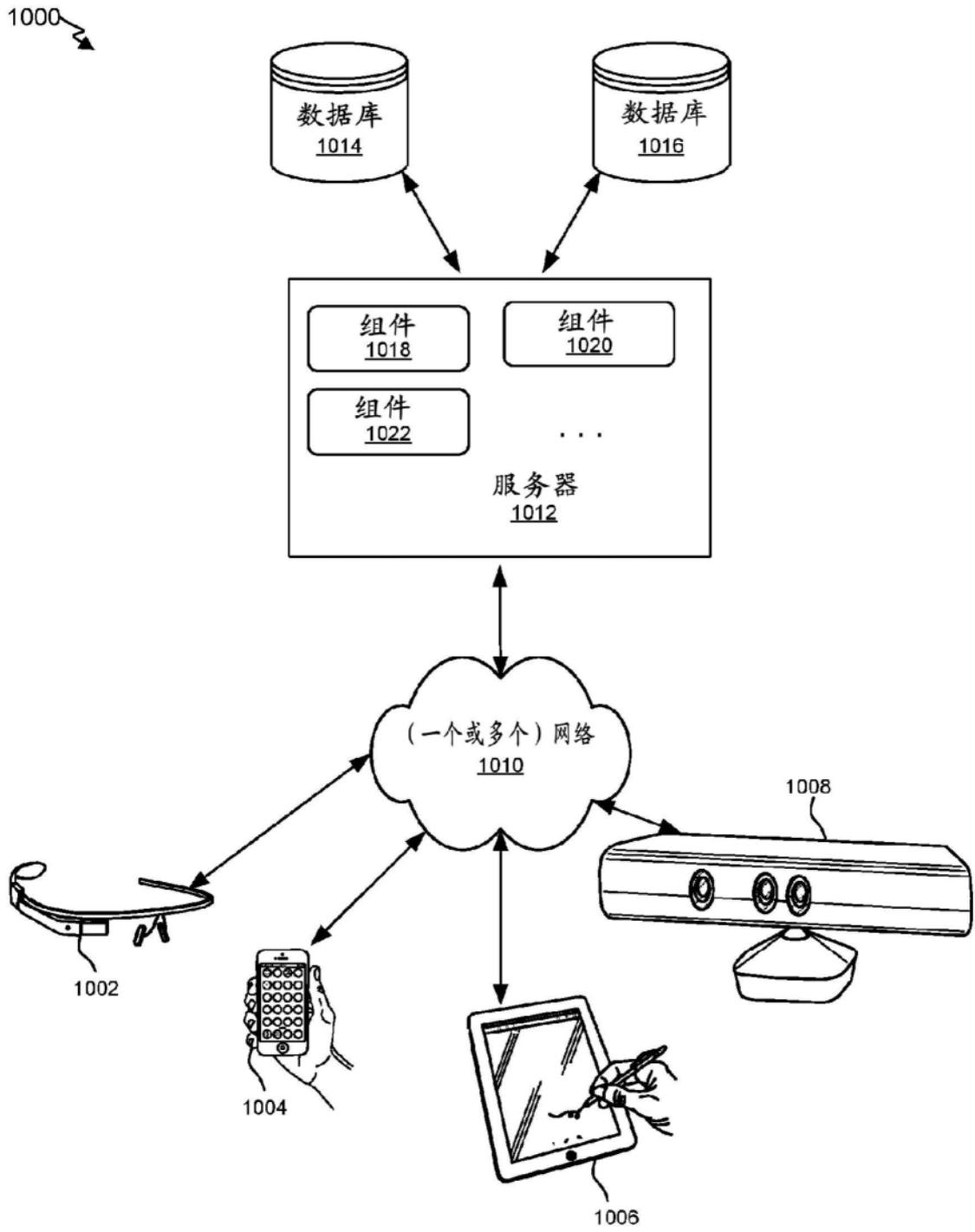


图10

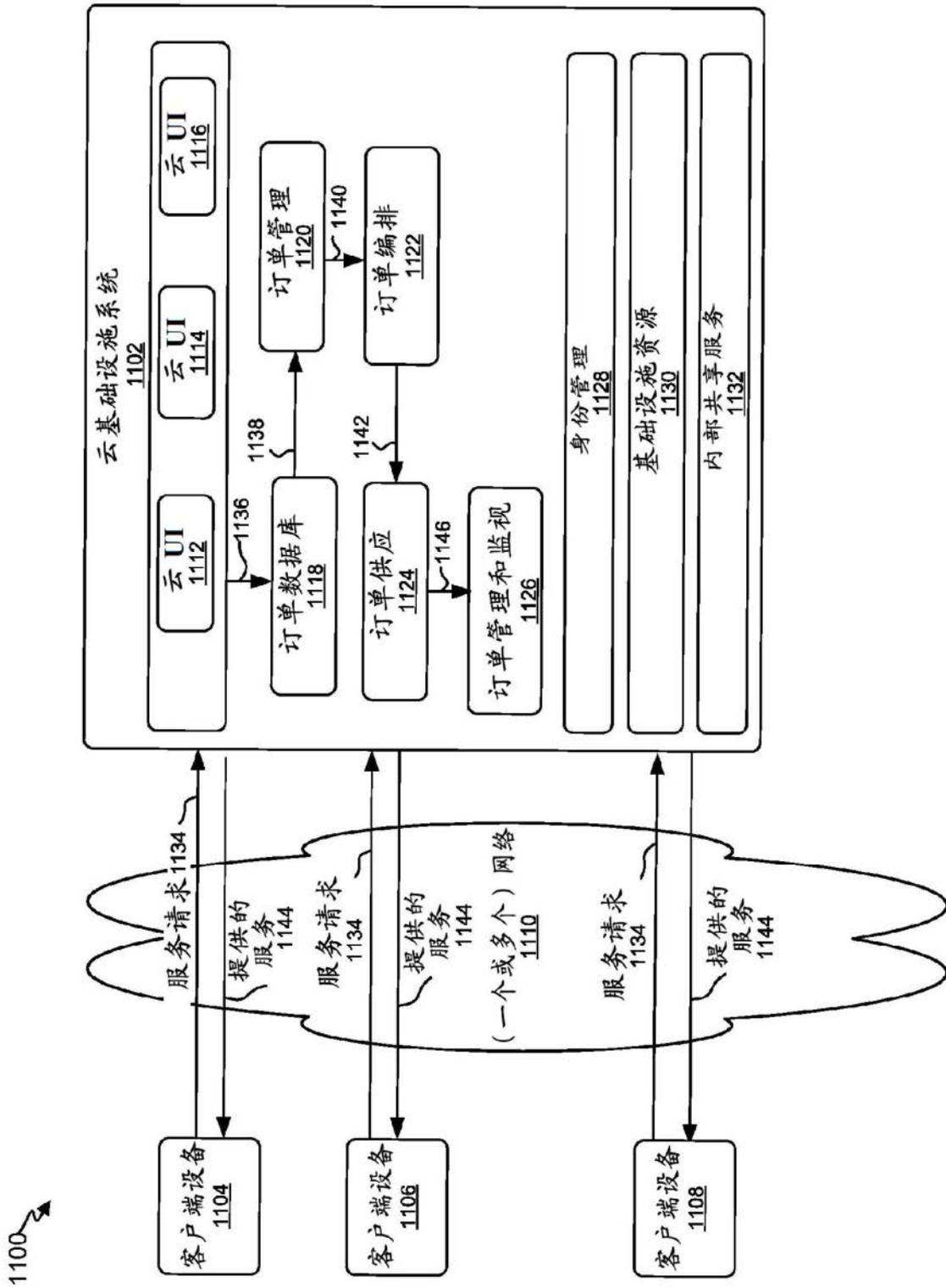


图11

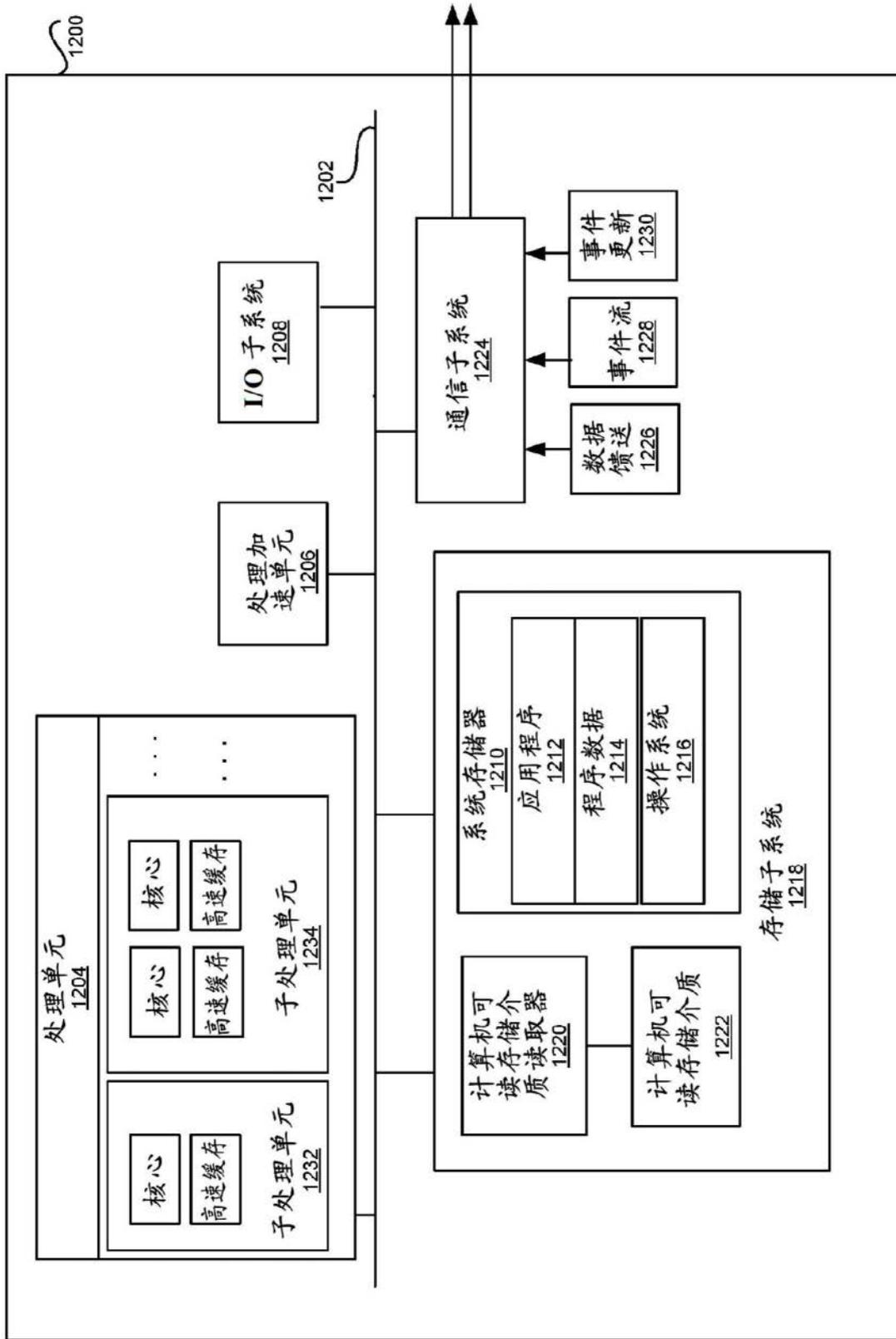


图12