(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 118018591 A
(43)申请公布日 2024.05.10

(21)申请号 202211587987.2

(22)申请日 2022.12.09

(30)优先权数据
17/983,045 2022.11.08 US

(71)申请人 SAP欧洲公司
地址 德国瓦尔多夫

(72)发明人 P.埃伯莱因

(74)专利代理机构 北京市柳沈律师事务所 11105
专利代理师 邵亚丽

(51)Int.Cl.
*H04L 67/142*(2022.01)
*H04L 67/143*(2022.01)
*H04L 67/10*(2022.01)
*H04L 67/50*(2022.01)

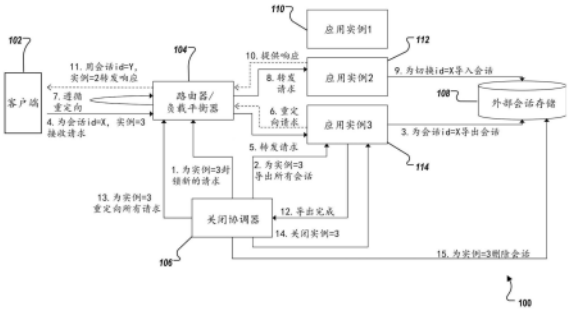
权利要求书4页 说明书16页 附图3页

(54)发明名称

具有会话切换的云计算应用的协调关闭

(57)摘要

在一个实现中,关闭协调器向路由器发送指令,以不再向应用实例分配新的会话,其中应用实例将被关闭。关闭协调器向应用实例发送请求,以导出用于切换至新的应用实例的关联的会话。关闭协调器接收来自应用实例的指示,即应用实例不能处理进一步的请求,因为所有关联的会话已经导出至外部会话存储。关闭协调器指示路由器将对应用实例的请求重定向至新的应用实例。关闭协调器关闭应用实例。在限定的超时时段之后,关闭协调器从外部会话存储中删除所有关联的会话的剩余会话。




CN 118018591 A

1.一种计算机实现的方法,包括:

由关闭协调器向路由器发送指令,以不再向应用实例分配新的会话,其中所述应用实例将被关闭;

由所述关闭协调器向所述应用实例发送请求,以导出用于切换至新的应用实例的关联的会话;

由所述关闭协调器从所述应用实例接收指示,所述指示表明所述应用实例不能处理进一步的请求,因为所有关联的会话已经导出至外部会话存储;

由所述关闭协调器指示所述路由器,将对所述应用实例的请求重定向至所述新的应用实例;

由所述关闭协调器关闭所述应用实例;以及

在限定的超时时段之后,由所述关闭协调器从所述外部会话存储中删除所有关联的会话的剩余会话。

2.根据权利要求1所述的计算机实现的方法,包括:

由所述路由器从可用应用实例的列表中移除所述应用实例,以防止在所述应用实例上创建新的会话。

3.根据权利要求1所述的计算机实现的方法,包括:

由所述应用实例迭代所有会话以创建可导出会话;以及

导出所述可导出会话至所述外部会话存储。

4.根据权利要求1所述的计算机实现的方法,包括:

由所述路由器从客户端接收具有会话标识符(ID)和所述应用实例标识的请求;

从所述路由器向所述应用实例转发所述请求;

由所述应用实例向所述路由器响应重定向请求,所述重定向请求包括所述路由器的原始地址、无效的会话缓存文件、包括所述会话ID的切换缓存文件、以及指示所述客户端何时应该重试所述请求的重试后值;以及

从所述路由器向所述客户端发送所述重定向请求。

5.根据权利要求4所述的计算机实现的方法,包括:

由所述客户端等待由所述重试后值规定的持续时间,以允许完成从所述应用实例导出会话状态至所述外部会话存储;

由所述客户端向所述路由器重试所述请求作为新的请求,其中所述请求不包括会话缓存文件,而包括具有所述切换缓存文件值的新的切换缓存文件;

由所述路由器从所述客户端接收所述新的请求;

由所述路由器检测所述新的请求不包括会话缓存文件;以及

由所述路由器生成新的应用实例,利用所述新的应用实例分派所述新的请求。

6.根据权利要求5所述的计算机实现的方法,包括:

由所述新的应用实例从所述路由器接收所述新的请求;

由所述新的应用实例检测不存在会话缓存文件;

由所述新的应用实例生成新的会话;

由所述新的应用实例用标识所述新的会话的新的会话ID来设置新的会话缓存文件;

由所述新的应用实例检测新的切换缓存文件的存在;

由所述新的应用实例使用所述新的切换缓存文件来检索被包括在所述新的切换缓存文件中的所述会话ID;以及

导入存储的与所述应用实例相关联的会话状态至所述新的会话。

7.根据权利要求6所述的计算机实现的方法,包括:

由所述新的应用实例处理所述新的请求;

由所述新的应用实例向所述路由器响应新的响应,所述新的响应包括所述新的会话缓存文件中的所述新的会话ID和具有无效状态的所述新的切换缓存文件;

由所述路由器向所述客户端转发所述新的响应;以及

由所述路由器作为粘性会话更新所述新的会话ID与所述新的应用实例之间的映射。

8.一种存储可由计算机系统执行的一个或多个指令的非暂时性计算机可读介质,其执行的操作包括:

由关闭协调器向路由器发送指令,以不再向应用实例分配新的会话,其中所述应用实例将被关闭;

由所述关闭协调器向所述应用实例发送请求,以导出用于切换至新的应用实例的关联的会话;

由所述关闭协调器从所述应用实例接收指示,所述指示表明所述应用实例不能处理进一步的请求,因为所有关联的会话已经导出至外部会话存储;

由所述关闭协调器指示所述路由器,将对所述应用实例的请求重定向至所述新的应用实例;

由所述关闭协调器关闭所述应用实例;以及

在限定的超时时段之后,由所述关闭协调器从所述外部会话存储中删除所有关联的会话的剩余会话。

9.根据权利要求8所述的非暂时性计算机可读介质,包括可由计算机系统执行的一个或多个指令,以执行操作用于由所述路由器从可用应用实例的列表中移除所述应用实例以防止在所述应用实例上创建新的会话。

10.根据权利要求8所述的非暂时性计算机可读介质,包括可由计算机系统执行的一个或多个指令,以执行操作用于:

由所述应用实例迭代所有会话以创建可导出会话;以及

导出所述可导出会话至所述外部会话存储。

11.根据权利要求8所述的非暂时性计算机可读介质,包括可由计算机系统执行的一个或多个指令,以执行操作用于:

由所述路由器从客户端接收具有会话标识符(ID)和所述应用实例标识的请求;

从所述路由器向所述应用实例转发所述请求;

由所述应用实例向所述路由器响应重定向请求,所述重定向请求包括所述路由器的原始地址、无效的会话缓存文件、包括所述会话ID的切换缓存文件、以及指示所述客户端何时应该重试所述请求的重试后值;以及

从所述路由器向所述客户端发送所述重定向请求。

12.根据权利要求11所述的非暂时性计算机可读介质,包括可由计算机系统执行的一个或多个指令,以执行操作用于:

由所述客户端等待由所述重试后值规定的持续时间,以允许完成从所述应用实例导出会话状态至所述外部会话存储;

由所述客户端向所述路由器重试所述请求作为新的请求,其中所述请求不包括会话缓存文件,而包括具有所述切换缓存文件值的新的切换缓存文件;

由所述路由器从所述客户端接收所述新的请求;

由所述路由器检测所述新的请求不包括会话缓存文件;以及

由所述路由器生成新的应用实例,利用所述新的应用实例分派所述新的请求。

13.根据权利要求12所述的非暂时性计算机可读介质,包括可由计算机系统执行的一个或多个指令,以执行操作用于:

由所述新的应用实例从所述路由器接收所述新的请求;

由所述新的应用实例检测不存在会话缓存文件;

由所述新的应用实例生成新的会话;

由所述新的应用实例用标识所述新的会话的新的会话ID来设置新的会话缓存文件;

由所述新的应用实例检测新的切换缓存文件的存在;

由所述新的应用实例使用所述新的切换缓存文件来检索被包括在所述新的切换缓存文件中的所述会话ID;以及

导入存储的与所述应用实例相关联的会话状态至所述新的会话。

14.根据权利要求13所述的非暂时性计算机可读介质,包括可由计算机系统执行的一个或多个指令,以执行操作用于:

由所述新的应用实例处理所述新的请求;

由所述新的应用实例向所述路由器响应新的响应,所述新的响应包括所述新的会话缓存文件中的所述新的会话ID和具有无效状态的所述新的切换缓存文件;

由所述路由器向所述客户端转发所述新的响应;以及

由所述路由器作为粘性会话更新所述新的会话ID与所述新的应用实例之间的映射。

15.一种计算机实现的系统,包括:

一个或多个计算机;以及

一个或多个计算机存储器装置,与所述一个或多个计算机可互操作地耦合,并且具有存储一个或多个指令的有形的非暂时性的机器可读介质,当由所述一个或多个计算机执行所述指令时,所述指令执行一个或多个操作,包括:

由关闭协调器向路由器发送指令,以不再向应用实例分配新的会话,其中所述应用实例将被关闭;

由所述关闭协调器向所述应用实例发送请求,以导出用于切换至新的应用实例的关联的会话;

由所述关闭协调器从所述应用实例接收指示,所述指示表明所述应用实例不能处理进一步的请求,因为所有关联的会话已经导出至外部会话存储;

由所述关闭协调器指示所述路由器,将对所述应用实例的请求重定向至所述新的应用实例;

由所述关闭协调器关闭所述应用实例;以及

在限定的超时时段之后,由所述关闭协调器从所述外部会话存储中删除所有关联的会

话的剩余会话。

16.根据权利要求15所述的计算机实现的系统,包括一个或多个指令,当由所述一个或多个计算机执行所述指令时,所述指令执行一个或多个操作用于由所述路由器从可用应用实例的列表中移除所述应用实例以防止在所述应用实例上创建新的会话。

17.根据权利要求15所述的计算机实现的系统,包括一个或多个指令,当由所述一个或多个计算机执行所述指令时,所述指令执行一个或多个操作用于:

由所述应用实例迭代所有会话以创建可导出会话;以及

导出所述可导出会话至所述外部会话存储。

18.根据权利要求15所述的计算机实现的系统,包括一个或多个指令,当由所述一个或多个计算机执行所述指令时,所述指令执行一个或多个操作用于:

由所述路由器从客户端接收具有会话标识符(ID)和所述应用实例标识的请求;

从所述路由器向所述应用实例转发所述请求;

由所述应用实例向所述路由器响应重定向请求,所述重定向请求包括所述路由器的原始地址、无效的会话缓存文件、包括所述会话ID的切换缓存文件、以及指示所述客户端何时应该重试所述请求的重试后值;以及

从所述路由器向所述客户端发送所述重定向请求。

19.根据权利要求18所述的计算机实现的系统,包括一个或多个指令,当由所述一个或多个计算机执行所述指令时,所述指令执行一个或多个操作用于:

由所述客户端等待由所述重试后值规定的持续时间,以允许完成从所述应用实例导出会话状态至所述外部会话存储;

由所述客户端向所述路由器重试所述请求作为新的请求,其中所述请求不包括会话缓存文件,而包括具有所述切换缓存文件值的新的切换缓存文件;

由所述路由器从所述客户端接收所述新的请求;

由所述路由器检测所述新的请求不包括会话缓存文件;以及

由所述路由器生成新的应用实例,利用所述新的应用实例分派所述新的请求。

20.根据权利要求19所述的计算机实现的系统,包括一个或多个指令,当由所述一个或多个计算机执行所述指令时,所述指令执行一个或多个操作用于:

由所述新的应用实例从所述路由器接收所述新的请求;

由所述新的应用实例检测不存在会话缓存文件;

由所述新的应用实例生成新的会话;

由所述新的应用实例用标识所述新的会话的新的会话ID来设置新的会话缓存文件;

由所述新的应用实例检测新的切换缓存文件的存在;

由所述新的应用实例使用所述新的切换缓存文件来检索被包括在所述新的切换缓存文件中的所述会话ID;以及

导入存储的与所述应用实例相关联的会话状态至所述新的会话。

# 具有会话切换的云计算应用的协调关闭

## 背景技术

[0001] 在云计算环境中，数量可变的应用实例并行处理客户端请求。对于某些事件，诸如缩减需要较少应用实例的部署、软件维护或可用性区域撤离(在数据中心基础设施出现问题的情况下)，必须关闭一些应用实例，并且需要将它们的工作负载重新分布至其他实例。尽管云应用应该是完全无状态的，这使得能够启用快速会话清空(drain)，但在现实中，大多数云应用仍然具有某种程度的状态，这种状态保持在一长串请求中，因此会持续很长一段时间。因为应用实例的必要关闭而丢失状态信息会对用户造成干扰并且效率低下。

## 发明内容

[0002] 本公开描述了具有会话切换的云计算应用的协调关闭。

[0003] 在一个实现中，一种计算机实现的方法包括：由关闭协调器向路由器发送指令，以不再向应用实例分配新的会话，其中该应用实例将被关闭；由关闭协调器向应用实例发送请求，以导出用于切换至新的应用实例的关联的会话；由关闭协调器从应用实例接收指示，指示表明应用实例不能处理进一步的请求，因为所有关联的会话已经导出至外部会话存储；由关闭协调器指示路由器，将对应用实例的请求重定向至新的应用实例；由关闭协调器关闭应用实例；以及在限定的超时时段之后，由关闭协调器从外部会话存储中删除所有关联的会话的剩余会话。

[0004] 所描述的主题可以使用计算机实现的方法来实现；非暂时性计算机可读介质存储计算机可读指令以执行计算机实现的方法；以及计算机实现的系统包括与一个或多个计算机可互操作地耦合的一个或多个计算机存储器装置，并具有存储指令的有形的非暂时性的机器可读介质，当由一个或多个计算机执行时，执行计算机实现的方法/存储在非暂时性的计算机可读介质上的计算机可读指令。

[0005] 可以实现本说明书中描述的主题以实现一个或多个以下优点。首先，所提出的解决方案提供了在关闭应用实例之前清空会话的现有技术的替代方案。在现有技术下，长时间运行的会话是有问题的，因为关闭需要及时发生，并且简单地终止在限定的时间目标之后仍在运行的会话，这会导致不期望的中断。其次，在新的方法下，通过在应用实例之间移动粘性会话，可以无缝地(即不让用户注意到)切换长时间运行的进程。第三，新方法有助于使云计算环境看起来始终可用，即使诸如软件维护、生命周期管理和执行扩展操作等任务需要在后台执行。第四，所描述的方法允许告知用户当前运行的会话切换，提供指示用户何时可以恢复工作的时间估计(例如进度条或图形指示器)。在切换进程花费的时间由于会话状态的大小而超过几秒钟的情况下，这是很重要的。

[0006] 在详细描述、权利要求和附图中阐述了本说明书主题的一个或多个实现的细节。根据详细描述、权利要求和附图，本主题的其他特征、方面和优点对于本领域普通技术人员来说将变得显而易见。

## 附图说明

[0007] 图1是示出根据本公开的实现的用于协调的关闭具有会话切换的云计算应用和所涉及的组件的方法的示例的流程图。

[0008] 图2是示出根据本公开的实现的在云平台计算环境中的请求流和所涉及的缓存文件的流程图。

[0009] 图3是示出根据本公开的实现的用于提供与所描述的算法、方法、功能、进程、流程和过程相关联的计算功能的计算机实现的系统的示例的框图。

[0010] 不同附图中相同的附图标记和名称表示相同的元素。

[0011] 具体实现

[0012] 以下详细描述描述了具有会话切换的云计算应用的协调的关闭，并且呈现其以使本领域的任何技术人员能够在一个或多个特定实现的上下文中制作和使用所公开的主题。在不脱离本公开的范围的情况下，可以对所公开的实现进行各种修改、变更和置换，这些修改、变更和置换对于本领域普通技术人员来说是显而易见的，所定义的一般原理可以应用于其他实现和应用。在一些情况下，对于获得所描述主题的理解来说不必要的并且在本领域普通技术人员的技能范围内的一个或多个技术细节可以省略，从而不会模糊一个或多个所描述的实现。本公开不旨在限于所描述或示出的实现，而是符合与所描述的原理和特征一致的最宽范围。

[0013] 在云计算中，应用实例不是单独处理的，而是集体处理的。也就是说，不将单个应用实例视为是特殊的，因为它类似于任何其他应用实例，并且可以随时启动、关闭、删除、重启等。由于各种因素，对云计算环境中的应用进行集体考虑是必要的。

[0014] 首先，云是水平扩展的，也就是说，当有更多的负载时，会启动附加的应用实例，因此负载会分布在更多的实例上。另一方面，这意味着如果需要更多可用容量，也可以关闭应用实例，以减少资源消耗和多余实例的相关的成本。

[0015] 其次，软件更新是通过替换完整的应用实例来应用的，而不是在它们运行时修补进程，这是传统的预置企业资源规划(enterprise resource planning,ERP)系统的模型。这种“淘汰并替换(rip-and-replace)”类型的方法意味着运行新软件版本的应用实例相对于仍然运行旧软件版本的现有应用实例并行启动。一旦旧软件版本实例上不再有用户，就可以将其关闭。理想情况下，这应该尽快发生，以避免由于运行的软件实例数量翻倍而增加资源消耗/成本。除非用户自愿结束他们的会话，否则会话不会在较旧和较新的软件版本软件实例之间切换，因此等待用户完成他们的会话会导致不必要的延迟。

[0016] 第三，在称为可用性区域的隔离故障单元中组织数据中心。通常，每个可用性区域都有独立的硬件、网络、电源甚至冷却系统。如果一个可用性区域变得不稳定，软件可以在该区域的其余可用性区域中继续运行。但这意味着在受影响的可用性区域中运行的应用实例必须关闭，并且正在运行的工作负载必须转移至不同可用性区域中的其他应用实例。

[0017] 当前的软件体系结构是根据这些需求构建的，在很大程度上，当需要时，通过清空活动会话并在所有会话结束后关闭实例，可以在应用实例之间无缝地转移用户工作负载。然而，尽管云应用应该是完全无状态的，这使得能够启用快速会话清空，但在现实中，大多数云应用仍然具有某种程度的状态，这种状态保持在一长串请求中(因此会持续一段时间)。例如，应用可能有安全会话，如果会话丢失，它将要求用户重新登录。其他示例可能包

括草稿的业务状态(诸如正在编辑的业务对象)或交互优化且必将重新启动的计划进程。然而,特别是在关键情况下,诸如当由于硬件问题而需要紧急撤离可用性区域时,关闭之前的宽限期对于实现复杂业务场景的应用来说可能不够长,不足以实现充分的会话清空并避免打扰用户工作。

[0018] 在云计算中,数量可变的应用实例并行处理客户端请求。如前所述,对于某些事件(诸如缩小部署规模(需要较少的实例)、软件维护(例如部署)或可用性区域撤离(例如数据中心基础设施问题)),必须关闭某些应用实例,并将其相关的工作负载重新分布至其他应用实例。如本领域普通技术人员将理解的,除了列出的原因之外,可能还有其他原因需要应用实例的快速关闭。

[0019] 重要的是,关闭和重新分布事件对客户端是透明的,并且不会对活动任务引入明显的中断。由于每个应用实例处理数百或数千个会话,切换它们必须是一个渐进的过程,其中每个单独的会话切换不影响已经切换或仍将切换的其他会话的并行处理。

[0020] 尽管云应用应该是完全无状态的,这使得能够启用快速会话清空,但在现实中,大多数云应用仍然具有某种程度的状态,这种状态保持在一长串请求中,因此会持续一段时间。因为应用实例的必要关闭而丢失状态信息会对用户造成干扰并且效率低下。需要一种解决方案,允许从应用会话中捕获会话状态,并将它们移至接管相应应用会话的工作负载的其他应用实例。

[0021] 在某些情况下,会话状态可能非常大(例如,在复杂的ERP应用中),每次请求都要花费太多时间将会话状态存储至中央位置。在其他情况下,可能会有许多更改会话状态的非常小的请求,使得与实际生产工作相比,预防性会话导出的开支太高。因此,就性能、资源消耗和成本而言,在每个请求之后简单地导出保存在会话状态中的所有数据是太低效和昂贵的,所以需要一种更优化的解决方案,其仅在需要时在那些事件中支持会话切换。

[0022] 分布式状态是必须被保留的,而且对于在关闭时会话切换至的目标应用实例是可访问的。复杂的计划和优化应用跨多个单独的客户端请求来执行庞大的操作,从而建立状态并保存在存储器中,直到到达某个检查点。在微服务架构中,该状态并不是全都保存在面向用户的应用中,而是大部分保存在应用消耗的上游服务中。因此,当应用实例需要关闭时,不仅应用实例自身的状态必须切换至另一个目标应用实例,还需要保留上游服务中的相关状态,并且对于将应用会话移交至的另一个目标应用实例是可访问的。

[0023] 解决方案的实现独立于云平台基础设施也很重要。有相当多的云计算平台技术可以提供基本的应用实例管理和客户端请求路由功能。至少为了效率和可移植性,实现会话切换解决方案的应用开发者应该不需要知道将部署应用的特定云计算环境。为了支持所描述的解决方案,应当实现独立于公共云平台基础设施工作的通用协议(即不需要改变云平台基础设施)。

[0024] 本文描述了一种用于云计算环境的非强力解决方案,以主动切换会话,而不是被动地等待会话结束。在强力解决方案中,所有会话状态都是外部化和分布式的。然而,所描述的解决方案仅在需要时切换会话,避免了当前解决方案的资源和处理开支的浪费。所描述的解决方案通过将单个应用实例的会话切换至其他应用实例,而不是丢失/终止单个应用实例,来协调用户未注意到的单个应用实例的关闭。在复杂的应用场景中(例如,长时间运行的计划和优化应用已经建立了几个小时的会话),这是一项基本能力,可使软件维护、

生命周期管理和性能扩展操作能在后台启用，而不会在任何时候扰乱用户。所描述的方法的核心是中央协调会话切换，但是具有所涉及组件的分布式和自调节责任。

[0025] 图1是示出根据本公开的实现的用于协调的关闭具有会话切换的云计算应用和所涉及的组件的方法100的示例的流程图。

[0026] 图1示出了所涉及的组件，包括客户端102、路由器/负载平衡器104、关闭协调器106、外部会话存储108、应用实例1 110、应用实例2 112和应用实例3 114。在一些实现中，路由器/负载平衡器104可以是超缩放器或定制服务；关闭协调器106可以是定制软件程序；外部会话存储108可以是内存数据库、常规数据库、对象存储或其他数据结构；应用实例1-3(110、112和114)可以是Java、Node.js、C/C++、或Rust。在图1的所述示例中，期望关闭应用实例3 114。

[0027] 为了表述清楚，下面的描述一般是在本描述中的其他附图的背景下描述方法100。然而，将会理解的是，方法100可以例如由任何系统、环境、软件和硬件或者系统、环境、软件和硬件的组合来执行，视情况而定。在一些实现中，方法100的各个步骤可以并行、组合、循环或以任何顺序运行。

[0028] 在1处，关闭协调器106向路由器/负载平衡器104发送请求，指示路由器/负载平衡器不再向要被关闭的应用实例(在该示例中为应用实例3 114)分配新的会话。

[0029] 在一些实现中，路由器/负载平衡器104保持应用实例处理其对应会话(即粘性会话)的数据映射，并将由会话缓存文件标识的正在运行的会话分派给相应的应用实例。新的会话(即没有有效的会话缓存文件)由算法(例如循环法)分配给任何可用的应用实例。

[0030] 在从关闭协调器106接收到请求之后，路由器/负载平衡器104从可用应用实例的列表中移除应用实例3 114，因此在应用实例3 114上不创建新的会话。然而，继续分派正在运行的会话而没有变化(也就是说，仍然将对要被关闭的应用实例的请求分派给该应用实例)，从而在切换进程期间提供不间断的服务。方法100从1前进到2。

[0031] 在2处，关闭协调器106向应用实例3 114发送应该将其关闭的请求，指示应用实例3 114导出其会话以进行切换。同时，应用实例3 114将不会接收对新的会话的请求，因为路由器/负载平衡器104不会向应用实例3 114分配新的会话，但是应用实例3 114仍将接收对其将处理的现有会话的请求，直到对于特定会话的切换已经开始。这允许连续处理，因为导出大量会话将需要一些时间。方法100从2前进到3。

[0032] 在3处，应用实例3 114迭代其所有会话，并将会话导出(例如连续地)至外部会话存储108。尚未开始导出的会话请求仍在处理中。对于其他情况，参考下文。方法100从3前进到4。

[0033] 在4处，假设路由器/负载平衡器104接收到对已经开始(或者甚至完成)切换(导出)的会话的请求(例如，对于会话ID＝X和实例＝3，来自客户端102)，照常处理该请求，因为路由器/负载平衡器104不知道(并且不需要知道)会话切换的进度。路由器/负载平衡器104根据会话缓存文件中会话ID的存在以及对应用实例的映射来操作。方法100从4前进到5。

[0034] 在5处，路由器/负载平衡器104将请求转发给原始应用实例3 114(即会话曾经位于的位置)。方法100从5前进到6。

[0035] 在6处，当应用实例3 114接收到对已经开始(或者甚至完成)切换的会话的请求

时，应用实例3 114不能再处理该请求，因为它可能影响会话状态，这将使(正在运行的或者完成的)会话导出无效。因此，应用实例3 114以至路由器/负载平衡器的原始地址的重定向请求来应答该请求，使会话缓存文件无效，并设置切换缓存文件。应用实例3 114还指定重试后(retry-after)值，该值指示客户端102何时应该在新(＝旧)地址重试请求。

[0036] 重试后值是完成会话状态导出所需的估计持续时间，假设需要在导入能够开始之前完成导出。如果应用实例3 114和外部会话存储108支持流数据，则重试后值可设置为零。此外，如果会话导出已经完成，则重试后值为零。切换缓存文件包括标识原始会话的会话标识符(ID)，并且它现在还用作将会话状态导出至外部会话存储108的密钥。方法100从6前进到7。

[0037] 在7处，当客户端102接收到重定向请求时，客户端102等待由重试后值指定的持续时间，以允许会话状态的导出完成(即如果重试后值大于零)。由于这个等待时间发生在客户端侧，所以它不会经历超时，并且将完全知道会话切换所需的小延迟。对于面对客户端的终端用户，这可以通过进度条(“导出...”)或其他指示等待时间进展/减少的图形指示器来可视化。

[0038] 在等待时间之后(或者如果重试后值等于零，则立即)，客户端102重试该请求，将该请求指向重定向请求中指定的地址，该地址恰好与原始地址相同。该地址指向路由器/负载平衡器104，并且标识应用的任何实例，但是它不是特定于应用实例3 114的，因此该地址在切换期间不会改变。然而，第二请求将不包括会话缓存文件，因为这在重定向请求中是无效的。相反，该请求将携带新设置的切换缓存文件。方法100从7前进到8。

[0039] 在8处，路由器/负载平衡器104接收来自客户端102的第二请求，并检测到没有设置会话缓存文件。通过应用常规的负载平衡算法(例如循环法)，路由器/负载平衡器104生成应该将请求分派到的新的应用实例(这里是应用实例2 112)。因为关闭协调器106最初已经从可用实例列表中移除了将要被关闭的应用实例3 114，因此所确定的应用实例将必然不同于原始应用实例3114。方法100从8前进到9。

[0040] 在9处，应用实例2 112已经接收到该请求，并且检测到不存在会话缓存文件。因此，应用实例2 112生成新的会话，并将会话ID设置为新的会话缓存文件。应用实例2 112还检测切换缓存文件，并在外部会话存储108中检索导出的会话状态的密钥。应用实例2 112将存储的会话状态导入至新创建的会话中。在一些实现中，可以通过流数据来执行导入(如果应用实例2 112和外部会话存储108支持的话)，或者应用实例2 112需要等待，直到从外部会话存储108的导入已经完成。在一些实现中，可以在外部会话存储108中实现锁定机制(例如，一旦已完成导出，通过定义在外部会话存储108中设置的“就绪”类型标志，用于由导出应用实例设置的会话状态密钥)。因为对目标应用实例2 112的请求已经由重试后值延迟，所以通常没有等待时间(或者只有最小的等待时间)，直到可以访问外部会话存储108。

[0041] 可选地，如果所存储的会话状态的导入花费大量时间，则目标应用实例可以配置为向客户端发送另一个重定向请求，具有对导入所存储的会话状态将花费的时间进行估计的重试后值。重定向地址再次与原始地址相同，但是这一次会话缓存文件已经设置为新的会话ID，并且使得切换缓存文件无效(因为已经开始会话状态导入)。以这种方式，客户端102可以再次避免经历超时，并且还可以通过进度条(“正在导入...”)或指示进展/减少的等待时间的其他图形指示器，向面对客户端的终端用户可视化预期延迟。如果存储的会话

状态很小，并且可以快速执行导入，则可以省略第二重定向。方法100从9前进到10。

[0042] 在10处，一旦目标应用实例2 112完成了存储的会话状态导入，应用实例2 112继续照常处理请求。当应用实例2 112响应客户端102时，它包括在切换期间创建的会话缓存文件中的新的会话ID，并使切换缓存文件无效(如果还没有通过如9中可选描述的重定向来完成的话)。方法100从10前进到11。

[0043] 在11处，路由器/负载平衡器104将响应转发给客户端102。在这样做时，路由器/负载平衡器104更新会话ID与应用实例2 112之间的映射，使其成为粘性会话，这将确保再次像以前一样再分派进一步的请求至应用实例2 112(如果这还没有通过如9中可选描述的重定向来完成的话)。方法100从11前进到12。

[0044] 在12处，一旦应用实例3 114已经完成了将所有相关联的会话导出到外部会话存储108，应用实例3 114就向关闭协调器106返回报告，指示因为已经导出了所有会话并且应用实例3 114准备好关闭，所以不能再处理请求。方法100从12前进到13。

[0045] 在13处，因为可能仍然存在与将要被关闭的应用实例3 114相关联的会话(如4至11所述，如果没有触发重定向的最近请求)，关闭协调器106调用路由器/负载平衡器104来完成仍然分配给应用实例3 114的所有剩余会话的重定向。从这一点开始，路由器/负载平衡器104将用重定向来响应映射至应用实例3 114的任何请求，使会话缓存文件无效，并添加从原始会话缓存文件复制会话ID的切换缓存文件。因为已经完成所有会话的导出，重试后值设置为零。换句话说，路由器/负载平衡器组件仅对于已经开始会话切换的会话，逐个案例地接管先前由应用实例3 114执行的过程。当完成导出时，现在这可以由路由器/负载平衡器组件无条件地完成，将应用实例3 114从这一任务中解脱出来。方法100从13前进到14。

[0046] 在14处，当路由器/负载平衡器104处理指向应用实例3 114(要被关闭)的所有进一步请求时，关闭协调器106现在可以安全地触发应用实例3 114的关闭。方法100从14前进到15。

[0047] 在15处，当最大会话超时到期时，关闭协调器106从外部会话存储108中删除所有剩余的会话导出。这些是在开始会话切换后不再从客户端102接收请求的会话。在一些实现中，为了有效地标识这些会话，会话可能已经标记有应用实例的ID，该应用实例在其将其会话导出至外部会话存储108时关闭。在15之后，方法100可以停止。

[0048] 图2是示出根据本公开的实现的在云平台(Cloud Foundry)计算环境中的请求流200和所涉及的缓存文件的流程图。图2示出了所涉及的组件，包括客户端102、路由器104(例如，如图1所示的路由器/负载平衡器104)、应用实例2 112和应用实例3 114。在图2的所述示例中，期望关闭应用实例3 114。

[0049] 为了表述清楚，下面的描述一般是在本描述中的其他附图的背景下描述方法200。然而，将会理解的是，方法200可以例如由任何系统、环境、软件和硬件或者系统、环境、软件和硬件的组合来执行，视情况而定。在一些实现中，方法200的各个步骤可以并行、组合、循环或以任何顺序运行。

[0050] 图2示出了云平台计算环境中的请求流和所涉及的缓存文件的示例。这里，会话ID与应用实例之间的映射实现为客户端侧缓存文件(即VCAP_ID用于标识应用实例)，通过检查会话缓存文件(即JSESSIONID)在路由器104级将其引入。这也解释了为什么路由器104不

能简单地改变会话ID与应用实例之间的映射，因为路由器104不在本地拥有该信息(其存储在客户端侧)。因此，需要重定向机制来更新客户端102处的设置，这只有在由客户端102根据其自己的时间表发送下一个请求时才有可能。

[0051] 在202处，客户端102向路由器104发送对应用实例的请求。方法200从202前进到204。

[0052] 在204处，路由器104接收来自客户端102的请求，并检测到没有设置会话缓存文件。通过应用常规的负载平衡算法(例如循环法)，路由器104生成应将请求分派至的新的应用实例(这里是应用实例3 114)，并将其已接收的请求转发给该应用实例。方法200从204前进到206。

[0053] 在206处，应用实例3 114接收来自路由器104的请求，并检测到不存在会话缓存文件。因此，应用实例3 114生成新的会话，并将会话ID X设置为新的会话缓存文件(JSESSIONID＝X)，并将缓存文件连同应用响应一起返回给路由器104。方法200从206前进到208。

[0054] 在208处，路由器104接收来自应用实例3 114的响应，并检测到存在会话缓存文件，指示该应用已经生成了会话。因此，路由器104将应用实例3设置为新的VCAP_ID缓存文件(VCAP_ID＝3)，建立粘性会话。它将会话缓存文件和VCAP_ID缓存文件连同应用响应一起返回给客户端102。方法200从208前进到209。

[0055] 在209处，作为已经从关闭协调器106接收了导出其所有会话(未示出)的关闭请求的结果，应用实例3 114将会话X导出至外部会话存储(例如，如图1中描述的外部会话存储108)。方法200从209前进到210。

[0056] 在210处，客户端102向路由器104发送对应用实例的第二请求。方法200从210前进到212。

[0057] 在212处，路由器104接收来自客户端102的请求，并检测到设置了VCAP_ID缓存文件。客户端102将其已接收的请求转发给相应的应用实例3114。方法200从212前进到214。

[0058] 在214处，应用实例3 114接收来自路由器104的请求，并检测到其已导出了在会话缓存文件(JSESSIONID＝X)中包括的会话X。因此，应用实例3 114生成切换缓存文件，并将会话ID X设置为新的切换缓存文件(HANDOVERID＝X)。应用实例3 114还通过使会话缓存文件到期(JSESSIONID＝<expired>)来使其无效，并将缓存文件连同重定向请求一起作为响应返回给路由器104。方法200从214前进到216。

[0059] 在216处，路由器104接收来自应用实例3 114的响应，并检测到会话缓存文件已经无效。因此，路由器104通过使VCAP_ID缓存文件到期(VCAP_ID＝<expired>)来使粘性会话无效。路由器104将到期的会话和VCAP ID缓存文件以及切换缓存文件连同应用响应一起返回给客户端102。方法200从216前进到218。

[0060] 在218处，客户端102跟随作为对其第二请求的响应而接收的重定向请求，并向路由器104发送对应用实例的第三请求。因为会话和VCAP_ID缓存文件两者都已到期，该请求仅包括切换缓存文件(HANDOVERID＝X)。方法200从218前进到220。

[0061] 在220处，路由器104接收来自客户端102的请求，并检测到没有设置会话缓存文件。通过应用常规的负载平衡算法(例如循环法)，路由器104生成应将请求分派至的新的应用实例(这里是应用实例2 112)，并将它已经接收到的请求转发到该应用实例。方法200从

220前进到222。

[0062] 在222处，应用实例2 112接收来自路由器104的请求，并检测到不存在会话缓存文件。因此，应用实例2 112生成新的会话，并将会话ID Y设置为新的会话缓存文件(JSESSIONID＝Y)。应用实例2 112还检测到存在切换缓存文件(HANDOVERID＝X)。因此，应用2 112将会话X导入会话Y。应用实例2 112还通过使切换缓存文件到期(HANDOVERID＝<expired>)来使其无效，并将到期的缓存文件连同新的会话缓存文件和应用响应一起返回给路由器104。方法200从222前进到224。

[0063] 在224处，路由器104接收来自应用实例2 112的响应，并检测到存在新的会话缓存文件，指示该应用已经生成了新的会话。因此，路由器104将应用实例2设置为新的VCAP_ID缓存文件(VCAP_ID＝2)，这建立了新的粘性会话。路由器104将会话缓存文件和VCAP_ID缓存文件连同应用响应一起返回给客户端102。在220之后，方法200可以停止。

[0064] 图3是示出根据本公开的实现的用于提供与所描述的算法、方法、功能、进程、流程和过程相关联的计算功能的计算机实现的系统300的示例的框图。在所示的实现中，系统300包括计算机302和网络330。

[0065] 所示计算机302旨在涵盖任何计算装置，诸如服务器、台式计算机、膝上型/笔记本计算机、无线数据端口、智能电话、个人数据助理(PDA)、平板计算机、这些装置中的一个或多个处理器或计算装置的组合，包括计算装置的物理或虚拟实例、或计算装置的物理或虚拟实例的组合。另外，计算机302可以包括输入装置，诸如小键盘、键盘或触摸屏，或者可以接受用户信息的输入装置的组合，以及在图形类型用户界面(UI)(或GUI)或其他UI上传达与计算机302的操作相关联的信息的输出装置，包括数字数据、视觉、音频、另一种类型的信息，或者多种类型的信息的组合。

[0066] 计算机302可以充当分布式计算系统中的角色，例如作为客户端、网络组件、服务器或数据库或者另一种持久性角色，或用于执行本公开所述主题的角色组合。所示的计算机302与网络330可通信地耦合。在一些实现中，计算机302的一个或多个组件可以被配置为在包括云计算、本地或全球的环境或环境组合内操作。

[0067] 在高层次上，计算机302是可操作用于接收、发送、处理、存储或管理与所述主题相关联的数据和信息的电子计算装置。根据一些实现，计算机302还可以包括或可通信地耦合至服务器，诸如应用服务器、电子邮件服务器、网页服务器、缓存服务器或流数据服务器，或者服务器的组合。

[0068] 计算机302可以通过网络330接收请求(例如，从在另一个计算机302上执行的客户端软件应用)，并通过使用软件应用或软件应用的组合处理所接收的请求来响应所接收的请求。此外，请求也可以从内部用户(例如，从命令控制台或通过另一内部访问方法)、外部或第三方、或其他实体、个人、系统或计算机发送到计算机302。

[0069] 计算机302的每个组件可以使用系统总线303进行通信。在一些实现中，计算机302的任何或所有组件，包括硬件、软件、或硬件和软件的组合，可以使用应用编程接口(API)312、服务层313或API 312和服务层313的组合经由系统总线303进行对接。API 312可以包括用于例程、数据结构和对象类的规范。API 312可以是独立于或依赖于计算机语言的，并且指的是完整接口、单个功能或者甚至是API的集合。服务层313向计算机302或可通信地耦合至计算机302的其他组件(无论是否示出)提供软件服务。使用服务层313的所有服务消费

者都可以访问计算机302的功能。诸如由服务层313提供的那些软件服务通过限定的接口提供可重用的、限定的功能。例如，接口可以是以计算语言(例如JAVA或C++)或计算语言的组合编写的软件，并以特定格式(例如可扩展标记语言(XML))或格式的组合提供数据。尽管被示为计算机302的集成组件，但可选实现可以将API 312或服务层313示为相对于计算机302的其他组件或可通信地耦合至计算机302的其他组件(无论是否示出)的独立组件。此外，API 312或服务层313的任何或所有部分可以实现为另一软件模块、企业应用或硬件模块的子模块或次模块，而不脱离本公开的范围。

[0070] 计算机302包括接口304。尽管被示为单个接口304，但是根据计算机302的特定需求、期望或特定实现，可以使用两个或更多个接口304。计算机302使用接口304来与分布式环境中通信地链接至网络330的另一计算系统(无论是否示出)进行通信。一般地，接口304可操作来与网络330通信，并且包括在软件、硬件或软件和硬件的组合中编码的逻辑。更具体地，接口304可以包括支持与通信相关联的一个或多个通信协议的软件，使得网络330或接口304的硬件可操作来在所示计算机302以内和之外传达物理信号。

[0071] 计算机302包括处理器305。尽管被示为单个处理器305，但是根据计算机302的特定需求、期望或特定实现，可以使用两个或更多个处理器305。一般地，处理器305执行指令并操纵数据以执行计算机302的操作以及本公开中描述的任何算法、方法、功能、进程、流程和过程。

[0072] 计算机302还包括数据库306，其可以为计算机302、通信地链接到网络330的另一组件(无论是否示出)或计算机302和另一组件的组合来保存数据。例如，数据库306可以是存储与本公开一致的数据的存储器内的或常规的数据库。在一些实现中，根据计算机302和所描述的功能的特定需求、期望或特定实现，数据库306可以是两种或更多种不同数据库类型的组合(例如混合存储器内的和常规的数据库)。尽管被示为单个数据库306，但是根据计算机302和所描述的功能的特定需求、期望或特定实现，可以使用相似或不同类型的两个或多个数据库。虽然数据库306被示为计算机302的集成组件，但是在可选实现中，数据库306可以在计算机302的外部。数据库306可以保存、处理和提供与所述方法一致的任何数据类型。

[0073] 计算机302还包括存储器307，其可以为计算机302、通信地链接至网络330的另一个或多个组件(无论是否示出)或计算机302和另一组件的组合来保存数据。存储器307可以存储与本公开一致的任何数据。在一些实现中，根据计算机302和所描述的功能的特定需求、期望或特定实现，存储器307可以是两种或更多种不同类型的存储器的组合(例如半导体和磁存储的组合)。尽管被示为单个存储器307，但是根据计算机302和所描述的功能的特定需求、期望或特定实现，可以使用两个或更多个存储器307或者类似或不同类型的存储器。虽然存储器307被示为计算机302的集成组件，但是在可选实现中，存储器307可以在计算机302的外部。

[0074] 应用308是算法软件引擎，根据计算机302的特定需求、期望或特定实现提供功能，特别是关于本公开中描述的功能。例如，应用308可以充当一个或多个组件、模块或应用。此外，尽管被示为单个应用308，但是应用308可以实现为计算机302上的多个应用308。此外，尽管被示为集成至计算机302，但是在可选实现中，应用308可以在计算机302的外部。

[0075] 计算机302还可以包括电源314。电源314可以包括可充电或不可充电的电池，其可

以被配置为用户可替换的或非用户可替换的。在一些实现中，电源314可以包括功率转换或管理电路(包括充电、待机或另一功率管理功能)。在一些实现中，电源314可以包括电源插头，以允许将计算机302插入墙壁插座或另一电源中，以例如给计算机302供电或给可充电电池充电。

[0076] 可以有任意数量的计算机302与包括计算机302的计算机系统相关联或在其外部，每个计算机302经由网络330进行通信。此外，在不脱离本公开的范围的情况下，术语“客户端”、“用户”或其他适当的术语可以酌情互换使用。此外，本公开设想许多个用户可以使用一个计算机302，或者一个用户可以使用多个计算机302。

[0077] 所述主题的实现可以包括一个或多个单独或组合的特征。

[0078] 例如，在第一实现中，一种计算机实现的方法，包括：由关闭协调器向路由器发送指令，以不再向应用实例分配新的会话，其中该应用实例将被关闭；由关闭协调器向应用实例发送请求，以导出用于切换至新的应用实例的关联的会话；由关闭协调器从应用实例接收指示，指示表明应用实例不能处理进一步的请求，因为所有关联的会话已经导出至外部会话存储；由关闭协调器指示路由器，将对应用实例的请求重定向至新的应用实例；由关闭协调器关闭应用实例；以及在限定的超时时段之后，由关闭协调器从外部会话存储中删除所有关联的会话的剩余会话。

[0079] 前述和其他所描述的实现均可以可选地包括一个或多个以下特征：

[0080] 第一特征，可与任何后述特征组合，包括由路由器从可用应用实例的列表中移除应用实例，以防止在应用实例上创建新的会话。

[0081] 第二特征，可与任何前述或后述特征组合，包括：由应用实例迭代所有会话以创建可导出会话；以及将可导出会话导出至外部会话存储。

[0082] 第三特征，可与任何前述或后述特征组合，包括：由路由器从客户端接收具有会话标识符(ID)和应用实例标识的请求；以及从路由器向应用实例转发该请求；由应用实例向路由器响应重定向请求，包括路由器的原始地址、无效的会话缓存文件、包括会话ID的切换缓存文件、以及指示客户端何时应该重试请求的重试后值；以及从路由器向客户端发送重定向请求。

[0083] 第四特征，可与任何前述或后述特征组合，包括：由客户端等待由重试后值规定的持续时间，以允许完成从应用实例导出会话状态至外部会话存储；由客户端向路由器重试该请求作为新的请求，其中该请求不包括会话缓存文件，而包括具有切换缓存文件值的新的切换缓存文件；由路由器从客户端接收新的请求；由路由器检测新的请求不包括会话缓存文件；以及由路由器生成新的应用实例，利用该应用实例分派新的请求。

[0084] 第五特征，可与任何前述或后述特征组合，包括：由新的应用实例从路由器接收新的请求；由新的应用实例检测不存在会话缓存文件；由新的应用实例生成新的会话；由新的应用实例用标识新的会话的新的会话ID来设置新的会话缓存文件；由新的应用实例检测新的切换缓存文件的存在；由新的应用实例使用新的切换缓存文件来检索被包括在新的切换缓存文件中的会话ID；以及将存储的与应用实例相关联的会话状态导入至新的会话。

[0085] 第六特征，可与任何前述或后述特征组合，包括：由新的应用实例处理新的请求；由新的应用实例向路由器响应新的响应，该新的响应包括新的会话缓存文件中的新的会话ID和具有无效状态的新的切换缓存文件；由路由器向客户端转发新的响应；以及由路由器

作为粘性会话更新在新的会话ID与新的应用实例之间的映射。

[0086] 在第二实现中，一种存储可由计算机系统执行的一个或多个指令的非暂时性计算机可读介质，其执行的操作包括：由关闭协调器向路由器发送指令，以不再向应用实例分配新的会话，其中该应用实例将被关闭；由关闭协调器向应用实例发送请求，以导出用于切换至新的应用实例的关联的会话；由关闭协调器从应用实例接收指示，指示表明应用实例不能处理进一步的请求，因为所有相关联的会话已经导出至外部会话存储；由关闭协调器向路由器指示，将对应用实例的请求重定向至新的应用实例；由关闭协调器关闭应用实例；以及在限定的超时时段之后，由关闭协调器从外部会话存储中删除所有关联的会话的剩余会话。

[0087] 前述和其他所描述的实现均可以可选地包括一个或多个以下特征：

[0088] 第一特征，可与任何后述特征组合，包括可由计算机系统执行的一个或多个指令，以执行操作用于由路由器从可用应用实例列表中移除应用实例，以防止在应用实例上创建新的会话。

[0089] 第二特征，可与任何前述或后述特征组合，包括可由计算机系统执行的一个或多个指令，以执行操作用于：由应用实例迭代所有会话以创建可导出会话；以及将可导出会话导出至外部会话存储。

[0090] 第三特征，可与任何前述或后述特征组合，包括可由计算机系统执行的一个或多个指令，以执行操作用于：由路由器从客户端接收具有会话标识符(ID)和应用实例标识的请求；以及从路由器向应用实例转发该请求；由应用实例向路由器响应重定向请求，该重定向请求包括路由器的原始地址、无效的会话缓存文件、包括会话ID的切换缓存文件以及指示客户端何时应该重试请求的重试后值；以及从路由器向客户端发送重定向请求。

[0091] 第四特征，可与任何前述或后述特征组合，包括可由计算机系统执行的一个或多个指令，以执行操作用于：由客户端等待由重试后值规定的持续时间，以允许完成从应用实例导出会话状态至外部会话存储；由客户端向路由器重试该请求作为新的请求，其中该请求不包括会话缓存文件，而包括具有切换缓存文件值的新的切换缓存文件；由路由器从客户端接收新的请求；由路由器检测新的请求不包括会话缓存文件；以及由路由器生成新的应用实例，利用该新的应用实例分派新的请求。

[0092] 第五特征，可与任何前述或后述特征组合，包括可由计算机系统执行的一个或多个指令，以执行操作用于：由新的应用实例从路由器接收新的请求；由新的应用实例检测不存在会话缓存文件；由新的应用实例生成新的会话；由新的应用实例用标识新的会话的新的会话ID来设置新的会话缓存文件；由新的应用实例检测新的切换缓存文件的存在；由新的应用实例使用新的切换缓存文件来检索被包括在新的切换缓存文件中的会话ID；以及将存储的与应用实例相关联的会话状态导入至新的会话。

[0093] 第六特征，可与任何前述或后述特征组合，包括可由计算机系统执行的一个或多个指令，以执行操作用于：由新的应用实例处理新的请求；由新的应用实例向路由器响应新的响应，该新的响应包括新的会话缓存文件中的新的会话ID和具有无效状态的新的切换缓存文件；由路由器向客户端转发新的响应；以及由路由器作为粘性会话更新在新的会话ID与新的应用实例之间的映射。

[0094] 在第三实现中，一种计算机实现的系统，包括：一个或多个计算机；以及与一个或

多个计算机可互操作地耦合的一个或多个计算机存储器装置，并且具有存储一个或多个指令的有形的非暂时性的机器可读介质，当由一个或多个计算机执行指令时，指令执行一个或多个操作，包括：由关闭协调器向路由器发送指令，以不再向应用实例分配新的会话，其中该应用实例将被关闭；由关闭协调器向应用实例发送请求，以导出用于切换至新的应用实例的关联的会话；由关闭协调器从应用实例接收指示，指示表明应用实例不能处理进一步的请求，因为所有相关联的会话已经导出至外部会话存储；由关闭协调器向路由器指示，将对应用实例的请求重定向至新的应用实例；由关闭协调器关闭应用实例；以及在限定的超时时段之后，由关闭协调器从外部会话存储中删除所有关联的会话的剩余会话。

[0095] 前述和其他所描述的实现均可以可选地包括一个或多个以下特征：

[0096] 第一特征，可与任何后述特征组合，包括一个或多个指令，当由一个或多个计算机执行时，执行一个或多个操作用于由路由器从可用应用实例列表中移除应用实例以防止在应用实例上创建新的会话。

[0097] 第二特征，可与任何前述或后述特征组合，包括一个或多个指令，当由一个或多个计算机执行时，执行一个或多个操作用于：由应用实例迭代所有会话以创建可导出会话；以及将可导出会话导出至外部会话存储。

[0098] 第三特征，可与任何前述或后述特征组合，包括一个或多个指令，当由一个或多个计算机执行时，执行一个或多个操作用于：由路由器从客户端接收具有会话标识符(ID)和应用实例标识的请求；以及从路由器向应用实例转发该请求；由应用实例向路由器响应重定向请求，该重定向请求包括路由器的原始地址、无效的会话缓存文件、包括会话ID的切换缓存文件以及指示客户端何时应该重试请求的重试后值；以及从路由器向客户端发送重定向请求。

[0099] 第四特征，可与任何前述或后述特征组合，包括一个或多个指令，当由一个或多个计算机执行时，执行一个或多个操作用于：由客户端等待由重试后值规定的持续时间，以允许完成从应用实例导出会话状态至外部会话存储；由客户端向路由器重试该请求作为新的请求，其中该请求不包括会话缓存文件，而包括具有切换缓存文件值的新的切换缓存文件；由路由器从客户端接收新的请求；由路由器检测新的请求不包括会话缓存文件；以及由路由器生成新的应用实例，利用该应用实例分派新的请求。

[0100] 第五特征，可与任何前述或后述特征组合，包括一个或多个指令，当由一个或多个计算机执行时，执行一个或多个操作用于：由新的应用实例从路由器接收新的请求；由新的应用实例检测不存在会话缓存文件；由新的应用实例生成新的会话；由新的应用实例用标识新的会话的新的会话ID来设置新的会话缓存文件；由新的应用实例检测新的切换缓存文件的存在；由新的应用实例使用新的切换缓存文件来检索被包括在新的切换缓存文件中的会话ID；以及将存储的与应用实例相关联的会话状态导入至新的会话。

[0101] 第六特征，可与任何前述或后述特征组合，包括一个或多个指令，当由一个或多个计算机执行时，执行一个或多个操作用于：由新的应用实例处理新的请求；由新的应用实例向路由器响应新的响应，该新的响应包括新的会话缓存文件中的新的会话ID和具有无效状态的新的切换缓存文件；由路由器向客户端转发新的响应；以及由路由器作为粘性会话更新在新的会话ID与新的应用实例之间的映射。

[0102] 本说明书中描述的主题和功能操作的实现可以在数字电子电路、有形体现的计算

机软件或固件、计算机硬件中实现，包括本说明书中公开的结构及其结构等同，或者它们中的一个或多个的组合。所述主题的软件实现可以实现为一个或多个计算机程序，即在有形的非暂时性计算机可读介质上编码的计算机程序指令的一个或多个模块，用于由计算机或计算机实现的系统执行或控制其操作。可选地或附加地，程序指令可以在人工生成的传播信号中/上编码，例如机器生成的电、光或电磁信号，生成该信号以编码用于向接收器设备传输的信息，以便由计算机或计算机实现的系统执行。计算机存储介质可以是机器可读存储装置、机器可读存储基质、随机或串行存取存储装置或计算机存储介质的组合。配置一个或多个计算机意味着一个或多个计算机已经安装了硬件、固件或软件(或硬件、固件和软件的组合)，使得当软件由一个或多个计算机执行时，执行特定的计算操作。然而，计算机存储介质不是传播的信号。

[0103] 术语“实时”、“实时的”、“实时地”、“实(快)时(RFT，real(fast)time)”、“近实时(NRT，near(ly)real-time)”、“准实时”或类似术语(如本领域普通技术人员所理解的那样)，是指动作和响应在时间上接近，使得个体觉察动作和响应基本上同时发生。例如，在个体访问数据的动作之后，响应显示(或启动显示)数据的时间差可以小于1毫秒(ms)、小于1秒(s)或小于5秒。虽然不需要立即显示(或启动显示)所请求的数据，但是考虑到所描述的计算系统的处理限制以及例如收集、精确测量、分析、处理、存储或传输数据所需的时间，其显示(或启动显示)没有任何故意的延迟。

[0104] 术语“数据处理设备”、“计算机”、“计算装置”或“电子计算机装置”(或本领域普通技术人员理解的等效术语)指的是数据处理硬件，并涵盖用于处理数据的各种设备、装置和机器，包括例如可编程处理器、计算机或多个处理器或计算机。计算机还可以是或进一步包括专用逻辑电路，例如中央处理单元(CPU，central processing unit)、现场可编程门阵列(FPGA，field-programmable gate array)或专用集成电路(ASIC，application-specific integrated circuit)。在一些实现中，计算机或计算机实现的系统或专用逻辑电路(或者计算机或计算机实现的系统和专用逻辑电路的组合)可以是基于硬件或软件的(或者基于硬件和软件两者的组合)。计算机可以可选地包括为计算机程序创建执行环境的代码，例如构成处理器固件、协议栈、数据库管理系统、操作系统或执行环境的组合的代码。本公开考虑使用具有操作系统的计算机或计算机实现的系统，例如LINUX、UNIX、WINDOWS、MAC OS、ANDROID或IOS，或者操作系统的组合。

[0105] 计算机程序，也可以被称为或描述为程序、软件、软件应用、单元、模块、软件模块、脚本、代码或其他组件，可以以任何形式的编程语言编写，包括编译或解释语言，或者声明性或过程性语言，并且可以以任何形式部署，包括例如作为计算环境中使用的独立程序、模块、组件或子例程。计算机程序可以但不需要对应于文件系统中的文件。程序可以存储在保存其他程序或数据的文件的一部分中，例如一个或多个脚本存储在标记语言文档中、专用于所讨论程序的单个文件中或多个协同文件中，例如存储一个或多个模块、子程序或代码部分的文件。计算机程序可以部署为在一个计算机上或者在位于一个地点或分布在多个地点并通过通信网络互连的多个计算机上执行。

[0106] 虽然各图中所示的程序部分可以示为使用各种对象、方法或其他进程实现所述特征和功能的单独组件，诸如单元或模块，但程序也可以酌情包括多个子单元、子模块、第三方服务、组件、库和其他组件。相反，各种组件的特征和功能可以酌情组合成单个组件。用于

进行计算确定的阈值可以静态地、动态地或者静态和动态兼有地确定。

[0107] 所描述的方法、进程或逻辑流程代表与本公开一致的功能的一个或多个示例，并且不旨在将本公开限于所描述或图示的实现，而是符合与所描述的原理和特征一致的最宽范围。所描述的方法、进程或逻辑流程可以由执行一个或多个计算机程序的一个或多个可编程计算机来执行，以通过对输入数据进行操作并生成输出数据来执行功能。方法、进程或逻辑流程也可以由专用逻辑电路来执行，并且计算机也可以实现为专用逻辑电路，例如CPU、FPGA或ASIC。

[0108] 用于执行计算机程序的计算机可以基于通用或专用微处理器、两者兼有或者另一种类型的CPU。一般地，CPU将从存储器接收指令和数据，并向存储器写入指令和数据。计算机的基本元件是用于执行指令的CPU，以及用于存储指令和数据的一个或多个存储装置。一般地，计算机还将包括或可操作地耦合至一个或多个用于存储数据的大容量存储装置，例如磁盘、磁光盘或光盘，从其接收数据或向其传输数据，或两者兼有。然而，计算机不需要有这样的装置。此外，计算机可以嵌入在另一个装置中，例如移动电话、个人数字助理(PDA，personal digital assistant)、移动音频或视频播放器、游戏控制台、全球定位系统(GPS，global positioning system)接收器或便携式存储器存储装置，例如通用串行总线(USB，universal serial bus)闪存驱动器，仅举几例。

[0109] 用于存储计算机程序指令和数据的非暂时性计算机可读介质可以包括所有形式的永久/非永久或易失/非易失存储器、介质和存储器装置，包括例如半导体存储器装置，例如随机存取存储器(RAM，random access memory)、只读存储器(ROM，read-only memory)、相变存储器(PRAM，phase change memory)、静态随机存取存储器(SRAM，static random access memory)、动态随机存取存储器(DRAM，dynamic random access memory)、可擦除可编程只读存储器(EPROM，erasable programmable read-only memory)、电可擦除可编程只读存储器(EEPROM，electrically erasable programmable read-only memory)和闪存装置；磁性装置，例如磁带、盒带、卡带、内部/可移动磁盘；磁光盘；以及光学存储装置，例如数字多功能/视频光盘(DVD，digital versatile/video disc)、压缩光盘(CD，compact disc)-ROM、DVD+/-R、DVD-RAM、DVD-ROM、高分辨率/密度(HD，high-definition/density)-DVD和蓝光/蓝光光盘(BD，BLU-RAY DISC))以及其他光学存储技术。存储器可以存储各种对象或数据，包括高速缓存、类、框架、应用、模块、备份数据、作业、网页、网页模板、数据结构、数据库表、存储动态信息的储存库或者包括任何参数、变量、算法、指令、规则、约束条件或引用参考的其他适当信息。此外，存储器可以包括其他适当的数据，诸如日志、策略、安全或访问数据或者报告文件。处理器和存储器可以由专用逻辑电路来补充或并入其中。

[0110] 为了提供与用户的交互，本说明书中所述主题的实现可以在计算机上实现，该计算机具有用于向用户显示信息的显示装置，例如阴极射线管(CRT，cathode ray tube)、液晶显示器(LCD，liquid crystal display)、发光二极管(LED，light emitting diode)或等离子监视器，以及用户可以通过其向计算机提供输入的键盘和指示装置，例如鼠标、轨迹球或轨迹板。也可以使用触摸屏向计算机提供输入，诸如具有压力敏感性的平板计算机表面或使用电容或电传感的多点触摸屏。可以使用其他类型的装置来与用户交互。例如，提供给用户的反馈可以是任何形式的感觉反馈(诸如视觉、听觉、触觉或反馈类型的组合)。来自用户的输入可以以任何形式接收，包括声音、语言或触觉输入。此外，计算机可以通过向用户

使用的客户端计算装置发送文档和从用户使用的客户端计算设备接收文档来与用户交互(例如通过响应于从网页浏览器接收的请求向用户的移动计算装置上的网页浏览器发送网页)。

[0111] 术语“图形用户界面(GUI,graphical user interface)”可以用单数或复数来描述一个或多个图形用户界面以及特定图形用户界面的每个显示。因此,GUI可以表示任何图形用户界面,包括但不限于网页浏览器、触摸屏或命令行界面(CLI,command line interface),其处理信息并有效地向用户呈现信息结果。一般而言,GUI可以包括许多用户界面(UI,user interface)元素,其中一些或全部与网页浏览器相关联,诸如交互式字段、下拉列表和按钮。这些和其他UI元素可以与网页浏览器的功能有关或代表网页浏览器的功能。

[0112] 本说明书中所述主题的实现可以在计算系统中实现,该计算系统包括后端组件(例如作为数据服务器),或包括中间件组件(例如应用服务器),或包括前端组件(例如具有图形用户界面或网页浏览器的客户端计算机,用户可通过其与本说明书中所述主题的实现进行交互),或一个或多个此类后端、中间件或前端组件的任何组合。系统的组件可以通过任何形式或介质的有线或无线数字数据通信(或数据通信的组合)互连,例如通信网络。通信网络的示例包括局域网(LAN,local area network)、无线电接入网(RAN,radio access network)、城域网(MAN,metropolitan area network)、广域网(WAN,wide area network)、全球微波互联接入(WIMAX,Worldwide Interoperability for Microwave Access)、使用例如802.11x或其他协议的无线局域网(WLAN,wireless local area network)、互联网的全部或一部分、另一通信网络或通信网络的组合。通信网络可以与例如互联网协议(IP,Intemet Protocol)分组、帧中继帧、异步传输模式(ATM,Asynchronous Transfer Mode)信元、语音、视频、数据或网络节点之间的其他信息进行通信。

[0113] 计算系统可以包括客户端和服务器。客户端和服务器一般彼此远离,并且通常通过通信网络进行交互。客户端和服务器的关系是凭借在各自的计算机上运行的计算机程序而产生的,并且彼此具有客户端-服务器关系。

[0114] 虽然本说明书包含许多具体实现的细节,但这些细节不应解释为对任何发明概念的范围或可要求保护的范围的限制,而是作为可以针对特定发明概念的特定实现的特征的描述。本说明书中在单独实现的上下文中描述的某些特征也可以在单个实现中组合实现。相反,在单个实现的上下文中描述的各种特征也可以在多个实现中、单独地或以任何子组合来实现。此外,尽管先前描述的特征可以描述为在某些组合中起作用,并且甚至最初如此要求保护,但是在一些情况下,来自所要求保护的组合的一个或多个特征可以从该组合中删除,并且所要求保护的组合可以指向子组合或子组合的变型。

[0115] 对主题的特定实现已经进行了描述。对本领域技术人员显而易见的是,所描述的实现的其他实现、变更和置换在所附权利要求的范围内。虽然在附图或权利要求中以特定的顺序描述了操作,但是这不应该理解为要求以所示的特定顺序或连续依次执行这些操作,或者要求执行所有示出的操作(可以认为一些操作是可选的),以达至期望的结果。在某些情况下,多任务处理或并行处理(或多任务处理和并行处理的组合)可能是有利的,并且认为是适宜执行的。

[0116] 不应将前述实现中各种系统模块和组件的分离或集成理解为要求在所有实现中

进行这种分离或集成，并且应理解，所述程序组件和系统一般可一起集成在单个软件产品中或打包成多个软件产品。

[0117] 因此，之前描述的示例性实现并不界定或约束本发明。在不脱离本公开的范围的情况下，其他改动、替换和变更也是可能的。

[0118] 此外，任何要求保护的实现认为是至少适用于计算机实现的方法；存储计算机可读指令以执行计算机实现的方法的非暂时性计算机可读介质；以及计算机系统，包括与硬件处理器可互操作地耦合的计算机存储器，该硬件处理器配置为执行计算机实现的方法或存储在非暂时性计算机可读介质上的指令。

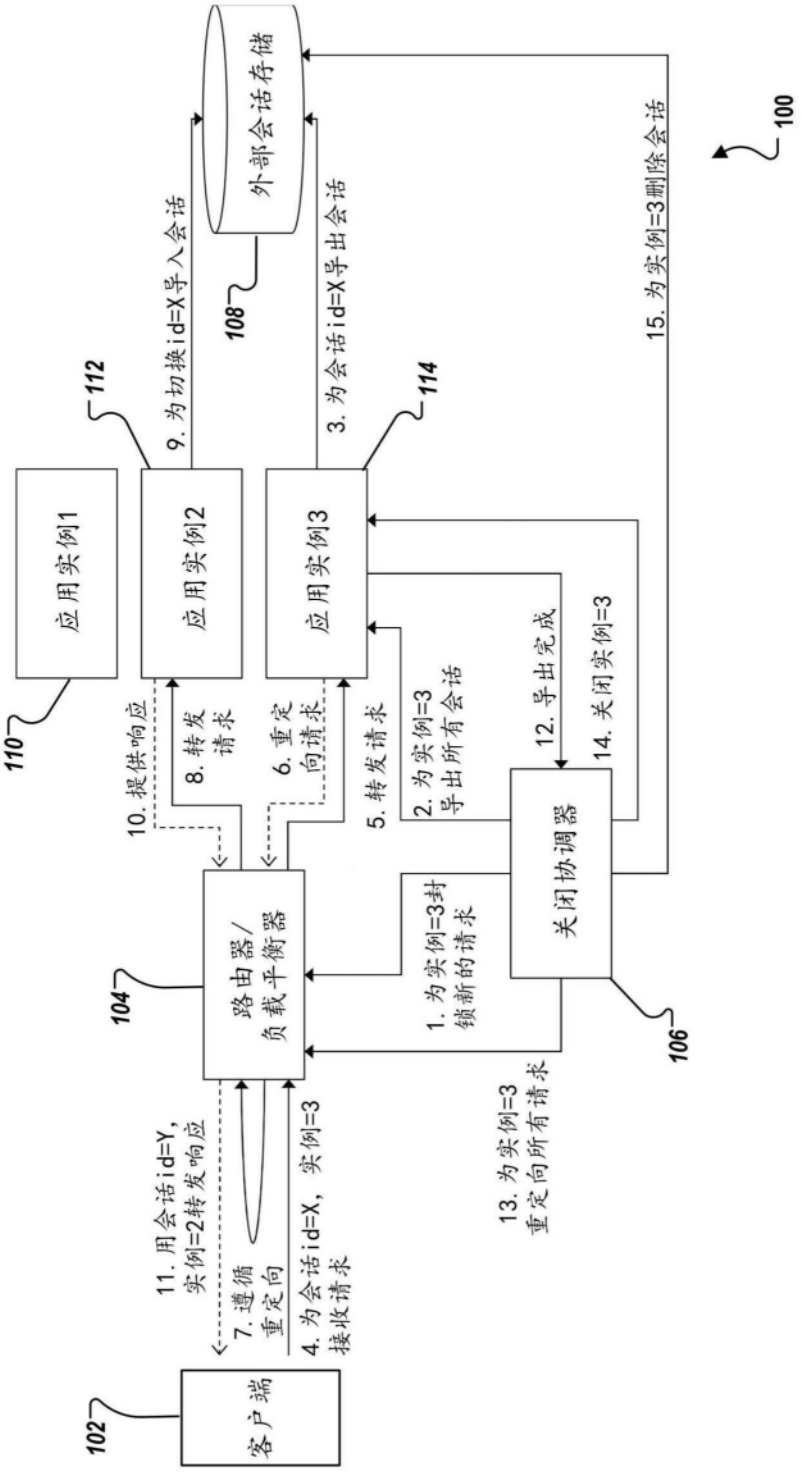

外部会话存储
108
100
15. 为实例=3删除会话
9. 为切换id=X导入会话
3. 为会话id=X导出会话
112
114
应用实例1
应用实例2
应用实例3
110
10. 提供响应
8. 转发请求
6. 重定向请求
5. 转发请求
2. 为实例=3导出所有会话
12. 导出完成
14. 关闭实例=3
路由器/负载平衡器
关闭协调器
104
106
1. 为实例=3封锁新的请求
11. 用会话id=Y，实例=2转发响应
7. 遵循重定向
4. 为会话id=X，实例=3接收请求
13. 为实例=3重定向所有请求
客户端
102

图1

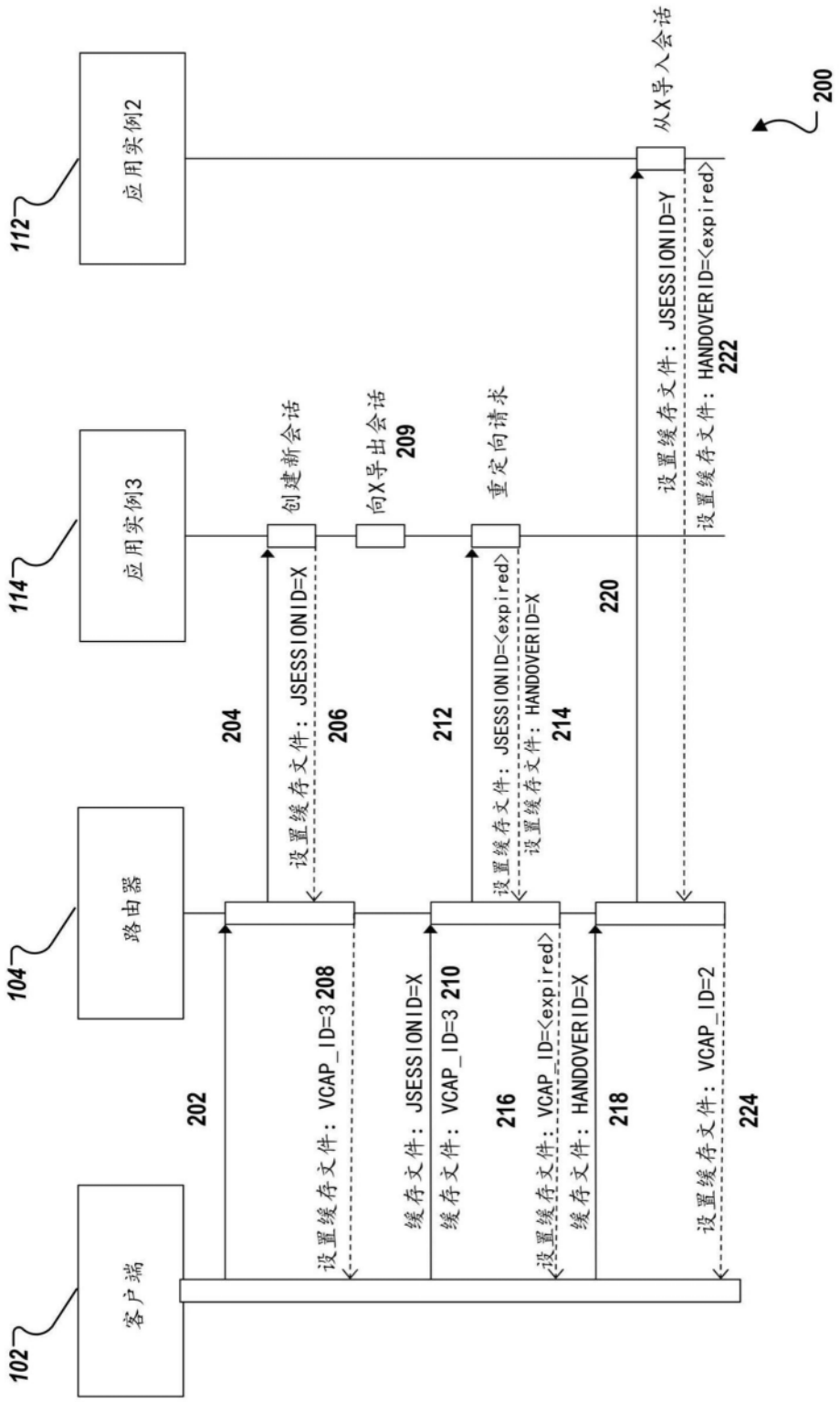

客户端
102
路由器
104
应用实例3
114
应用实例2
112
200
从X导入会话
202
204
设置缓存文件：VCAP_ID=3 208
创建新会话
设置缓存文件：JSESSIONID=X
206
缓存文件：JSESSIONID=X
缓存文件：VCAP_ID=3 210
向X导出会话
209
212
设置缓存文件：VCAP_ID=<expired>
216
设置缓存文件：JSESSIONID=<expired>
重定向请求
214
缓存文件：HANDOVERID=X
设置缓存文件：HANDOVERID=X
218
220
设置缓存文件：JSESSIONID=Y
设置缓存文件：HANDOVERID=<expired>
222
设置缓存文件：VCAP_ID=2
224

图2

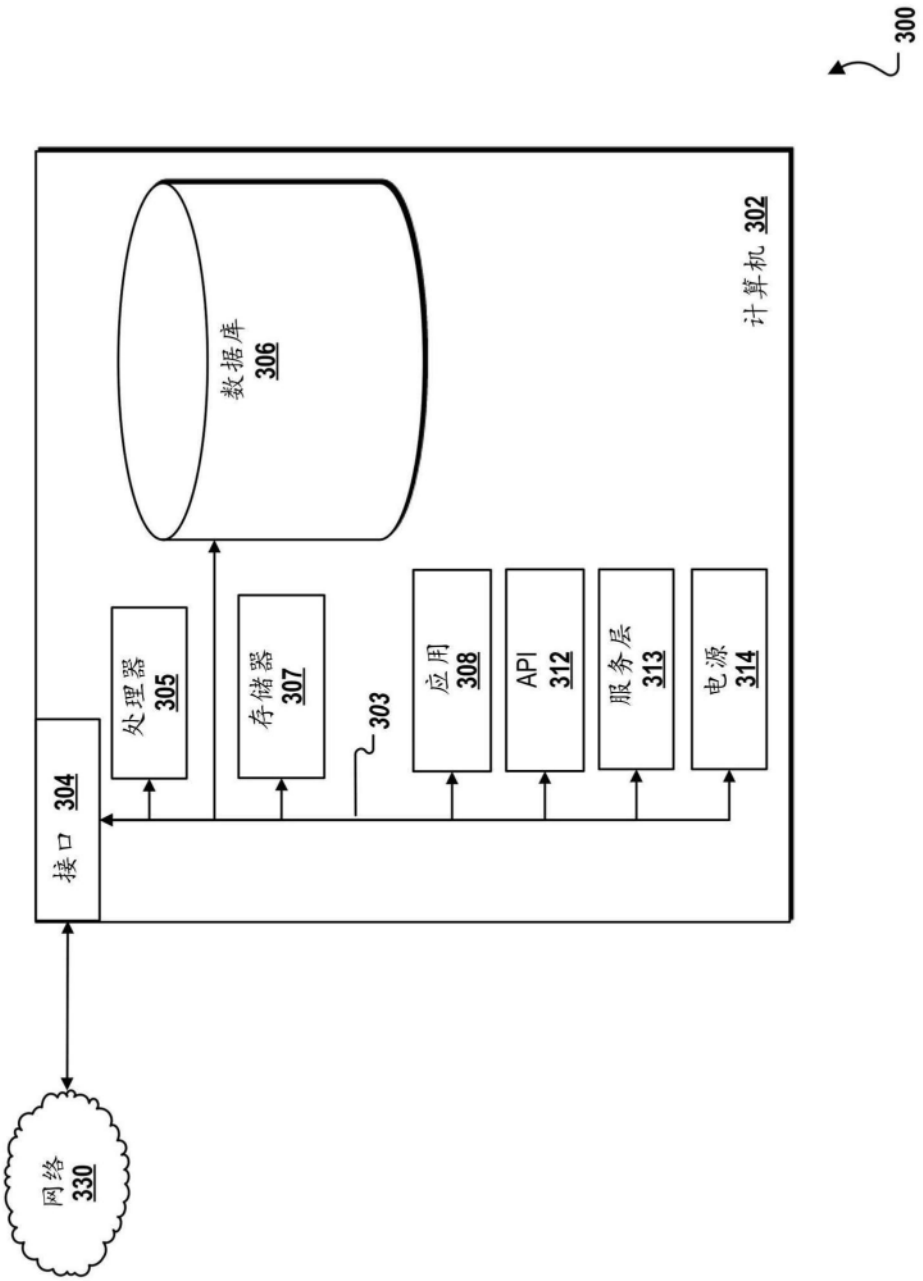

300
计算机 302
数据库 306
处理器 305
存储器 307
303
应用 308
API 312
服务层 313
电源 314
接口 304
网络 330

图3