(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 107209682 A
(43)申请公布日 2017.09.26

(21)申请号 201580066167.0

(22)申请日 2015.12.04

(30)优先权数据
14/562,577 2014.12.05 US

(85)PCT国际申请进入国家阶段日
2017.06.05

(86)PCT国际申请的申请数据
PCT/US2015/064071 2015.12.04

(87)PCT国际申请的公布数据
WO2016/090292 EN 2016.06.09

(71)申请人 亚马逊技术有限公司
地址 美国华盛顿州

(72)发明人 肖恩·菲利普·雷克
德里克·史蒂文·曼纳林
迪伦·钱德勒·托马斯
蒂莫西·艾伦·瓦格纳 赵鑫

(74)专利代理机构 北京天昊联合知识产权代理有限公司 11112
代理人 顾丽波 李荣胜

(51)Int.Cl.
G06F 9/455(2006.01)
G06F 9/50(2006.01)
H04L 12/24(2006.01)

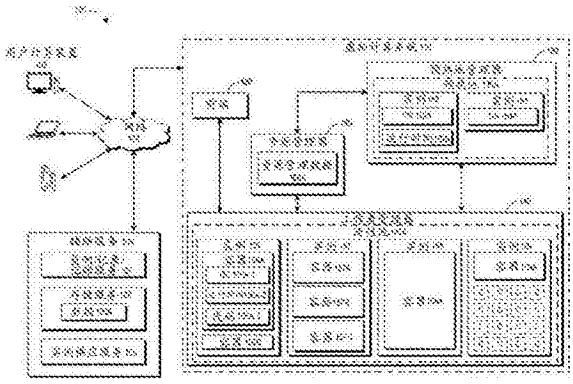
权利要求书3页 说明书20页 附图8页

(54)发明名称

资源调整的自动管理

(57)摘要

本文提供了一种系统，所述系统用于提供自动资源重新调整。所述系统可以被配置以维持多个虚拟机实例。所述系统可以被进一步配置，以接收执行程序代码的请求以及分配计算资源用于在所述虚拟机实例中的一个上执行所述程序代码。分配的用于执行所述程序代码的所述资源量可以由所述请求指定并且可以视需要进行调整。



CN 107209682 A

1.一种被调适以从虚拟计算群提供低延时计算能力的系统,所述系统包括:

电子数据存储区,其被配置以存储用户的至少一个程序代码;以及

虚拟计算系统,其包括被调适以执行特定计算机可执行指令的一个或多个硬件计算装置,所述虚拟计算系统与所述电子数据存储区通信并且被配置以至少:

在一个或多个物理计算装置上维持多个虚拟机实例,其中所述多个虚拟机实例包括:预热池,所述预热池包括上面加载有一个或多个软件部件并且等待被分配给用户的虚拟机实例;以及活性池,所述活性池包括当前分配给一个或多个用户的虚拟机实例;

接收在所述虚拟计算系统上执行与用户相关联的程序代码的请求,所述请求包括指示所述程序代码和与所述程序代码相关联的所述用户的信息;

从所述预热池或所述活性池选择将用于执行所述程序代码的虚拟机实例,所述虚拟机实例具有计算资源;

在所选择的虚拟机实例中创建容器,所述容器具有所述计算资源的第一量;

引起与所述用户相关联的所述程序代码从所述电子数据存储区加载到所述容器上,并且在所述容器中执行;

基于所述程序代码在所述容器中的所述执行确定应调整所述第一量以提高所述程序代码的未来执行;

调整所选择的虚拟机实例中的所述容器,以使得所述容器具有所述计算资源的第二量,所述第二量与所述第一量不同;以及

引起所述程序代码在具有所述计算资源的所述第二量的所述调整的容器中执行。

2.如权利要求1所述的系统,其中所述虚拟计算系统被进一步配置以向所述用户提供通知,所述通知包括以下中的一个:(a)指示应增加所述第一量;(b)指示所述第一量应增加第一调整量;(c)指示应减少所述第一量;或(d)指示所述第一量应减少第二调整量。

3.一种系统,其包括:

虚拟计算系统,其包括执行特定计算机可执行指令的一个或多个硬件计算装置并且被配置以至少:

在一个或多个物理计算装置上维持多个虚拟机实例;

接收在所述虚拟计算系统上执行与用户相关联的程序代码的请求,所述请求包括指示所述程序代码和与所述程序代码相关联的所述用户的信息;

从所述多个虚拟机实例选择将用于执行所述程序代码的虚拟机实例,所选择的虚拟机实例具有计算资源;

在所选择的虚拟机实例中创建容器,所述容器具有所述计算资源的第一量;

引起与所述用户相关联的所述程序代码加载到所述容器上,并且在所述容器中执行;

在所述程序代码在所选择的虚拟机实例中创建的所述容器中的一个或多个执行期间,监测所述程序代码对所述计算资源的使用;以及

基于所述程序代码对所述计算资源的所述使用确定应调整被分配用于执行所述程序代码的所述第一量。

4.如权利要求3所述的系统,其中所述虚拟计算系统被进一步配置以:

基于所述请求中指示的用户指定的量或与所述请求、所述程序代码或在所述虚拟计算系统上执行所述程序代码的所述用户相关联的其它配置信息确定所述第一量;

将所述程序代码对所述计算资源的所述使用与所述请求中指示的所述用户指定的量进行比较;以及

提供应如何调整所述第一量的指示。

5.如权利要求4所述的系统,其中提供所述指示包括以下中的一个:(a)指示应增加所述第一量;(b)指示所述第一量应增加第一调整量;(c)指示应减少所述第一量;或(d)指示所述第一量应减少第二调整量。

6.如权利要求3所述的系统,其中所述虚拟计算系统被进一步配置以:

基于所述程序代码对所述计算资源的所述使用确定所述计算资源的与所述第一量不同的第二量;以及

分配所述计算资源的所述第二量用于执行所述程序代码。

7.如权利要求6所述的系统,其中所述虚拟计算系统被配置以确定所述第二量,以使得所述第二量在与所述程序代码相关联的用户所指定的阈值范围内。

8.如权利要求3所述的系统,其中所述虚拟计算系统被进一步配置,以允许所述程序代码对所述计算资源的所述使用超出所述第一量超出量,所述超出量小于最大超出量。

9.一种计算机实现方法,其包括:

如由配置有特定可执行指令的一个或多个计算装置所实现,

在一个或多个物理计算装置上维持多个虚拟机实例;

接收在所述虚拟计算系统上执行与用户相关联的程序代码的请求,所述请求包括指示所述程序代码和与所述程序代码相关联的所述用户的信息;

从所述多个虚拟机实例选择将用于执行所述程序代码的虚拟机实例,所选择的虚拟机实例具有计算资源;

在所选择的虚拟机实例中创建容器,所述容器具有所述计算资源的第一量;

引起与所述用户相关联的所述程序代码加载到所述容器上,并且在所述容器中执行;

在所述程序代码在所选择的虚拟机实例中创建的所述容器中的一个或多个执行期间,监测所述程序代码对所述计算资源的使用;以及

基于所述程序代码对所述计算资源的所述使用确定应调整被分配用于执行所述程序代码的所述第一量。

10.如权利要求9所述的计算机实现方法,其进一步包括:

基于所述请求中指示的用户指定的量或与所述请求、所述程序代码或在所述虚拟计算系统上执行所述程序代码的所述用户相关联的其它配置信息确定所述第一量;

将所述程序代码对所述计算资源的所述使用与所述请求中指示的所述用户指定的量进行比较;以及

提供应如何调整所述第一量的指示。

11.如权利要求10所述的计算机实现方法,其中提供所述指示包括以下中的一个:(a)指示应增加所述第一量;(b)指示所述第一量应增加第一调整量;(c)指示应减少所述第一量;或(d)指示所述第一量应减少第二调整量。

12.如权利要求9所述的计算机实现方法,其进一步包括:

基于所述程序代码对所述计算资源的所述使用确定所述计算资源的与所述第一量不同的第二量;以及

分配所述计算资源的所述第二量用于执行所述程序代码。

13.如权利要求12所述的计算机实现方法，其中确定所述第二量包括确定所述第二量以使得所述第二量在与所述程序代码相关联的用户所指定的阈值范围内。

14.如权利要求9所述的计算机实现方法，其进一步包括允许所述程序代码对所述计算资源的所述使用超出所述第一量超出量，所述超出量小于最大超出量。

15.一种存储计算机可执行指令的计算机可读、非暂时性存储介质，所述计算机可执行指令当由一个或多个计算装置执行时配置所述一个或多个计算装置以执行操作，所述操作包括：

在一个或多个物理计算装置上维持多个虚拟机实例；

接收在所述虚拟计算系统上执行与用户相关联的程序代码的请求，所述请求包括指示所述程序代码和与所述程序代码相关联的所述用户的信息；

从所述多个虚拟机实例选择将用于执行所述程序代码的虚拟机实例，所选择的虚拟机实例具有计算资源；

在所选择的虚拟机实例中创建容器，所述容器具有所述计算资源的第一量；

引起与所述用户相关联的所述程序代码加载到所述容器上，并且在所述容器中执行；

在所述程序代码在所选择的虚拟机实例中创建的所述容器中的一个或多个执行期间，监测所述程序代码对所述计算资源的使用；以及

基于所述程序代码对所述计算资源的所述使用确定应调整被分配用于执行所述程序代码的所述第一量。

16.如权利要求15所述的计算机可读、非暂时性存储介质，其中所述操作进一步包括：

基于所述请求中指示的用户指定的量或与所述请求、所述程序代码或在所述虚拟计算系统上执行所述程序代码的所述用户相关联的其它配置信息确定所述第一量；

将所述程序代码对所述计算资源的所述使用与所述请求中指示的所述用户指定的量进行比较；以及

提供应如何调整所述第一量的指示。

17.如权利要求16所述的计算机可读、非暂时性存储介质，其中提供所述指示包括以下中的一个：(a)指示应增加所述第一量；(b)指示所述第一量应增加第一调整量；(c)指示应减少所述第一量；或(d)指示所述第一量应减少第二调整量。

18.如权利要求15所述的计算机可读、非暂时性存储介质，其中所述操作进一步包括：

基于所述程序代码对所述计算资源的所述使用确定所述计算资源的与所述第一量不同的第二量；以及

分配所述计算资源的所述第二量用于执行所述程序代码。

19.如权利要求18所述的计算机可读、非暂时性存储介质，其中确定所述第二量包括确定所述第二量以使得所述第二量在与所述程序代码相关联的用户所指定的阈值范围内。

20.如权利要求15所述的计算机可读、非暂时性存储介质，其中所述操作进一步包括允许所述程序代码对所述计算资源的所述使用超出所述第一量超出量，所述超出量小于最大超出量。

资源调整的自动管理

[0001] 其它申请的交叉引用

[0002] 本申请的申请人于2014年9月30日提交了以下美国专利申请：

[0003] 申请No. 标题

[0004] 14/502,992 “THREADING AS A SERVICE”

[0005] 另外，本申请的申请人于2014年12月5日提交了以下美国专利申请：

[0006] 申请No. 标题

[0007] 14/562,601 “AUTOMATIC DETERMINATION OF

[0008] RESOURCE SIZING”

[0009] 以上引用的申请的公开内容特此以引用的方式整体并入。

[0010] 背景

[0011] 一般来说，计算装置利用通信网络或一系列通信网络以交换数据。公司和组织操作互连若干计算装置以支持操作或向第三方提供服务的计算机网络。计算系统可能定位在单个地理位置处，或者定位在多个截然不同的地理位置处(例如，经由专用或公共通信网络互连)。具体地说，数据中心或数据处理中心，在本文中通常称作“数据中心”，可能包括若干互连的计算系统以向数据中心的用户提供计算资源。所述数据中心可能是代表组织操作的专用数据中心，或者可能是代表或为了公众的利益而操作的公共数据中心。

[0012] 为了有助于提高数据中心资源的利用率，虚拟化技术可以允许单个物理计算装置来托管虚拟机的一个或多个实例，所述一个或多个实例作为独立计算装置向数据中心的用户进行显示和操作。在虚拟化技术的情况下，单个物理计算装置可以动态方式创建、维持、删除或另外管理虚拟机。反过来，用户可以从数据中心请求计算机资源，包括单个计算装置或联网计算装置配置，并且被提供有数量不断变化的虚拟机资源。

[0013] 在一些情境中，可以根据若干虚拟机实例类型配置虚拟机实例以提供特定功能。例如，各种计算装置可能与操作系统或操作系统配置、虚拟硬件资源和软件应用的不同组合相关联，以使得计算装置能够更加有效地提供不同的所需的功能，或提供类似的功能。这些虚拟机实例类型配置通常包含在装置图像内，所述装置图像包括包含虚拟机启动时将运行的软件(例如，OS以及应用及其配置和数据文件等)的静态数据。所述装置图像通常存储在用于创建或初始化所述实例的磁盘中。因此，计算装置可以处理装置图像，以便实现所需的软件配置。

[0014] 附图简述

[0015] 本公开的前述方面和伴随的优点中的许多将与通过参考当结合附图进行时的下面的详述变得更好理解一样变得更容易理解，其中：

[0016] 图1是描绘根据示例性方面的用于提供低延时计算能力的说明性环境的框图。

[0017] 图2描绘根据示例性方面的提供用于管理低延时计算能力的资源管理器的计算装置的总体架构。

[0018] 图3是图示根据示例性方面的由资源管理器实现的资源调整例程的流程图。

[0019] 图4是图示根据另一示例性方面的由资源管理器实现的资源调整例程的流程图。

[0020] 图5是图示根据另一示例性方面的由资源管理器实现的资源调整例程的流程图。

[0021] 图6至图8图示根据示例性方面的由资源管理器执行的资源调整。

[0022] 图9至图11图示根据另一示例性方面的由资源管理器执行的资源调整。

[0023] 详述

[0024] 公司和组织不再需要获取和管理其自身的数据中心以便执行计算操作(例如,执行代码,包括线程、程序、功能、软件、例程、子例程、过程等)。随着云计算的出现,现在可以通过互联网在几分钟内获得且配置通常由硬件计算装置提供的存储空间和计算能力。因此,开发人员可以快速地购买所需要的计算资源量,而无需为获取物理机而烦恼。所述计算资源通常呈虚拟计算资源或虚拟机实例的形式进行购买。虚拟机的这些实例是物理机(例如,计算机)的软件实现,所述软件实现托管在物理计算装置上,并且可能包含通常提供在物理机上的操作系统和应用。使用一组计算资源(例如,存储器、CPU、磁盘、网络等)对这些虚拟机实例进行配置,在所述虚拟机实例上运行的应用可以请求并且可以与物理计算机相同的方式利用所述组计算资源。

[0025] 然而,即使当购买了虚拟计算资源(例如,呈虚拟机实例的形式),开发人员仍然需要决定购买多少和购买什么类型的虚拟机实例以及保持它们多长时间。例如,使用虚拟机实例的成本可能会根据它们被租用的类型和小时数而变化。除此以外,虚拟机可以租用的最小时间通常以小时为单位。另外,开发人员需要指定安装在虚拟机上的硬件资源和软件资源(例如,操作系统和语言运行时间的类型等)。他们可能会有的其它问题包括过度利用(例如,获取太少计算资源且遭遇性能问题)、利用不足(例如,获取的计算资源超出运行代码所必要的量,且因此付款过多)、对流量变化的预测(例如,以使得他们知道何时扩展或收缩)以及实例和语言运行时间启动延迟,所述启动延迟可能需要3至10分钟或更长,尽管用户可能期望以秒或甚至毫秒为单位的计算能力。因此,需要允许用户利用由服务提供商所提供的虚拟机实例的改进的方法。

[0026] 根据本公开的方面,通过维持一接收到用户请求就随时可用的预初始化的虚拟机实例池,可以大大缩短与执行用户代码(例如,实例和语言运行时间启动时间)相关联的延迟(有时称作延时)。

[0027] 一般来说,本公开的方面涉及虚拟机实例及其中所创建的容器的管理。具体地说,公开的系统和方法有助于虚拟计算系统中虚拟机实例的管理。虚拟计算系统维持虚拟机实例池,所述虚拟机实例池上加载有一个或多个软件部件(例如,操作系统、语言运行时间、库等)。维持虚拟机实例池可能涉及:创建新实例,从提供服务的外部实例获取新实例,破坏实例,分配/重新分配实例给用户,修改实例(例如,容器或其中的资源)等。池中的虚拟机实例可以被指定用于服务于用户请求以执行程序代码。在本公开中,短语“程序代码”、“用户代码”和“云功能”有时可以互换使用。程序代码可以在虚拟机实例上创建的隔离的容器中执行。因为等到接收请求的时候池中的虚拟机实例已经被启动且加载了特定操作系统和语言运行时间,所以大大缩短了与找出可以处理请求(例如,通过在虚拟机实例上创建的一个或多个容器中执行用户代码)的计算能力相关联的延迟。

[0028] 在另一方面中,虚拟计算系统可以监测和记录与分配的用于执行用户代码的资源量有关的信息。通过这样来做,虚拟计算系统可能够通过调整分配的资源量来识别提高用户代码执行性能的机会。可以通过增加过度利用情况下的分配的资源量来降低错误率,并

且可以通过减少利用不足情况下的分配的资源量来降低与执行用户代码相关联的成本。

[0029] 现在将参考附图描述本公开的特定实施方案和示例性应用。这些实施方案和示例性应用意在说明而非限制本公开。

[0030] 参考图1，将描述图示虚拟环境100的实施方案的框图。图1中示出的示例包括虚拟环境100，其中用户计算装置102的用户(例如，开发人员等)可以使用由虚拟计算系统110提供的虚拟计算资源运行各种程序代码。

[0031] 举例来说，各种示例性用户计算装置102被示出与虚拟计算系统110通信，包括台式计算机、笔记本电脑和手机。一般而言，用户计算装置102可能是诸如台式电脑、笔记本电脑、手机(或智能电话)、平板电脑、信息亭、无线装置和其它电子装置等任何计算装置。除此以外，用户计算装置102可能包括在相同或不同的数据中心上运行的web服务，其中例如不同的web服务可以编程方式彼此通信，以执行本文所描述的一个或多个技术。另外，用户计算装置102可能包括物联网(IoT)装置，诸如互联网电器和连接的装置。虚拟计算系统110可以向用户计算装置102提供一个或多个用户接口、命令行接口(CLI)、应用程序接口(API)和/或其它程序设计接口，以用于生成和上传用户代码，调用用户代码(例如，提交在实施虚拟计算系统110上执行用户代码的请求)，调度基于事件的作业或定时作业，跟踪用户代码，和/或查看与其请求和/或用户代码有关的其它记录或监测信息。虽然本文可能将一个或多个实施方案描述成使用用户接口，但是应了解，所述实施方案可能另外地或可选地使用任何CLI、API或其它程序设计接口。

[0032] 用户计算装置102通过网络104访问虚拟计算系统110。网络104可能是任何有线网络、无线网络或其组合。除此以外，网络104可能是个域网、局域网、广域网、无线广播网(例如，用于收音机或电视)、电缆网、卫星网、蜂窝电话网或以上的组合。例如，网络104可能是链接的网络的公共可接入网络，诸如互联网，所述公共可接入网络可能由各种不同的方操作。在一些实施方案中，网络104可能是专用或半专用网络，诸如企业或大学内联网。网络104可能包括一个或多个无线网络，诸如全球移动通信系统(GSM)网络、码分多址(CDMA)网络、长期演进(LTE)网络或任何其它类型的无线网络。网络104可以使用协议和部件以便经由互联网或其他前面提及的类型的网络中的任何一个进行通信。例如，网络104使用的协议可能包括超文本传送协议(HTTP)、安全HTTP(HTTPS)、消息队列遥测传输(MQTT)、受限应用协议(CoAP)等。用于经由互联网或其它前面提及的类型的通信网络中的任何一个进行通信的协议和部件对于本领域技术人员而言是熟知的，且因此本文中未更加详细地描述。

[0033] 虚拟计算系统110在图1中被描绘成在包括几个计算机系统的分布式计算环境中操作，所述几个计算机系统使用一个或多个计算机网络互连。虚拟计算系统110也可以在具有比图1中所图示的计算环境更少或更多数量的装置的计算环境内操作。因此，图1中对虚拟计算系统110的描绘应被视为是说明性的，而不是对本公开进行限制。例如，虚拟计算系统110或其各种组成可以实现各种Web服务部件、托管的或“云”计算环境和/或对等网络配置，以实现本文所描述的过程的至少一部分。

[0034] 另外，虚拟计算系统110可以在硬件和/或软件中实现，并且可能例如包括在物理计算机硬件上实现的一个或多个物理或虚拟服务器，所述物理计算机硬件被配置以执行计算机可执行指令以用于执行本文将描述的各种特征。所述一个或多个服务器可能是地理分散的或地理共址的，例如在一个或多个数据中心中。

[0035] 在图1中图示的环境中，虚拟环境100包括虚拟计算系统110，所述虚拟计算系统110包括前端120、预热池管理器130、工作者管理器140和资源管理器150。在描绘的示例中，虚拟机实例(“实例”)152、154被示出在由预热池管理器130管理的预热池130A中，且实例156、157、158、159被示出在由工作者管理器140管理的活性池140A中。对虚拟计算系统110内的各种部件的图示本质上是逻辑的，并且所述部件中的一个或多个可以由单个计算装置或多个计算装置实现。例如，实例152、154、156、157、158、159可以在位于各种不同的地理区域中的一个或多个物理计算装置上实现。类似地，前端120、预热池管理器130、工作者管理器140和资源管理器150中的每一个可以跨多个物理计算装置实现。可选地，前端120、预热池管理器130、工作者管理器140和资源管理器150中的一个或多个可以在单个物理计算装置上实现。在一些实施方案中，虚拟计算系统110可能包括多个前端、多个预热池管理器、多个工作者管理器和/或多个能力管理器。虽然图1的示例中示出六个虚拟机实例，但是本文所描述的实施方案并不受如此限制，并且本领域技术人员将了解，虚拟计算系统110可能包括使用任何数量的物理计算装置实现的任何数量的虚拟机实例。类似地，虽然图1的示例中示出单个预热池和单个活性池，但是本文所描述的实施方案并不受如此限制，并且本领域技术人员将了解，虚拟计算系统110可能包括任何数量的预热池和活性池。

[0036] 在图1的示例中，虚拟计算系统110被图示成连接至网络104。在一些实施方案中，虚拟计算系统110内的部件中的任何一个可以经由网络104与虚拟环境100的其它部件(例如，用户计算装置102和辅助服务106，其可能包括监测/记录/记账服务107、存储服务108、实例供应服务109和/或可以与虚拟计算系统110通信的其它服务)通信。在其它实施方案中，并不是虚拟计算系统110的所有部件都能够与虚拟环境100的其它部件通信。在一个示例中，仅前端120可以连接至网络104，并且虚拟计算系统110的其它部件可以经由前端120与虚拟环境100的其它部件通信。

[0037] 用户可以使用虚拟计算系统110执行其上面的用户代码。例如，用户可能希望结合用户已经开发的web或移动应用运行一段代码。一种运行代码的方法将是从提供基础设施作为服务的服务提供商获取虚拟机实例，配置所述虚拟机实例以适合用户的需求，以及使用所述配置的虚拟机实例以运行所述代码。可选地，用户可以发送代码执行请求至虚拟计算系统110。虚拟计算系统110可以基于代码执行请求处理计算能力(例如，容器、实例等，其在下面更加详细地描述)的获取和配置，并使用所述计算能力执行代码。虚拟计算系统110可以基于容量自动地扩展和收缩，从而将用户从不得不担心过度利用(例如，获取太少计算资源且遭遇性能问题)或利用不足(例如，获取的计算资源超出运行代码所必要的量，且因此付款过多)的负担中解脱出来。

[0038] 前端120处理在虚拟计算系统110上执行用户代码的所有请求。在一个实施方案中，前端120充当通往由虚拟计算系统110提供的所有其它服务的正门。前端120处理请求，并确保所述请求被适当地授权。例如，前端120可以确定与请求相关联的用户是否被授权访问请求中指定的用户代码。

[0039] 如本文中所使用的用户代码可能指代以特定程序语言编写的任何程序代码(例如，程序、例程、子例程、线程等)。在本公开中，术语“代码”、“用户代码”和“程序代码”可以互换使用。可以例如结合由用户开发的特定web应用或移动应用执行所述用户代码以完成特定任务。例如，用户代码可以JavaScript(node.js)、Java、Python和/或Ruby编写。请求可

能包括用户代码(或其位置)以及用于执行用户代码的一个或多个自变量。例如,用户可以提供用户代码以及执行所述用户代码的请求。在另一示例中,所述请求可以通过先前上传的程序代码的名称或其唯一ID将其识别(例如,使用用于上传所述代码的API)。在另一示例中,在虚拟计算系统110接收到请求之前,所述代码可能包括在所述请求中并且上传到分开的位置(例如,存储服务108或虚拟计算系统110内部的存储系统)。虚拟计算系统110可以基于处理请求时代码位于何处来改变其代码执行策略。

[0040] 前端120可以响应于来自用户的安全超文本传送协议(HTTPS)请求来接收执行所述用户代码的请求。而且,当执行用户代码时也可以处理和利用包括在HTTPS请求中的任何信息(例如,报头和参数)。如上面所讨论,包括例如HTTP、MQTT和CoAP的任何其它协议可以用于传送包含代码执行请求的消息至前端120。当检测到事件时前端120也可以接收执行所述用户代码的请求,所述事件诸如用户已经登记以触发自动请求生成的事件。例如,用户可能已经向辅助服务106登记所述用户代码并且指定当特定事件发生时(例如,上传新文件),发送执行用户代码的请求至前端120。可选地,用户可能已经登记了定时作业(例如,每24小时执行用户代码)。在此种示例中,当到了定时作业的预定时间时,可以发送执行用户代码的请求至前端120。在另一示例中,前端120可能具有一队列传入代码执行请求,并且当从虚拟计算系统的工作列队移除用户的批量作业时,前端120可以处理用户请求。在另一示例中,所述请求可能源自虚拟计算系统110内的另一部件或图1中未图示的其它服务器或服务。

[0041] 用户请求可以指定将一个或多个第三方库(包括本地库)与用户代码一起使用。在一个实施方案中,用户请求是包含用户代码和任何库(和/或其存储位置的标识)的ZIP文件。在一些实施方案中,用户请求包括指示以下信息的元数据:将执行的程序代码,编写程序代码的语言,与请求相关联的用户,和/或将被保存用于执行程序代码的计算资源(例如,存储器、CPU、存储装置、网络包等)。例如,程序代码可能设有先前由用户上传的、由虚拟计算系统110(例如,标准例程)提供和/或由第三方提供的请求。在一些实施方案中,资源级约束(例如,将分配多少存储器用于执行特定用户代码)被指定用于特定用户代码,并且在用户代码的每一次执行过程中可能不会改变。在所述情况下,虚拟计算系统110可以在接收到每一个别请求之前访问所述资源级约束,并且所述个别请求可能不会指定所述资源级约束。在一些实施方案中,资源级约束随着时间推移进行调整,并且在单个程序代码的不同执行过程中可能会变化。例如,相同的程序代码可以用于处理两组不同的数据,其中一组数据比另一组数据需要更多的资源。在此种情况下,用户可以为两个不同的执行指定不同的资源约束,或者虚拟计算系统110可以基于空间(例如,在虚拟计算系统110的其它部分中)或历史(例如,随时间推移的)趋势为用户和/或程序代码自动地调整分配至程序代码的每一执行的资源量。在一些实施方案中,用户请求可以指定诸如指示请求具有以执行用户代码的何种许可的许可数据等其它约束。所述许可数据可以被虚拟计算系统110用于访问专用资源(例如,在专用网络上)。

[0042] 在一些实施方案中,用户请求可以指定应被采用以便处理用户请求的行为。在所述实施方案中,用户请求可能包括用于启用一个或多个执行模式的指示符,与用户请求相关联的用户代码将在所述一个或多个执行模式中执行。例如,所述请求可能包括用于指示用户代码是否应在调试模式中执行的标志或报头,在所述调试模式下,可以结合用户代码

的执行生成的调试和/或记录输出被提供回给用户(例如,经由控制台用户接口)。在此种示例中,虚拟计算系统110可以检查请求并寻找标志或报头,并且如果存在,那么虚拟计算系统110可以修改执行用户代码的容器的行为(例如,记录设施),并且引起输出数据被提供回给用户。在一些实施方案中,行为/模式指示符被由虚拟计算系统110提供给用户的用户接口添加至所述请求。也可以基于提供在请求中的指示启用或禁用诸如资源代码剖析、远程调试等其它特征。

[0043] 在一些实施方案中,虚拟计算系统110可能包括多个前端120。在所述实施方案中,可以提供负载平衡器以例如呈循环方式将传入请求分配至多个前端120。在一些实施方案中,负载平衡器将传入请求分配至多个前端120的方式可能基于预热池130A和/或活性池140A的状态。例如,如果预热池130A中的容量被认为是充分的,那么请求可以基于前端120的个别能力(例如,基于一个或多个负载平衡限制)分配至多个前端120。另一方面,如果预热池130A中的容量小于阈值量,那么可以移除所述负载平衡限制中的一个或多个,以使得请求可以将从预热池130A取得的虚拟机实例的数量减少或最小化的方式分配至多个前端120。例如,虽然根据负载平衡限制,请求将被路由至前端A,但是如果前端A需要从预热池130A取出实例以服务于所述请求但是前端B可以使用在其活性池中的所述实例中的一个来服务于相同的请求,那么所述请求可以路由至前端B。

[0044] 预热池管理器130确保当虚拟计算系统110接收到在虚拟计算系统110上执行用户代码的请求时,虚拟机实例准备由工作者管理器140使用。在图1中图示的示例中,预热池管理器130管理预热池130A,所述预热池130A是可以用于服务于传入用户代码执行请求的预初始化和预配置的虚拟机实例的群组(有时称作池)。在一些实施方案中,预热池管理器130引起虚拟机实例在虚拟计算系统110内的一个或多个物理计算机器上被启动,并且被添加至预热池130A。在其它实施方案中,预热池管理器130与辅助虚拟机实例服务(例如,图1的实例供应服务109)通信,以创建和添加新的实例至预热池130A。在一些实施方案中,预热池管理器130可以利用虚拟计算系统110内的两个物理计算装置以及一个或多个虚拟机实例服务以获取和维持计算能力,所述计算能力可以用于服务于前端120所接收的代码执行请求。在一些实施方案中,虚拟计算系统110可能包括用于控制(例如,增大或减小)预热池130A中的可用容量的一个或多个逻辑旋钮或开关。例如,系统管理员可以使用此种旋钮或开关来增加高峰时段期间预热池130A中的可用容量(例如,预启动的实例的数量)。在一些实施方案中,预热池130A中的虚拟机实例可以基于预定的一组配置进行配置,所述一组配置独立于执行用户的代码的特定用户请求。预定的一组配置可能对应于用于执行用户代码的各种类型的虚拟机实例。预热池管理器130可以基于与当前或先前用户代码执行有关的一个或多个度量优化预热池130A中的虚拟机实例的类型和数量。

[0045] 如图1中所示,实例上可能加载有操作系统(OS)和/或语言运行时间。例如,由预热池管理器130管理的预热池130A包括实例152、154。实例152包括OS 152A和运行时间152B。实例154包括OS154A。在一些实施方案中,预热池130A中的实例可能还包括容器(其可能进一步包含操作系统、运行时间、用户代码等的副本),所述容器在下面更加详细地描述。虽然图1中示出实例152包括单个运行时间,但是在其它实施方案中,图1中描绘的实例还可能包括两个或两个以上运行时间,所述运行时间中的每一个可用于运行不同的用户代码。在一些实施方案中,预热池管理器130可以在预热池130A中维持实例列表。实例列表可以进一步

指定实例的配置(例如,OS、运行时间、容器等)。

[0046] 在一些实施方案中,预热池130A中的虚拟机实例可以用于服务任何用户的请求。在一个实施方案中,以相同或大致上类似的方式配置预热池130A中的所有虚拟机实例。在另一实施方案中,预热池130A中的虚拟机实例可以不同方式进行配置以适应不同用户的需要。例如,虚拟机实例上面可能加载有不同的操作系统,不同的语言运行时间,和/或不同的库。在另一实施方案中,可以相同或大致上类似的方式配置预热池130A中的虚拟机实例(例如,具有相同的OS、语言运行时间和/或库),但是这些实例中的一些可能具有不同的容器配置。例如,两个实例可能具有用于Python和Ruby二者的运行时间,但是一个实例可能具有被配置以运行Python代码的容器,且另一个实例可能具有被配置以运行Ruby代码的容器。在一些实施方案中,提供多个预热池130A,每一预热池具有被相同地配置的虚拟机实例。

[0047] 预热池管理器130可以预配置预热池130A中的虚拟机实例,以使得每一虚拟机实例被配置以满足可以请求的或由用于在虚拟计算系统110上执行程序代码的用户请求所指定的操作条件中的至少一个。在一个实施方案中,操作条件可能包括可能写入了潜在用户代码的程序语言。例如,所述语言可能包括Java、JavaScript、Python、Ruby等。在一些实施方案中,可能用来写入用户代码的语言集合可能限于预定的集合(例如,4种语言的集合,虽然一些实施方案中提供多于或少于四种语言的集合),以便有助于可能满足执行用户代码的请求的虚拟机实例的预初始化。例如,当用户经由虚拟计算系统110提供的用户接口配置请求时,所述用户接口可以提示用户指定用于执行用户代码的预定操作条件中的一个。在另一示例中,用于利用虚拟计算系统110提供的服务的服务级协议(SLA)可以指定用户请求应满足的一组条件(例如,编程语言、计算资源等),并且虚拟计算系统110可以假定请求满足处理请求的一组条件。在另一示例中,请求中指定的操作条件可能包括:将用于处理请求的计算能力的量;请求的类型(例如,HTTP与触发的事件);用于请求的超时(例如,阈值时间,之后请求可能会被终止);安全性策略(例如,可以控制预热池130A中的哪些实例可被哪个用户使用);以及等等。

[0048] 工作者管理器140管理用于服务于传入代码执行请求的实例。在图1中图示的示例中,工作者管理器140管理活性池140A,所述活性池是当前分配给一个或多个用户的虚拟机实例的群组(有时称作池)。虽然此处将虚拟机实例描述成被分配给特定用户,但是在一些实施方案中,所述实例可以分配给一组用户,以使得所述实例与所述一组用户相关联,并且所述群组的任何成员可以利用所述实例上的资源。例如,相同群组中的用户可能属于相同的安全群组(例如,基于其安全凭证),以使得已经在特定实例上的容器中执行一个成员的代码之后在相同实例上的另一容器中执行另一成员的代码不会构成安全风险。类似地,工作者管理器140可以根据一个或多个策略分配实例和容器,所述一个或多个策略规定哪些请求可以在哪些容器中执行以及哪些实例可以分配给哪些用户。示例性策略可以指定实例被分配给共享相同的账户(例如,用于访问由虚拟计算系统110提供的服务的账户)的用户的集合。在一些实施方案中,与相同的用户群组相关联的请求可以共享相同的容器(例如,如果相关联的用户代码是相同的)。在一些实施方案中,请求不区分群组的不同用户,并且简单地指示与所述请求相关联的用户所属的群组。

[0049] 在图1中图示的示例中,用户代码在被称作容器的隔离的计算系统中执行。容器是使用虚拟机实例上可用的资源创建在所述实例内的逻辑单元。例如,工作者管理器140可以

基于执行用户代码的请求中指定的信息来创建新的容器或将现有容器定位在活性池140A中的实例中的一个中，并将所述容器分配给所述请求以处理与所述请求相关联的用户代码的执行。在一个实施方案中，所述容器被实现为Linux容器。活性池140A中的虚拟机实例上可能创建有一个或多个容器，并且上面加载有与用户相关联的一个或多个程序代码(例如，在容器中的一个中，或在实例的本地缓存中)。

[0050] 如图1中所示，实例可能具有操作系统(OS)、语言运行时间和容器。容器上可能加载有OS和语言运行时间以及用户代码的副本。在图1的示例中，由工作者管理器140管理的活性池140A包括实例156、157、158、159。实例156具有容器156A、156B。容器156A中加载有OS 156A-1、运行时间156A-2和代码156A-3。在描绘的示例中，容器156A中加载有其自身的OS、运行时间和代码。在一个实施方案中，OS 156A-1(例如，其内核)、运行时间156A-2和/或代码156A-3在容器156A、156B(以及图1中未图示的任何其它容器)之间共享。在另一实施方案中，OS 156A-1(例如，在内核以外运行的任何代码)、运行时间156A-2和/或代码156A-3是独立副本，所述独立副本被创建用于容器156A，并且未与实例156上的其它容器共享。在另一实施方案中，OS 156A-1、运行时间156A-2和/或代码156A-3中的一些部分在实例156上的容器之间共享，并且其它部分是特定于容器156A的独立副本。实例157包括容器157A、157B、157C，实例158包括容器158A，并且实例159包括容器159A。

[0051] 在图1的示例中，图1中描绘的容器的大小可能与容器的实际大小成比例。例如，容器156A在实例156上占用的空间比容器156B占用的空间更大。类似地，容器157A、157B、157C、159A可能是相等大小的，且容器158A可能比容器157A、157B、157C、159A更大(例如，分配有更多的计算资源)。实例159中示出的被标记为“C”的虚线框指示实例上剩余的可以用于创建新实例的空间。在一些实施方案中，容器的大小可能是64MB或其任何倍数。在其它实施方案中，容器的大小可能是小于或等于创建有容器的实例的大小的任何任意大小。在一些实施方案中，容器的大小可能是小于、等于或大于创建有容器的实例的大小的任何任意大小。容器的大小可能超出实例的大小多少可以基于超出实例所提供的能力利用这些容器的可能性有多大进行确定。例如，具有1GB(总计5GB)的存储器大小的五个容器可以创建在具有4GB的存储器大小的实例中。如果容器中的每一个未达到1GB的全容量，那么所述容器可以正常工作，但是存在过度订阅。

[0052] 虽然图1的示例中未图示容器156B、157A、157B、157C、158A、159A内的部件，但是这些容器中的每一个可能具有各种操作系统、语言运行时间、库和/或用户代码。在一些实施方案中，实例上可能加载有用户代码(例如，在实例级缓存中)，并且这些实例内的容器中可能还加载有用户代码。在一些实施方案中，工作者管理器140可以在活性池140A中维持实例列表。实例列表可以进一步指定实例的配置(例如，OS、运行时间、容器等)。在一些实施方案中，工作者管理器140可以访问预热池130A中的实例列表(例如，包括实例的数量和类型)。在其它实施方案中，工作者管理器140从预热池管理器130请求计算能力，而无需知晓预热池130A中的虚拟机实例。

[0053] 在前端120已经成功地处理请求之后，工作者管理器140找出服务于在虚拟计算系统110上执行用户代码的请求的能力。例如，如果活性池140A中存在具有加载有相同用户代码(例如，容器156A中示出的代码156A-3)的容器的特定虚拟机实例，那么工作管理器140可以分配容器至所述请求，并且引起用户代码在容器中执行。可选地，如果用户代码在虚拟机

实例中的一个的本地缓存中可用(例如,存储在实例158上,但是不属于任何个别容器),那么工作者管理器140可以在此种实例上创建新的容器,将所述容器分配给请求,并引起所述用户代码在容器中加载和执行。

[0054] 如果工作者管理器140确定在活性池140A中的实例的任何一个(例如,容器中或实例的本地缓存中)上未找到与请求相关联的用户代码,那么工作者管理器140可以确定活性池140A中的实例的任何一个是否当前被分配给与请求相关联的用户以及是否具有用于处理当前请求的计算能力。如果存在此种实例,那么工作者管理器140可以在实例上创建新的容器,并且将所述容器分配给所述请求。可选地,工作者管理器140可以进一步配置被分配至用户的实例上的现有容器,并且将所述容器分配至所述请求。例如,工作者管理器140可以确定如果加载了当前用户请求所要求的特定库,那么所述现有容器可以用于执行用户代码。在此种情况下,工作者管理器140可以加载所述特定库和用户代码到容器上,并且使用容器来执行用户代码。

[0055] 如果活性池140A不包含当前分配给用户的任何实例,那么工作者管理器140从预热池130A拉出新的虚拟机实例,将所述实例分配给与请求相关联的用户,在实例上创建新的容器,将所述容器分配给请求,以及引起所述用户代码在容器上下载和执行。

[0056] 在一些实施方案中,虚拟计算系统110被调适以在接收(例如,由前端120接收)到用户代码不久之后开始用户代码的执行。可以将一个时间段确定为启动用户代码的执行(例如,在与用户相关联的虚拟机实例上的容器中)与接收用于执行用户代码的请求(例如,由前端接收)之间的时间差。虚拟计算系统110被调适以在小于预定的持续时间的时间段内开始用户代码的执行。在一个实施方案中,预定的持续时间是500ms。在另一实施方案中,预定的持续时间是300ms。在另一实施方案中,预定的持续时间是100ms。在另一实施方案中,预定的持续时间是50ms。在另一实施方案中,预定的持续时间是10ms。在另一实施方案中,预定的持续时间可能是从10ms至500ms的范围选择的任何值。在一些实施方案中,虚拟计算系统110被调适以当满足一个或多个条件时在小于预定的持续时间的时间段内开始用户代码的执行。例如,一个或多个条件可能包括以下中的任何一个:(1)在接收到请求时用户代码被加载到活性池140A中的容器上;(2)在接收到请求时用户代码被存储在活性池140A中的实例的代码缓存中;(3)在接收到请求时活性池140A包含分配给与所述请求相关联的用户的实例;或者(4)在接收到请求时预热池130A具有处理请求的能力。

[0057] 用户代码可以从辅助服务106诸如图1的存储服务108下载。图1中图示的数据108A可能包括:一个或多个用户上传的用户代码,与所述用户代码相关联的元数据,或者被虚拟计算系统110利用以执行本文所描述的一个或多个技术的任何其它数据。虽然图1的示例中仅图示存储服务108,但是虚拟环境100可能包括可以下载用户代码的其它级别的存储系统。例如,每一实例可能具有与上面创建有容器的实例物理地相关联的一个或多个存储系统(例如,驻留在上面运行有容器的物理计算系统上的本地存储装置)或逻辑地相关联的一个或多个存储系统(例如,与实例网络通信且提供在虚拟计算系统110内部或外侧的网络附接的存储系统)。可选地,所述代码可以从由存储服务108提供的基于web的数据存储区下载。

[0058] 一旦工作者管理器140将虚拟机实例中的一个定位在可以用于服务于用户代码执行请求的预热池130A中,预热池管理器130或工作者管理器140就从预热池130A取出实例,

并将其分配给与请求相关联的用户。分配的虚拟机实例被从预热池130A取出，并且被放置到活性池140A中。在一些实施方案中，一旦虚拟机实例已经分配给特定用户，相同的虚拟机实例就不能用于服务于任何其它用户的请求。这通过防止可能的用户资源的混合存放来向用户提供安全利益。可选地，在一些实施方案中，属于不同用户的多个容器(或分配给与不同用户相关联的请求)可能共存于单个虚拟机实例上。此种方法可以提高可用计算能力的利用率。在一些实施方案中，虚拟计算系统110可以维持单独的缓存，所述缓存中存储有用户代码以充当虚拟机实例的本地缓存与基于web的网络存储装置(例如，经由网络104可访问)之间的中间级缓存系统。

[0059] 在已经执行用户代码之后，工作者管理器140可以断开用于执行用户代码的容器以释放其占用的资源，以便用于实例中的其它容器。可选地，工作者管理器140可以保持容器运行，以将其用于服务于来自同一用户的额外请求。例如，如果另一请求与已经在容器中加载的相同用户代码相关联，那么所述请求可以被分配给相同的容器，由此消除与创建新的容器和在容器中加载用户代码相关联的延迟。在一些实施方案中，工作者管理器140可以断开创建有用于执行用户代码的容器的实例。可选地，工作者管理器140可以保持实例运行，以将其用于服务于来自同一用户的额外请求。对于完成执行用户代码之后是否保持容器和/或实例运行的确定可能基于阈值时间、用户的类型、用户的平均请求量和/或其它操作条件。例如，在已经过了阈值时间(例如，5分钟、30分钟、1小时、24小时、30天等)而无任何活动(例如，代码的运行)之后，容器和/或虚拟机实例被关闭(例如，删除、终止等)，且分配至所述容器和/或虚拟机实例的资源被释放。在一些实施方案中，在容器断开之前过去的阈值时间比实例断开之前过去的阈值时间短。

[0060] 在一些实施方案中，虚拟计算系统110可以当它服务于传入代码执行请求时提供数据给辅助服务106中的一个或多个。例如，虚拟计算系统110可以与监测/记录/记账服务107通信。监测/记录/记账服务107可能包括：用于管理从虚拟计算系统110接收的监测信息的监测服务，所述监测信息诸如虚拟计算系统110上的容器和实例的状态；用于管理从虚拟计算系统110接收的记录信息的记录服务，所述记录信息诸如由虚拟计算系统110上的容器和实例执行的活动；以及用于生成与在虚拟计算系统110上执行用户代码相关联的记账信息的记账服务(例如，基于由监测服务和记录服务管理的监测信息和/或记录信息)。除了如上所述可以由监测/记录/记账服务107(例如，代表虚拟计算系统110)执行的系统级活动以外，监测/记录/记账服务107还可以代表在虚拟计算系统110上执行的用户代码提供应用级服务。例如，监测/记录/记账服务107可以代表在虚拟计算系统110上执行的用户代码监测和/或记录各种输入、输出或其它数据和参数。虽然示出为单个块，但是监测、记录和记账服务107可以作为分开的服务提供。监测/记录/记账服务107可以与资源管理器150通信，以允许资源管理器150确定将用于在虚拟计算系统150上执行各种程序代码的合适的资源量。

[0061] 在一些实施方案中，工作者管理器140可以在由工作者管理器140管理的实例和容器(例如，活性池140A中的那些)上执行健康检查。例如，由工作者管理器140执行的健康检查可能包括确定由工作者管理器140管理的实例和容器是否具有以下问题中的任何问题：(1)错误配置的联网和/或启动配置，(2)存储器耗尽，(3)文件系统损坏，(4)内核不兼容和/或可能会损害实例和容器的性能的任何其它问题。在一个实施方案中，工作者管理器140定期执行健康检查(例如，每5分钟、每30分钟、每小时、每24小时等)。在一些实施方案中，健康

检查的频率可以基于健康检查的结果自动地调整。在其它实施方案中，健康检查的频率可以基于用户请求进行调整。在一些实施方案中，工作者管理器140可以在预热池130A中的实例和/或容器上执行类似健康检查。预热池130A中的实例和/或容器可以与活性池140A中的那些实例和容器一起管理或分开管理。在一些实施方案中，在预热池130A中的实例和/或容器的健康与活性池140A分开管理的情况下，预热池管理器130而不是工作者管理器140可以在预热池130A中的实例和/或容器上执行如上所述的健康检查。

[0062] 资源管理器150管理被分配用于处理在虚拟计算系统110上执行用户代码的传入请求的资源量。例如，资源管理器150可以与前端120、预热池管理器130、工作者管理器140和/或辅助服务106通信，以监测和管理分配给(和用于)在虚拟计算系统110上执行的各种程序代码的计算能力。虽然资源管理器150被图示成虚拟计算系统110内的截然不同的部件，但是资源管理器150的部分或全部功能可以由前端120、预热池管理器130、工作者管理器140和/或辅助服务106执行。例如，资源管理器150可以全部在虚拟计算系统110的其它部件中的一个内实现，或者以分布式方式跨虚拟计算系统110的其它部件实现。在图1的示例中，资源管理器150包括资源管理数据150A。资源管理数据150A可能包括有关以下信息的数据：传入请求的历史(例如，与特定程序代码相关联的传入请求的量，接收到这些请求的高峰时段等)；由传入请求指定的资源级约束；分配给传入请求的资源量；所述分配的资源的实际上被传入请求所利用的部分；以及可以被资源管理器150使用以监测、记录、调整、提高和/或优化在虚拟计算系统110上执行的程序代码的性能(例如，分配的资源的利用率)的任何其它特性或度量。资源管理数据150A可能还包括由用户指定的或由资源管理器150确定的用于管理虚拟计算系统110上的资源的任何管理策略，这在下面更加详细地描述。

[0063] 如上面所讨论，请求本身可以指定将用于执行与请求相关联的程序代码的计算资源(例如，存储器、CPU、存储装置、网络包等)的量。在此种请求已经被处理且虚拟机实例已经分配给与请求相关联的用户之后，资源管理器150可以分配请求中指定的资源量给请求，做法是通过创建具有相同的资源量的容器。例如，如果请求指定512MB的存储器将用于执行与请求相关联的程序代码，那么资源管理器150可以在分配给与请求相关联的用户的实例上创建具有512MB的存储器大小的容器。在一些实施方案中，与请求、程序代码或用户相关联的其它配置信息可以指定将用于执行程序代码的计算资源量。所述配置信息可以随同请求或与请求分开提供给虚拟计算系统110。资源管理器150可以维持请求中指定的资源量作为用于与请求相关联的特定程序代码的默认设置，并且为在虚拟计算系统110上处理的任何后续请求使用相同的量。在一些实施方案中，如果一个或多个后续请求指示与默认的量不同的资源量将分配给一个或多个后续请求，那么资源管理器150可以允许一个或多个后续请求超控此种默认设置。例如，后续请求可能指示比默认的量高出10％的资源量将分配给后续请求以用于执行程序代码。在确定后续请求指示替代的资源量将分配给后续请求以用于执行程序代码时，资源管理器150分配替代的资源量给后续请求。

[0064] 在一些实施方案中，请求中指定的资源量可能与分配给请求的实际资源量不同。例如，在一些情景中，虚拟计算系统110可以向请求分配超过请求中指定的资源量阈值百分比的资源量。在其它情况下，虚拟计算系统110可以向请求分配低于请求中指定的资源量阈值百分比的资源量。对特定资源是过度订阅还是订阅不足可以基于特定资源、用户、请求和/或提供特定资源的物理硬件(例如，与其相关联的任何容差或方差)的类型进行确定。在

一些实施方案中，分配给请求的资源量可能小于最大值(或高于请求中指定的量的百分比)和/或大于最小值(或低于请求中指定的量的百分比)，这基于特定资源、用户、请求和/或提供特定资源的物理硬件(例如，与其相关联的任何容差或方差)的类型进行确定。

[0065] 在一些实施方案中，请求中指定的资源的特定类型(例如，存储器)充当用于分配虚拟计算系统110上可用的其它类型资源的指导方针。例如，如果请求指定512MB的存储器将用于执行与请求相关联的程序代码，且分配给用户的实例上的存储器的总量(例如，物理或虚拟最大值)或可用的量(例如，当前未被其它容器占用的资源)是2GB，那么实例上可用的其它类型的资源(例如，CPU、存储装置、网络包等)也以成比例的量(例如，CPU的四分之一、存储装置的四分之一、网络包的四分之一等被分配给容器)进行分配。另一方面，如果分配给用户的实例上的存储器的总量或可用的量是1GB，那么CPU的一半、存储装置的一半、网络包的一半将被分配给容器。在一些实施方案中，可以由请求指定或者可以另外分配给容器的存储器的量在从64MB至1GB的范围内，增量为64MB。在一些实施方案中，请求可以指定其它量和/或可以分配给容器其它量。虽然存储器被用作示例，但是可以选择和使用任何其它资源作为用于设置将被分配用于处理请求(例如，执行程序代码)的所有资源量的指导方针。在一些实施方案中，单数形式的资源(例如，最易于理解、最用户友好、最基本、绝对数量最大或绝对数量最小的资源)被选择以代表所有其它形式的资源。请求可以指定可使用百分比而不是特定类型的资源量来分配所有资源。另外，请求可以指定一个以上资源的量。

[0066] 在一些实施方案中，资源管理器150可以定位具有指定的资源量的现有容器并且引起程序代码在现有容器中执行，而不是创建新的容器以及分配指定的资源量给所述容器。分配给现有容器的资源量并非正好匹配指定的资源量，但是在指定的资源量的阈值百分比内。在一些实施方案中，资源管理器150可以通过分配更少或更多的计算资源量来重新调整现有容器，并指定具有调整的资源大小的现有容器来处理与特定程序代码相关联的程序代码。资源管理器150是否可以动态地重新调整现有容器可能取决于程序代码所使用的语言运行时间。例如，Java运行时间可能不允许动态重新调整，而Python运行时间可能允许动态重新调整。

[0067] 在已经创建或定位到具有指定资源量的容器之后，在所述容器中执行与请求相关联的程序代码。分配给容器的资源量(例如，用户所请求的)和/或程序代码实际利用的资源量可以记录(例如，由监测/记录/记账服务107和/或资源管理器150记录)以用于进一步分析。例如，记录的信息可能包括程序代码在容器中的一个或多个执行期间实际使用的存储器的量、CPU周期的量、网络包的量以及存储装置的量。另外，记录的信息可能包括资源利用、错误率、延时和程序代码的执行期间遇到的任何错误或意外情况。在一些实施方案中，与分配给容器的资源量有关的任何错误(例如，存储器不足意外情况)标记有特殊记号并且由资源管理器150进一步分析。

[0068] 在一些实施方案中，资源管理器150可以创建或访问多个用户类别，并且为不同用户类别应用不同的规则。例如，对于较为复杂的用户，可以给予较多控制(例如，对整个个别资源参数的控制)，而对于其它用户，可以允许其控制仅单个代表性参数，且其它参数可以基于所述代表性参数进行调整。

[0069] 在一些实施方案中，资源管理器150可以基于监测/记录/记账服务107和/或资源管理器150记录的信息向用户提供有关以下的一些引导：用户可以执行哪些操作以提高程

序代码的性能或降低与在虚拟计算系统110上执行程序代码相关联的成本。例如，在看到存储器不足意外情况重复发生之后，资源管理器150可以向用户提供用户似乎一直将用于运行特定用户代码的存储器(或其它资源)设置过低的指示。类似地，在确定特定用户代码的调用长期以来仅使用分配给其的资源的小部分之后，资源管理器150可以向用户提供用户可能将存储器(或其它资源)设置过高的指示。所述指示可以指定特定资源应调整的量。在一些实施方案中，此种指示在资源管理器150已经处理了阈值数量个错误、意外情况或其它明显条件(例如，延时增加)之后提供给用户。资源管理器150可以经由包括电子邮件、推送通知服务、SMS、社交网络服务等任何通知机制提供指示。在一些实施方案中，如果将对一个或多个资源进行调整的量超出阈值或阈值百分比，那么向用户提供有关需要资源大小调整的指示。例如，如果资源管理器150确定用户指定的存储器大小应增加0.5％以实现更好或最佳性能，那么资源管理器150可能根本不会发送通知给用户，但是如果资源管理器150确定用户指定的存储器大小应增加10％以实现更好或最佳性能，那么资源管理器150可以发送通知给用户。

[0070] 在一些实施方案中，资源管理器150可以提供有限的过度订阅，其中允许程序代码的执行超出请求的资源量。例如，如果请求指定了64MB的存储器，那么资源管理器150可以允许程序代码使用至多70MB或80MB的存储器。在此种情况下，程序代码可以成功地执行，但是可能会向用户提供有关以下的通知：程序代码超出请求的存储器量，且执行程序代码的未来请求应指定更大的存储器量。过度订阅可能在阈值数量的使用之后到期。

[0071] 在一些实施方案中，虚拟计算系统110可以自动地调整分配给个别代码执行请求的资源量。例如，在用户可以指定资源参数的其它实施方案中，如果用户未能指定合适的资源量，那么程序代码的执行的性能结果可能会受损。例如，如果请求指定64MB的存储器将用于执行实际上需要1GB进行运行的特定程序代码，那么用户可能会碰到许多问题(例如，程序代码可能直接无法执行)。如果虚拟计算系统110依赖用户调整请求中指定的资源量，那么在用户解决问题之前，虚拟计算系统110可能会接收到成千上万的失败请求。在此种情境下，当检测到请求中指定的资源量不足时，资源管理器150可以自动地调整分配给执行特定程序代码的传入请求的资源量。在一些实施方案中，此种调整在资源管理器150已经处理了阈值数量个错误、意外情况或其它明显条件(例如，延时增加)之后进行。因此，虽然前几个请求即使在资源管理器150做出自动资源调整的情况下仍然会失败，但是后续请求最后可以产生所需的结果，甚至不需要用户干预。

[0072] 在一些实施方案中，资源管理器150可以利用代码特定特性以提高用于执行个别程序代码的资源大小。例如，负责图像处理的程序代码可能需要大量存储器，而执行数据库访问的程序代码可能不需要如此多的存储器。在另一示例中，对于特定程序代码，大多数时候64MB可能是足够的，但是虚拟计算系统110可能会在每晚8点钟接收到与用户相关联的代码执行请求的爆发，对此，资源管理器150可以分配较多存储器给负责处理与用户相关联的请求的容器和/或实例。所述代码特定特性可以由资源管理器150维持，而个别程序代码的资源大小可以相应地进行调整。

[0073] 在一些实施方案中，资源管理器150可以最初分配最大资源量给程序代码，并且在运行程序代码之后如果资源管理器150确定程序代码实际上使用最大量的1/10，那么资源管理器150可以为执行程序代码的后续请求分配最大量的一半。如果资源管理器150仍然确

定程序代码使用最大量的1/10，那么资源管理器150可以进一步将分配给程序代码的资源量减半。资源管理器150可以重复所述过程，直到程序代码使用分配给程序代码的很大一部分资源(例如，50％、75％或另一阈值)为止。

[0074] 在一些实施方案中，用户可以指定规定资源管理器150的行为的资源管理策略。例如，对价格很敏感的用户可能乐意忽略偶尔的错误，前提是这样做可以将他或她的成本最小化。因此，此类用户可能不愿意增加分配的用于执行他或她的程序代码的资源量，即使当前指定的资源量偶尔会造成存储器不足错误。另一方面，对错误很敏感的用户可能愿意采取任何措施以便避免错误或失误(例如，存储器不足错误、非常高的延时或某一其它问题)发生。此类用户可能更喜欢将他或他的程序代码遇到的错误的数量最小化，即使分配的用于执行程序代码的资源有时利用不足。在一些实施方案中，用户可以指定有关资源的成本、利用率、量等的止损极限(例如，下限和/或上限)。用户可以进一步限定所述止损极限，以使得其仅在指定的时间段适用。例如，用户可以指定用户在执行程序代码上愿意花费的最小金额和最大金额，但是用户也可以进一步指定在每一季度的最后一周期间将不应用所述限制。

[0075] 在一些实施方案中，资源管理器150可以选择性地过度订阅某些资源，前提是资源管理器150确定代码执行请求未充分利用所述资源。例如，资源管理器150可以分配1GB的存储器给具有4GB虚拟或物理存储器的虚拟机实例上的五个不同容器。过度订阅的程度可能基于创建在单个实例上的这些容器可能集体耗尽实例的最大资源量的可能性有多大。在一些实施方案中，用户可以被给予选择降低的成本利用过度订阅的资源的选项。

[0076] 在一些实施方案中，请求可能不会指定任何资源级约束，而是资源量由虚拟计算系统110自动确定以及资源量由虚拟计算系统110确定的请求被传达给用户(例如，以使得用户知晓被分配用于服务于用户的请求的资源量)。可选地，请求可以指定将用于运行用户代码的资源量，但是还指示如果虚拟计算系统110确定可以过度订阅请求的资源量(例如，通过与其他用户共享)而不会显著影响代码执行性能，那么虚拟计算系统110可以通过向用户少收费(例如，低于指定的资源量通常将花费的成本)来实现。

[0077] 在一些实施方案中，通过收缩或扩张特定容器调整分配给特定容器的资源量(例如，通过修改分配给现有容器的资源量)。可选地，可以通过创建具有调整的资源量的新容器以及通过注入对新容器的任何未来请求让旧容器停止运转来调整资源量。在一些实施方案中，重新调整基于程序代码所使用的特定语言运行时间的特性执行(例如，一些可能允许动态资源重新调整，而另一些可能不允许)。

[0078] 资源管理器150可能包括：如果有的话，用于调整用于执行用户代码的容器(例如，确定将分配给容器的资源量)的资源调整单元，以及用于向用户提供有关资源量应如何调整的通知的资源引导单元。资源管理器150的示例性配置在下面参考图2更加详细地描述。

[0079] 图2描绘管理虚拟计算系统110中的虚拟机实例的计算系统(称作资源管理器150)的总体架构。图2中描绘的资源管理器150的总体架构包括可以用于实现本公开的方面的计算机硬件和软件模块的布置。资源管理器150可能包括比图2中示出的那些元件更多(或更少)的元件。然而，并不需要示出所有这些大体常见的元件以便提供可实现的公开。如图所示，资源管理器150包括处理单元190、网络接口192、计算机可读介质驱动器194、输入/输出装置接口196，全部可以借助于通信总线彼此通信。网络接口192可以提供至一个或多个网

络或计算系统的连通性。处理单元190因此可以经由网络104从其它计算系统或服务接收信息和指令。处理单元190还可以向存储器180传达和从中传达，并且还可以经由输入/输出装置接口196进一步提供用于任选显示器(未图示)的输出信息。输入/输出装置接口196还可以从任选输入装置(未图示)接受输入。

[0080] 存储器180可能包含计算机程序指令(在一些实施方案中分组为模块)，处理单元190执行所述计算机程序指令以便实现本公开的一个或多个方面。存储器180大体上包括RAM、ROM和/或其它持久的、辅助的或非暂时性计算机可读介质。存储器180可以存储操作系统184，所述操作系统184提供在资源管理器150的一般管理和操作过程中由处理单元190使用的计算机程序指令。存储器180可能进一步包括计算机程序指令和用于实现本公开的方面的其它信息。例如，在一个实施方案中，存储器180包括生成用于在计算装置上显示的用户接口(和/或指令)的用户接口单元182，例如经由安装在计算装置上的诸如浏览器或应用的导航和/或浏览接口。除此以外，存储器180可能包括一个或多个数据储存库(未图示)和/或与一个或多个数据储存库(未图示)通信，例如以访问用户程序代码和/或库。

[0081] 除用户接口单元182以外和/或结合用户接口单元182，存储器180可能包括可以由处理单元190执行的资源调整单元186和资源引导单元188。在一个实施方案中，用户接口单元182、资源调整单元186和资源引导单元188独立地或共同地实现本公开的各种方面，例如：监测和记录程序代码在虚拟计算系统110上的执行，确定对分配给特定容器和/或请求的资源量进行调整的需要，向用户提供有关需要调整资源量的通知，自动地调整资源量等，如下面进一步描述。

[0082] 资源调整单元186监测用户代码在虚拟计算系统110上的执行，提供具有用于执行用户代码的指定的资源量的容器，以及调整分配给容器的资源量。例如，如果资源调整单元186确定用于执行特定程序代码的请求一直出现存储器不足错误，那么资源调整单元186可以增加分配给用于执行特定程序代码的后续请求的存储器的量。另一方面，如果资源调整单元186确定用于执行特定程序代码的请求一直仅使用分配给请求的资源的一部分，那么所述资源调整单元186可以减少分配给用于执行特定程序代码的后续请求的存储器的量。

[0083] 资源引导单元188向用户提供有关需要调整分配的用于服务于与用户相关联的请求的资源量的通知。例如，所述通知可能指示用户一直将用于执行特定程序代码的量指定过低或过高。所述通知可能进一步指定应如何调整资源量以获得提高的性能或最佳性能。

[0084] 虽然资源调整单元186和资源引导单元188在图2中被示出为资源管理器150的一部分，但是在其它实施方案中，资源调整单元186和资源引导单元188中的全部或一部分可以由虚拟计算系统110的其它部件和/或另一计算装置实现。例如，在本公开的某些实施方案中，与虚拟计算系统110通信的另一计算装置可能包括几个模块或部件，所述几个模块或部件以与被图示为资源管理器150的一部分的模块和部件类似的方式操作。

[0085] 现在转到图3，将描述由虚拟计算系统110的一个或多个部件(例如，资源管理器150)实现的例程300。虽然相关于资源管理器150的实现描述了例程300，但是相关领域技术人员应了解，可选部件可以实现例程300或者块中的一个或多个可以由不同的部件或以分布式方式实现。

[0086] 在说明性例程300的块302中，资源管理器150基于在虚拟计算系统110上执行程序代码的请求确定用户指定的第一计算资源的量。例如，第一计算资源可能是可以用于在虚

拟计算系统110上执行程序代码的存储器、CPU、磁盘空间或任何其它计算资源。在虚拟计算系统110上执行程序代码的请求可以指示将分配多少资源用于执行程序代码。

[0087] 接着，在块304中，资源管理器150基于用户指定的第一计算资源的量确定第二计算资源的对应的量。例如，资源管理器150可以通过计算用户指定的第一计算资源的量与可用于分配给请求的第一计算资源的最大量之间的比率来确定第二计算资源的对应的量。如果用户指定的量是512MB的存储器，且可用于分配给请求的最大存储器量是1GB，那么资源管理器150可以确定对应的量应是可用于分配给请求的第二计算资源的最大量的一半。例如，如果第二计算资源是磁盘空间，且8GB的磁盘空间可用于分配，那么资源管理器150可以确定对应的量应是4GB。

[0088] 在块306中，资源管理器150为执行程序代码的请求分配用户指定的第一计算资源(例如，存储器)的量和第二计算资源(例如，磁盘空间)的对应的量。在上面的示例中，资源管理器150可以在虚拟计算系统上可用的虚拟机实例中的一个上创建容器，其中所述容器分配有512MB的存储器和4GB的磁盘空间。虽然图3的示例中使用了第一计算资源和第二计算资源，但是也可以使用额外资源，且所述额外资源也可以根据用户指定的量与第一计算资源的最大量之间的比率进行调整。

[0089] 虽然上面已经参考块302至块306描述了图3的例程300，但是本文所描述的实施方案并不受如此限制，并且在不脱离本公开的精神的情况下可以省略、修改或切换一个或多个块。

[0090] 现在转到图4，将描述由虚拟计算系统110的一个或多个部件(例如，资源管理器150)实现的例程400。虽然相关于资源管理器150的实现描述了例程400，但是相关领域技术人员应了解，可选部件可以实现例程400或者块中的一个或多个可以由不同的部件或以分布式方式实现。

[0091] 在说明性例程400的块402中，资源管理器150基于在虚拟计算系统110上执行程序代码的请求确定用户指定的计算资源的量。例如，计算资源可能是可以用于在虚拟计算系统110上执行程序代码的存储器、CPU、磁盘空间或任何其它计算资源。用户指定的量可能是请求中指示的特定资源(例如，存储器)的量(例如，包括在请求中或可另外基于包括在请求中的信息确定)，所述请求指定将分配多少特定资源用于执行程序代码。例如，程序代码的开发人员可以在代码执行请求中指定将分配多少存储器(或其它计算资源)用于执行他或她的程序代码。

[0092] 接着，在块404中，资源管理器150分配用户指定的计算资源的量用于执行程序代码。例如，资源管理器150可以在与用户相关联且在活性池140A中的实例中创建具有用户指定的计算资源的量的容器。在另一示例中，资源管理器150可以从预热池130A选择实例，将选择的实例分配给用户，以及在选择的实例中创建具有用户指定的计算资源的量的容器。

[0093] 在块406中，资源管理器150在程序代码的一个或多个执行期间监测程序代码对计算资源的使用。例如，资源管理器150可以引起程序代码被加载到指定用于处理请求的容器上，并且引起程序代码在所述容器中执行。资源管理器150可以监测程序代码的一个或多个执行期间的一个或多个性能特性。所述性能特性可能包括：错误率、资源利用率、延时、资源使用％、用户请求的资源％等。

[0094] 在块408中，资源管理器150将程序代码对计算资源的使用与用户指定的量进行比

较，并且在块410中，资源管理器150基于所述比较确定应调整用户指定的被分配用于执行程序代码的计算资源的量。例如，如果用户请求了512MB的存储器但是在程序代码的一个或多个执行期间平均仅使用了64MB，那么资源管理器150可以确定可以减少用户指定的量而不显著影响程序代码的执行。另一方面，如果用户请求了512MB的存储器且在程序代码的一个或多个执行期间几乎全部请求的量均被使用，那么资源管理器150可以确定应增加用户指定的量以解决过度利用问题。

[0095] 在块412中，资源管理器150提供应如何调整用户指定的量的指示。例如，资源管理器150可以向用户提供电子邮件通知，指示用户指定的量应增加或减少合适的量。

[0096] 虽然上面已经参考块402至块412描述了图4的例程400，但是本文所描述的实施方案并不受如此限制，并且在不脱离本公开的精神的情况下可以省略、修改或切换一个或多个块。例如，虽然参考单个计算资源(例如，存储器)描述了例程400，但是可以在剩余资源维(例如，CPU、网络、磁盘等)中的每一个上执行类似技术。另外，虽然例程400被描述成在根据资源的基础上监测和比较单个计算资源(例如，将请求的存储器量与实际使用的存储器量进行比较，将请求的处理能力的量与实际使用的处理能力的量进行比较，等)，但是也可以使用类似技术在综合的基础上执行例程400。例如，资源管理器150可以将请求的或分配的和请求相关联的最受限资源的量与和请求相关联的程序代码所使用的最受限资源的实际量进行比较，并且如果请求的或分配的最受限资源的量在低于或高于由程序代码所使用的最受限资源的实际量的阈值百分比内，则块412中的指示可以提供给用户。在另一示例中，资源管理器150可以计算资源维中的每一个的平均利用率百分比，并且如果所述平均百分比低于或高于阈值利用率值(例如，低于10％，高于90％等)则向用户提供指示。提供给用户的指示可能还包括计算资源中的每一个的过度使用或使用不足的分类。例如，资源管理器150可以提供说明以下情况的指示：“您当前的资源大小刻度盘处在512MB，对存储器性能而言很好，因为您对存储器的使用从未超过该值，但是我们已经注意到在当前设置下常常会用光网络资源。您可以将资源大小刻度盘上调10％，以获得更多网络资源以及实现提高的代码执行性能”。

[0097] 现在转到图5，将描述由虚拟计算系统110的一个或多个部件(例如，资源管理器150)实现的例程500。虽然相关于资源管理器150的实现描述了例程500，但是相关领域人员将了解，可选部件可以实现例程500或者块中的一个或多个可以由不同的部件或以分布式方式实现。

[0098] 在说明性例程500的块502中，资源管理器150分配计算资源的第一量用于执行程序代码。例如，资源管理器150可以在与用户相关联且在活性池140A中的实例中创建具有计算资源的第一量的容器。在另一示例中，资源管理器150可以从预热池130A选择实例，将选择的实例分配给用户，以及在选择的实例中创建具有计算资源的第一量的容器。计算资源可能是可以用于在虚拟计算系统110上执行程序代码的存储器、CPU、磁盘空间或任何其它计算资源。所述第一量可以由资源管理器150基于包括在请求中的信息和/或基于所述包括在请求中的信息可确定的信息进行确定。所述信息可能包括用于对程序代码进行编码的编程语言、用户类型(例如，高用量用户或低用量用户)、程序代码的特性(例如，行数、昂贵调用的数量等)等。

[0099] 接着，在块504中，资源管理器150在程序代码的一个或多个执行期间监测程序代

码对计算资源的使用。例如，资源管理器150可以引起程序代码被加载到指定用于处理请求的容器上，并且引起程序代码在所述容器中执行。资源管理器150可以监测程序代码的一个或多个执行期间的一个或多个性能特性。所述性能特性可能包括：错误率、资源利用率、延时、资源使用％、用户请求的资源％等。

[0100] 在块506中，资源管理器150基于程序代码对计算资源的使用确定应调整被分配用于执行程序代码的计算资源的第一量。例如，如果在块502中分配了512MB的存储器用于执行程序代码，但是在程序代码的一个或多个执行期间平均仅使用了64MB，那么资源管理器150可以确定可以减少分配用于执行程序代码的量而不显著影响程序代码的性能。另一方面，如果分配了512MB的存储器用于执行程序代码且在程序代码的一个或多个执行期间几乎全部分配的量均被使用，那么资源管理器150可以确定应增加分配用于执行程序代码的量以解决过度利用问题。

[0101] 在块508中，资源管理器150确定计算资源的与所述第一量不同的第二量，并且在块510中，资源管理器150分配计算资源的第二量用于执行程序代码。在分配了512MB且平均使用了64MB的示例中，资源管理器150可以确定分配的存储器的量可以安全地减小至128MB而不影响程序代码的性能，并分配128MB用于执行程序代码(例如，用于程序代码的未来执行)。

[0102] 虽然上面已经参考块502至块512描述了图5的例程500，但是本文所描述的实施方案并不受如此限制，并且在不脱离本公开的精神的情况下可以省略、修改或切换一个或多个块。例如，如结合图4所讨论，虽然参考单个计算资源(例如，存储器)描述了例程500，但是可以在剩余资源维(例如，CPU、网络、磁盘等)中的每一个上执行类似技术。另外，虽然例程500被描述成在根据资源的基础上监测和比较单个计算资源(例如，将当前分配的存储器量与实际使用的存储器量进行比较，将当前分配的处理能力的量与实际使用的处理能力的量进行比较，等)，但是也可以使用类似技术在综合的基础上执行例程500。例如，资源管理器150可以将请求的或分配的和请求相关联的最受限资源的量与和请求相关联的程序代码所使用的最受限资源的实际量进行比较，并且块506中的确定可以基于请求的或分配的最受限资源的量是否在低于或高于由程序代码所使用的最受限资源的实际量的阈值百分比内。在另一示例中，资源管理器150可以计算资源维中的每一个的平均利用率百分比，并且基于所述平均百分比是否低于或高于阈值利用率值(例如，低于10％，高于90％等)进行确定。

[0103] 参考图6至图8，图示由资源管理器150根据示例性实施方案执行的资源重新调整。在图6的示例中，容器158A被利用以处理与特定程序代码相关联的传入代码执行请求。如图6中所示，容器158A具有27％的平均资源利用率。例如，特定程序代码在容器158A中的一个或多个执行期间，特定程序代码已经利用了分配给容器158A的27％的资源。资源管理器150可以将资源利用率与阈值(例如，75％)进行比较，并确定当前资源利用率过低且应减少分配给容器158A的资源量。

[0104] 在图7中，资源管理器150已经减少了分配给容器158A以用于执行特定程序代码的资源量(例如，应与实例158相关联的用户的请求，或基于由资源管理器150基于利用率做出的确定)，并且在特定程序代码在容器158A中的一个或多个额外执行之后，资源利用率已经提高至40％。虽然已经调整了现有容器(例如，容器158A)的资源大小，但是在另一实施方案中，可以替代地创建分配有调整的资源量的新容器，并且可以耗尽现有容器。如图8中所示，

分配给容器158A的资源量已经被进一步调整，以将资源利用率提高至80％。资源管理器150可以将资源利用率水平与阈值(例如，75％)进行比较，并且确定进一步资源重新调整是不必要的，并且将容器158A的资源水平维持在当前水平，直到资源管理器150稍后确定需要进一步重新调整为止。图6至图8中示出的其它部件与图1中示出的那些部件相同，并且因此为简洁起见，所述部件的细节被省略。

[0105] 参考图9至图11，图示由资源管理器150根据另一示例性实施方案执行的资源重新调整。在图9的示例中，容器159A被利用以处理与特定程序代码相关联的传入代码执行请求。如图9中所示，容器159A具有90％的平均错误率。例如，特定程序代码的执行中的90％可能已经产生一个或多个错误或者未能成功执行。资源管理器150可以将资源利用率与阈值(例如，5％)进行比较，并确定当前错误率过高且应增加分配给容器159A的资源量。

[0106] 在图10中，资源管理器150已经增加了分配给容器159A以用于执行特定程序代码的资源量(例如，应与实例159相关联的用户的请求，或基于由资源管理器150基于错误率做出的确定)，并且在特定程序代码在容器159A中的一个或多个额外执行之后，错误率已经降低至20％。虽然已经调整了现有容器(例如，容器159A)的资源大小，但是在另一实施方案中，可以替代地创建分配有调整的资源量的新容器，并且可以耗尽现有容器。如图11中所示，分配给容器159A的资源量已经被进一步调整，以将错误率提高至3％。资源管理器150可以将错误率与阈值(例如，5％)进行比较，并且确定进一步资源重新调整是不必要的，并且将容器159A的资源水平维持在当前水平，直到资源管理器150稍后确定需要进一步重新调整为止。图9至图11中示出的其它部件与图1中示出的那些部件相同，并且因此为简洁起见，所述部件的细节被省略。

[0107] 本领域及其他的技术人员将了解，本公开中所描述的所有功能可能体现在由公开的部件和移动通信装置的一个或多个物理处理器执行的软件中。所述软件可以持久地存储在任何类型的非易失性存储装置中。

[0108] 除非另有特别说明或者另外根据使用在上下文内理解，否则尤其诸如“可能(“can”、“could”、“might”或“may”)”等条件语言通常意在表达某些实施方案包括而其它实施方案不包括某些特征、要素和/或步骤。因此，所述条件语言一般并非意在暗示特征、要素和/或步骤无论如何都是一个或多个实施方案所必需的，或者一个或多个实施方案无论有或没有用户输入或提示都必然包括用于决定任何特定实施方案中是否包括或将执行这些特征、要素和/或步骤的逻辑。

[0109] 本文所描述的和/或附图中所描绘的流程图中的任何过程描述、要素或块应被理解成可能代表代码的包括用于实现过程中的特定逻辑功能或步骤的一个或多个可执行指令的模块、区段或部分。替代实现包括在本文所描述的实施方案的范围内，其中要素或功能可以被删除，可以按不同于所示出或所讨论的顺序来执行，包括大致上同时地或者逆序地，这取决于所涉及的功能，如本领域技术人员将理解。将进一步了解的是，上述数据和/或部件可以存储在计算机可读介质上，并使用与存储诸如CD-ROM、DVD-ROM或网络接口等计算机可执行部件的计算机可读存储介质相关联的驱动机构加载到计算装置的存储器中。另外，所述部件和/或数据可能包括在单个装置中或以任何方式分布。因此，通用计算装置可以被配置，以使用上述处理和/或执行各种数据和/或部件实现本公开的过程、算法和方法论。

[0110] 应强调的是，可以对上述实施方案做出许多变化和修改，其要素将被理解成也存

在于其它可接受示例中。所有所述修改和变化在本文中意在包括在本公开的范围内且受以下权利要求保护。

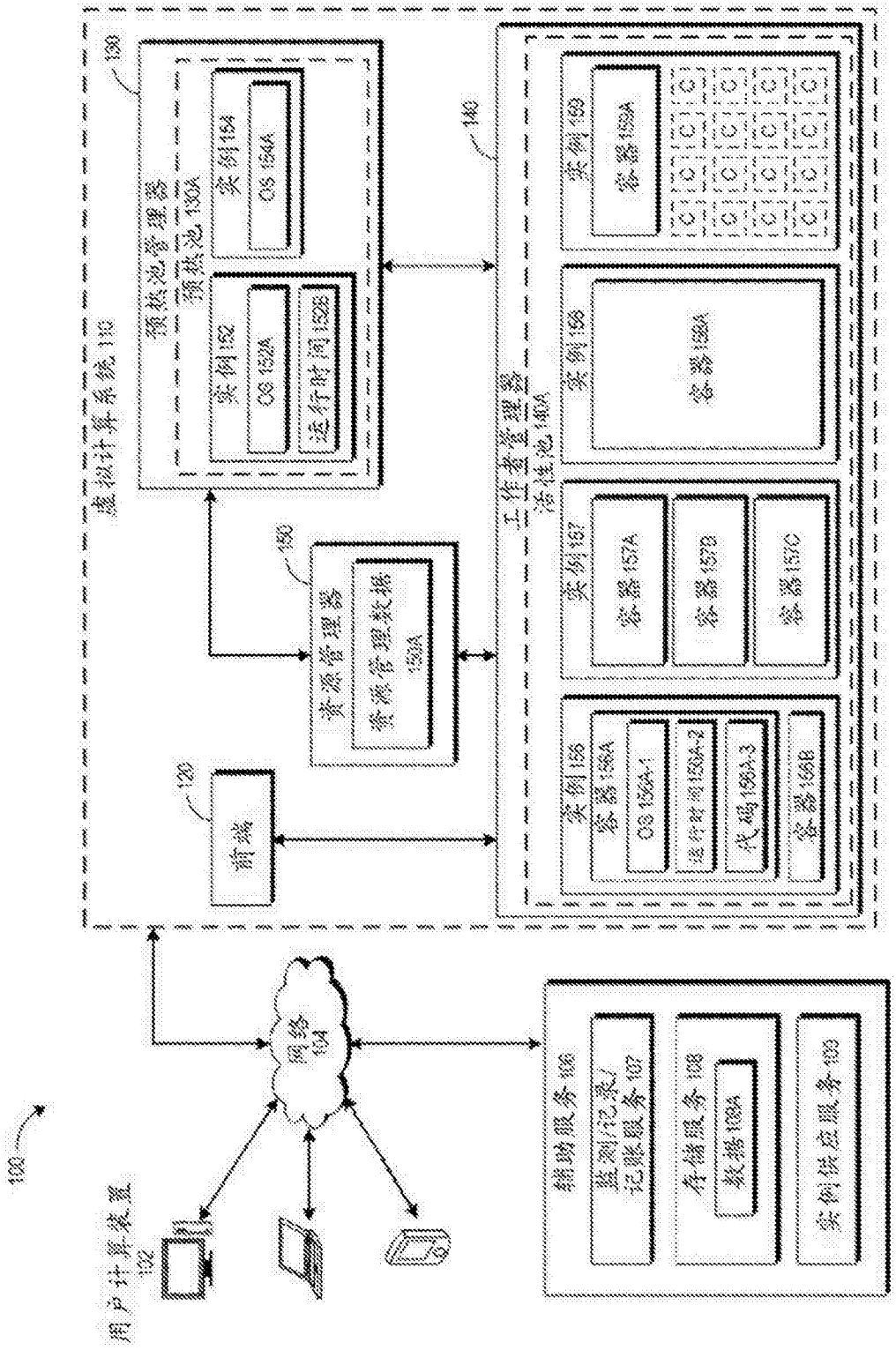

100
用户计算装置
102
网络
104
辅助服务 106
监测/记录/记账服务 107
存储服务 108
数据 108A
实例供应服务 109
虚拟计算系统 110
预热池管理器
预热池 130A
实例 154
OS 154A
实例 152
OS 152A
运行时间 152B
140
工作者管理器
活性池 140A
实例 193
容器 193A
实例 196
容器 196A
实例 187
容器 187A
容器 187B
容器 187C
实例 155
容器 155A
OS 150A-1
运行时间 150A-2
代码 150A-3
容器 155B
150
资源管理器
资源管理数据 150A
120
前端

图1

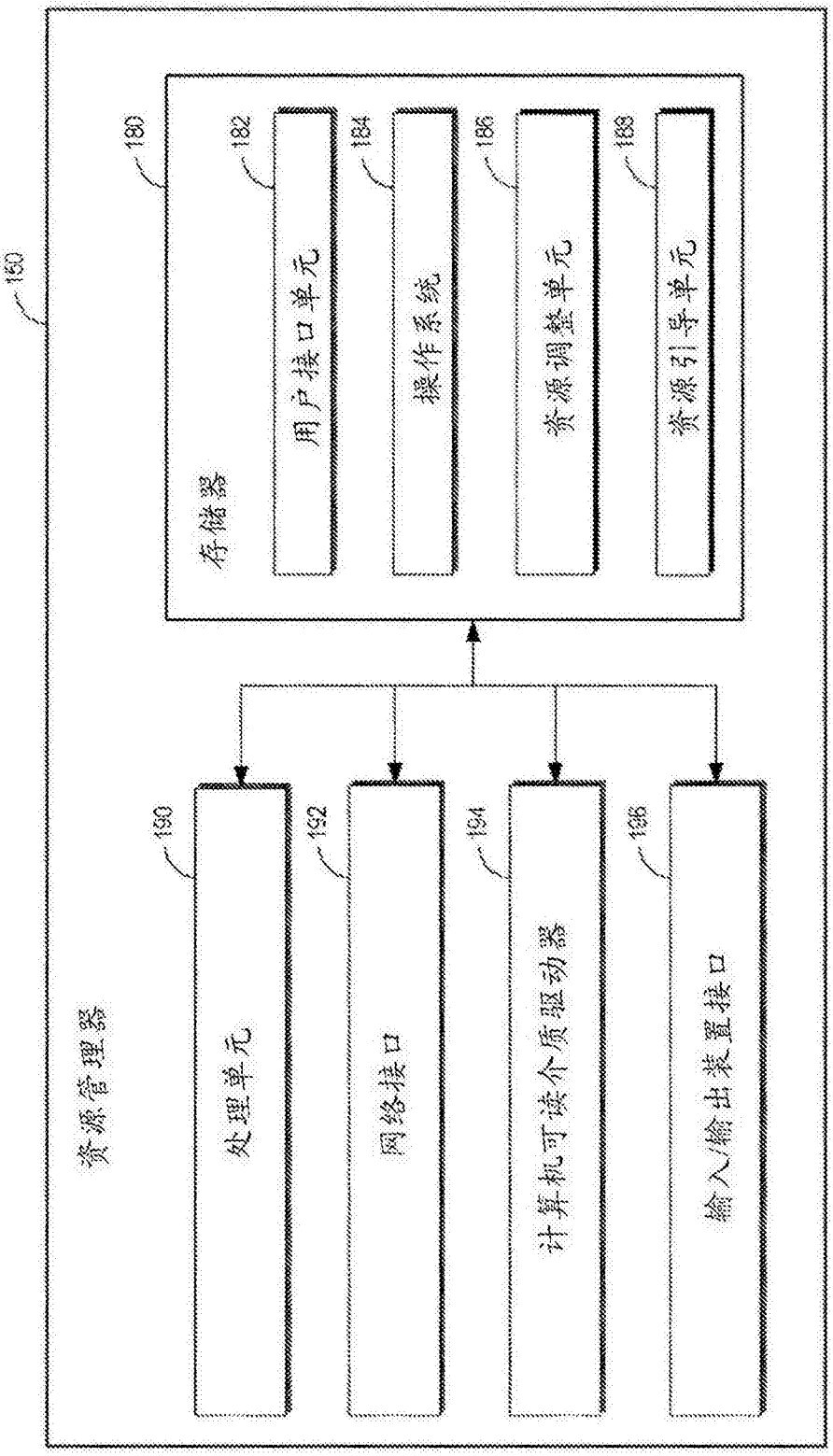

150
资源管理器
180
存储器
182
用户接口单元
184
操作系统
186
资源调整单元
188
资源引导单元
190
处理单元
192
网络接口
194
计算机可读介质驱动器
196
输入/输出装置接口

图2

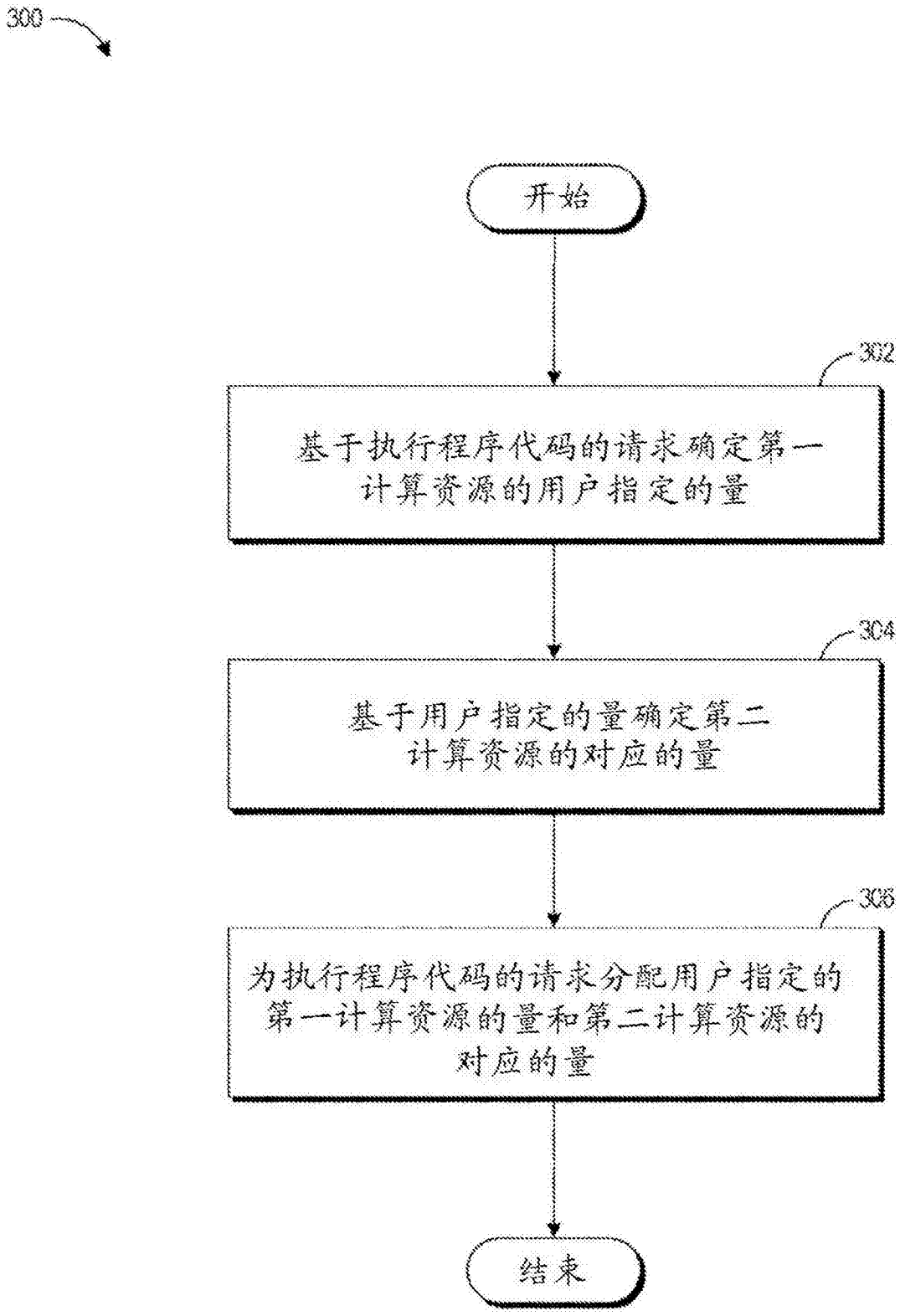

300
开始
302
基于执行程序代码的请求确定第一计算资源的用户指定的量
304
基于用户指定的量确定第二计算资源的对应的量
306
为执行程序代码的请求分配用户指定的第一计算资源的量和第二计算资源的对应的量
结束

图3

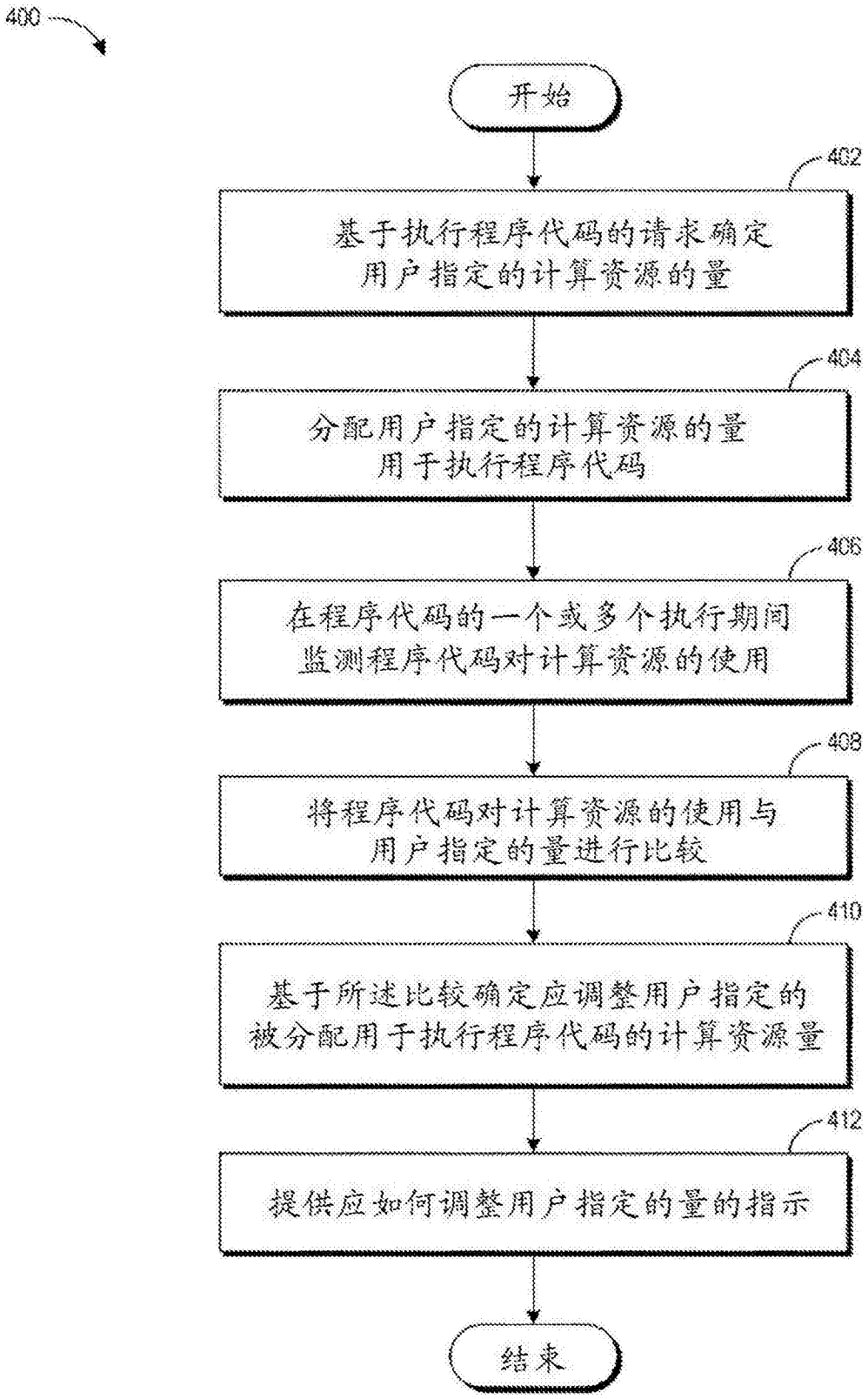

400
开始
402
基于执行程序代码的请求确定
用户指定的计算资源的量
404
分配用户指定的计算资源的量
用于执行程序代码
406
在程序代码的一个或多个执行期间
监测程序代码对计算资源的使用
408
将程序代码对计算资源的使用与
用户指定的量进行比较
410
基于所述比较确定应调整用户指定的
被分配用于执行程序代码的计算资源量
412
提供应如何调整用户指定的量的指示
结束

图4

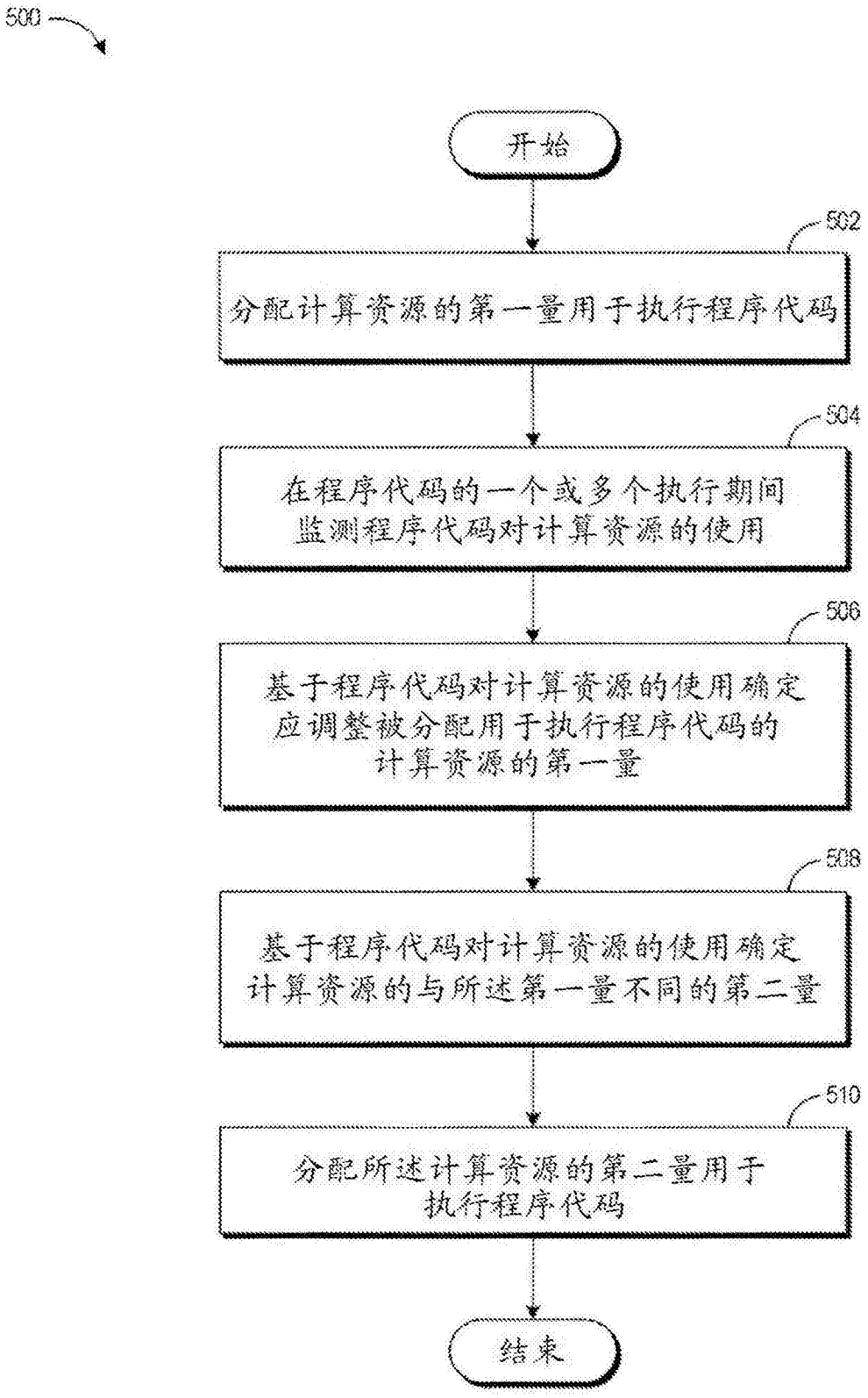

500
开始
502
分配计算资源的第一量用于执行程序代码
504
在程序代码的一个或多个执行期间
监测程序代码对计算资源的使用
506
基于程序代码对计算资源的使用确定
应调整被分配用于执行程序代码的
计算资源的第一量
508
基于程序代码对计算资源的使用确定
计算资源的与所述第一量不同的第二量
510
分配所述计算资源的第二量用于
执行程序代码
结束

图5

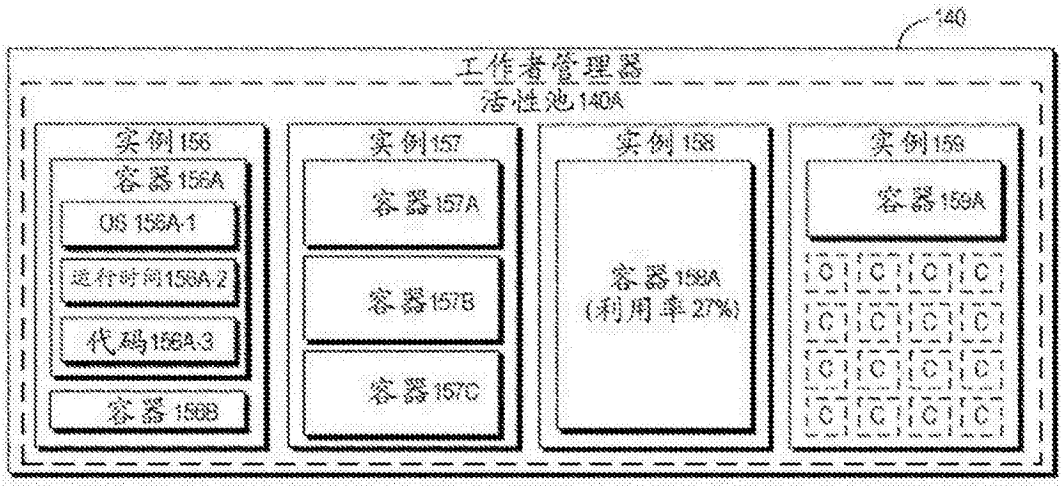

140
工作者管理器
活性池140A
实例156
容器156A
OS 156A-1
运行时间156A-2
代码156A-3
容器156B
实例157
容器157A
容器157B
容器157C
实例158
容器158A
(利用率27%)
实例159
容器159A

图6

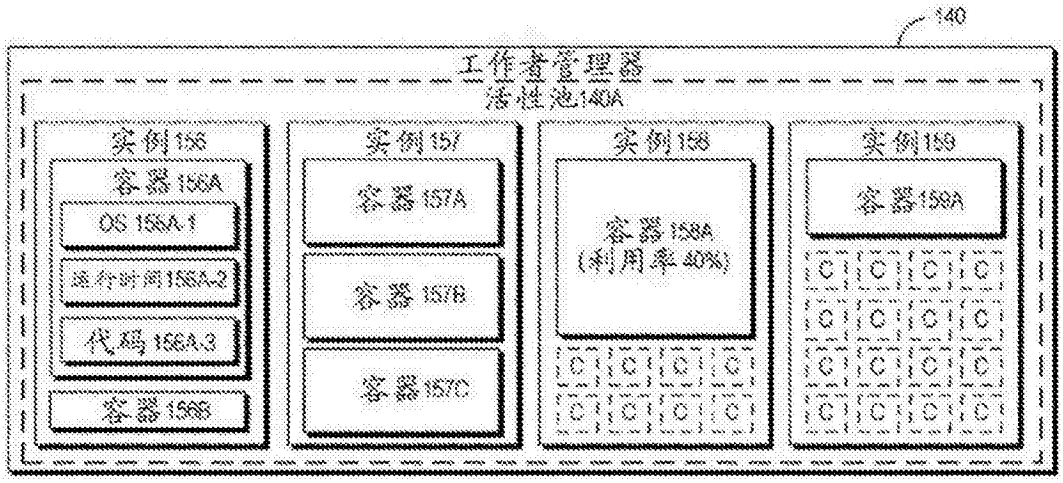

140
工作者管理器
活性池140A
实例156
容器156A
OS 156A-1
运行时间156A-2
代码156A-3
容器156B
实例157
容器157A
容器157B
容器157C
实例158
容器158A
(利用率40%)
实例159
容器159A

图7

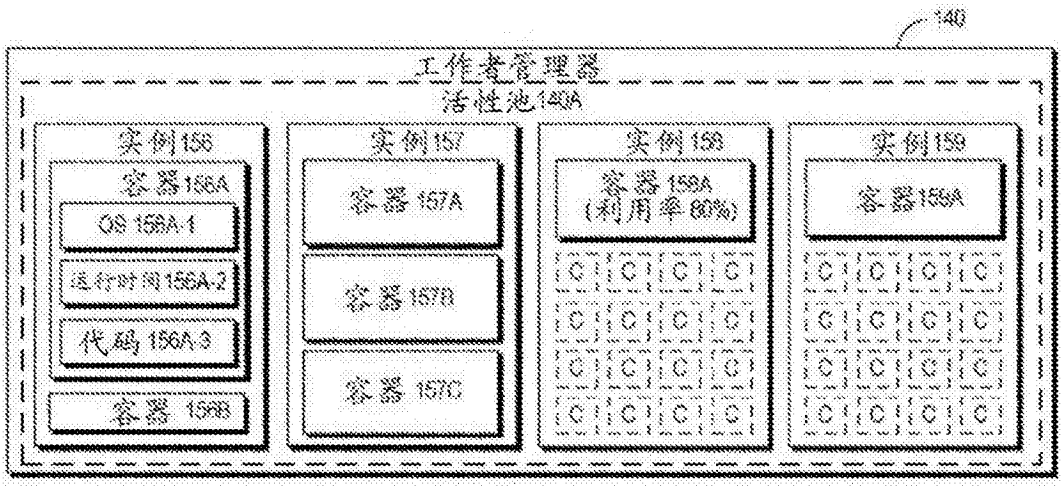

140
工作者管理器
活性池140A
实例156
容器156A
OS 156A-1
运行时间156A-2
代码156A-3
容器156B
实例157
容器157A
容器157B
容器157C
实例158
容器158A
(利用率80%)
实例159
容器159A

图8

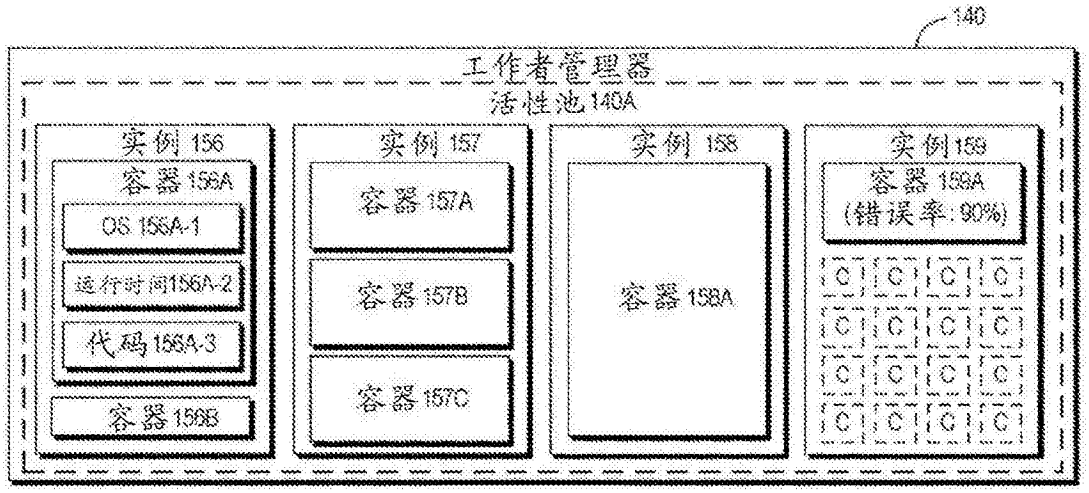

140
工作者管理器
活性池140A
实例 156
容器 156A
OS 156A-1
运行时间156A-2
代码156A-3
容器 156B
实例 157
容器 157A
容器 157B
容器 157C
实例 158
容器 158A
实例159
容器 159A
(错误率:90%)
C

图9

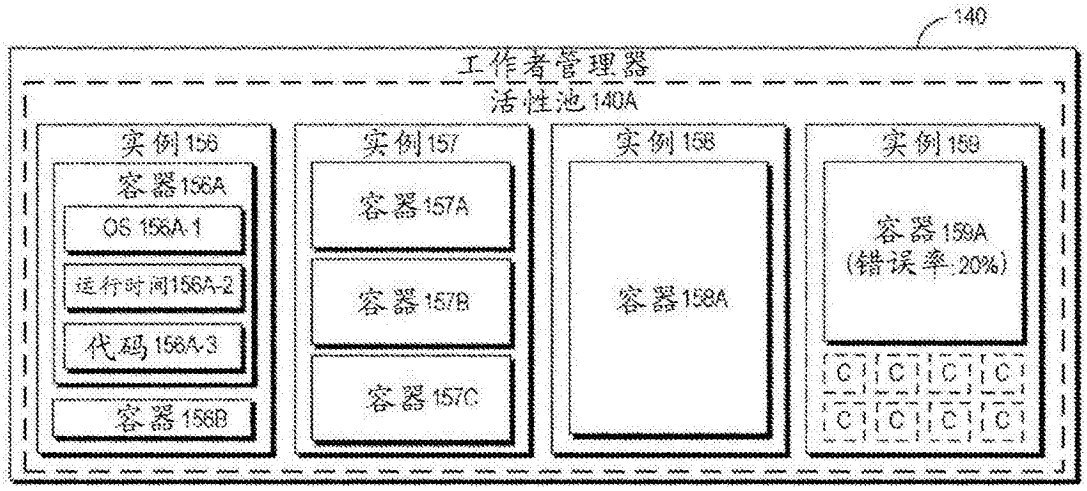

140
工作者管理器
活性池140A
实例 156
容器 156A
OS 156A-1
运行时间156A-2
代码156A-3
容器 156B
实例157
容器 157A
容器 157B
容器 157C
实例 158
容器 158A
实例 159
容器 159A
(错误率:20%)
C

图10

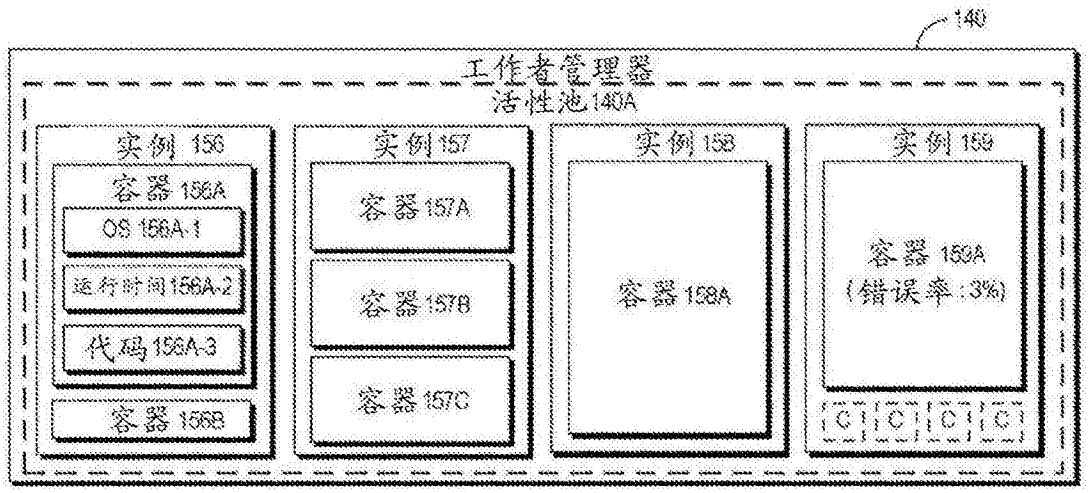

140
工作者管理器
活性池140A
实例 156
容器 156A
OS 156A-1
运行时间156A-2
代码156A-3
容器 156B
实例157
容器 157A
容器 157B
容器 157C
实例158
容器 158A
实例159
容器 159A
(错误率:3%)
C
C
C
C

图11