



(12) 发明专利申请

(10) 申请公布号 CN 115769191 A

(43) 申请公布日 2023. 03. 07

(21) 申请号 202180047877.4

(22) 申请日 2021.06.24

(30) 优先权数据

16/921,381 2020.07.06 US

(85) PCT国际申请进入国家阶段日

2023.01.04

(86) PCT国际申请的申请数据

PCT/IB2021/055603 2021.06.24

(87) PCT国际申请的公布数据

W02022/009011 EN 2022.01.13

(71) 申请人 国际商业机器公司

地址 美国纽约阿芒克

(72) 发明人 T·泰伊克 S·阿格拉瓦尔

D·皮特纳 M·施温德

(74) 专利代理机构 北京市金杜律师事务所

11256

专利代理师 鄢迅

(51) Int.Cl.

G06F 9/455 (2006.01)

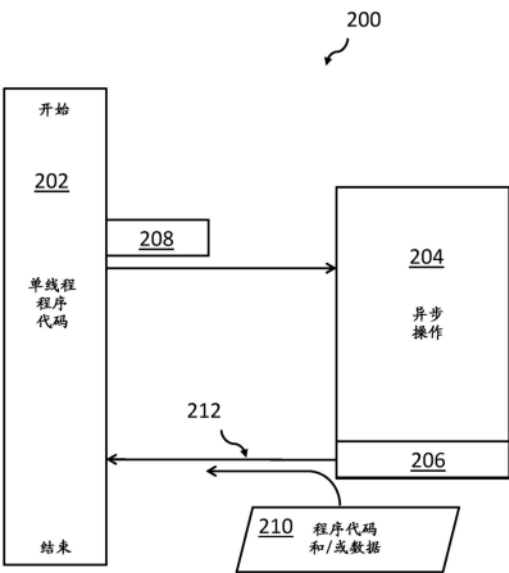
权利要求书3页 说明书13页 附图6页

(54) 发明名称

在云计算环境中管理异步操作

(57) 摘要

处理器可以执行程序代码的异步操作,将与异步操作相关的进程休眠,以及释放除了相关系统存储器之外的相关云运行时平台。另外,该处理器可以在进程的休眠期间执行异步操作,在异步操作完成之后拦截对进程发起的完成函数,将附加程序代码和数据中的至少一项注入到完成函数中,将进程的取消休眠并且重新分配进程的被释放的云运行时平台相关资源,并且执行完成函数,向进程返回异步操作的结果数据以及附加程序代码和数据中的至少一项。



1. 一种用于在无服务器云运行时平台上执行单线程的基于异步I/O的程序代码的计算机实现的方法,所述方法包括:

执行所述程序代码的异步操作,其中所述异步操作将利用完成函数而被完成;
将与所述异步操作相关的进程休眠;
释放所述进程的除了相关系统存储器之外的云运行时平台相关资源;
在将所述进程休眠的同时执行所述异步操作;
在所述异步操作的完成之后拦截对所述进程发起的完成函数;
将附加程序代码和数据中的至少一项注入到所述完成函数中;
将所述进程取消休眠,并且重新分配所述进程的被释放的云运行时平台相关资源;以及

执行所述完成函数向所述进程返回所述异步操作的结果数据以及所述附加程序代码和所述数据中的至少一项,其中使得能够在所述无服务器云运行时平台上执行所述单线程的基于异步I/O的程序代码。

2. 根据权利要求1所述的方法,其中将所述进程取消休眠包括:
执行针对用于所述进程的资源的位置。

3. 根据权利要求2所述的方法,其中所述放置例程是所述云运行时平台的资源的地理可用性的函数。

4. 根据权利要求2或3所述的方法,其中所述放置例程还包括:
基于所述异步操作的位置,确定要被取消休眠的所述进程的资源放置。

5. 根据权利要求1至4中的一项所述的方法,还包括:
基于被休眠的所述进程的位置,确定针对所述异步操作所需的资源的资源放置。

6. 根据权利要求1至5中的一项所述的方法,其中所述异步操作由所述云运行时平台实现为服务。

7. 根据权利要求1至6中的一项所述的方法,其中所述异步操作是长期运行的异步输入/输出操作、定时器操作、或者针对输入操作的等待操作。

8. 根据权利要求1至7中的一项所述的方法,所述方法还包括:

由已经发起所述异步操作的所述执行的放置例程基于与所述异步操作相关联的数据分类和相应的参数来确定针对要被取消休眠的所述进程的资源放置。

9. 一种用于在无服务器云运行时平台上执行单线程的基于异步I/O的程序代码的运行系统,所述运行时系统包括:

存储器;
与所述存储器通信的处理器,所述处理器被配置为执行包括以下的操作:
执行所述程序代码的异步操作,其中,所述异步操作将利用完成函数而被完成;
将与所述异步操作相关的进程休眠;
释放所述进程的除了相关系统存储器之外的云运行时平台相关资源;
在将所述进程休眠期间执行所述异步操作;
在所述异步操作的完成之后拦截对所述进程发起的完成函数;
将附加程序代码和数据中的至少一项注入到所述完成函数中;
将所述进程取消休眠,并且重新分配所述进程的被释放的云运行时平台相关资源;以

及

执行所述完成函数向所述进程返回所述异步操作数据的结果以及所述附加程序代码和所述数据中的至少一项,其中使得能够执行单线程的异步I/O。

10. 根据权利要求9所述的运行时系统,其中将所述进程取消休眠包括:

执行针对用于所述进程的资源放置例程。

11. 根据权利要求10所述的运行时系统,其中所述放置例程是所述云运行时平台的资源的地理可用性的函数。

12. 根据权利要求10或11所述的运行时系统,其中所述放置例程还包括:

基于所述异步操作的位置,确定要被取消休眠的所述进程的资源放置。

13. 根据权利要求9至12中的一项所述的运行时系统,其中所述处理器还被配置为执行包括以下的操作:

基于被休眠的所述进程的位置,确定针对所述异步操作所需的资源的资源放置。

14. 根据权利要求9至13中的一项所述的运行时系统,其中所述异步操作由所述云运行时平台实现为服务。

15. 根据权利要求9至14中的一项所述的运行时系统,其中所述异步操作是长期运行的异步输入/输出操作、定时器操作、或者针对输入操作的等待操作。

16. 一种用于在无服务器云运行时平台上执行单线程的基于异步I/O的程序代码的计算机程序产品,所述计算机程序产品包括计算机可读存储介质,所述计算机可读存储介质具有随其体现的程序指令,所述程序指令由一个或多个计算系统或控制器可执行以执行操作,所述操作包括:

执行所述程序代码的异步操作,其中,所述异步操作将利用完成函数而被完成;

将与所述异步操作相关的进程休眠;

释放所述进程的除了相关系统存储器之外的云运行时平台相关资源;

在将所述进程休眠期间执行所述异步操作;

在所述异步操作的完成之后拦截对所述进程发起的完成函数;

将附加程序代码和数据中的至少一项注入到所述完成函数中;

将所述进程取消休眠,并且重新分配所述进程的被释放的云运行时平台相关资源;以

及

执行所述完成函数向所述进程返回所述异步操作数据的结果以及所述附加程序代码和所述数据中的至少一项,其中使得能够在所述无服务器云运行时平台上执行所述单线程的基于异步I/O的程序代码。

17. 根据权利要求16所述的计算机程序产品,其中将所述进程取消休眠包括:

执行针对用于所述进程的资源放置例程。

18. 根据权利要求17所述的计算机程序产品,其中所述放置例程是所述云运行时平台的资源的地理可用性的函数。

19. 根据权利要求17或18所述的计算机程序产品,其中所述放置例程还包括:

基于所述异步操作的位置,确定要被取消休眠的所述进程的资源放置。

20. 根据权利要求16至19中的一项所述的计算机程序产品,其中所述控制器还被配置为执行包括以下的操作:

基于被休眠的所述进程的位置,确定针对所述异步操作所需的资源的资源放置。

在云计算环境中管理异步操作

技术领域

[0001] 本公开总体上涉及管理云计算环境中的异步操作,并且更具体地涉及在无服务器云运行时上执行单线程的基于异步I/O的程序代码。

背景技术

[0002] 云计算系统、环境和框架在企业计算中经常由于其在资源和资本费用方面的灵活性而变成主流。此外,大量编程和操作模型已经被开发并且经常使用开源变得可用,以探索面向服务的架构、虚拟机、无状态操作的能力和在没有服务器(例如,无服务器)的情况下操作服务的选项。

[0003] 在多租户云环境中,不同的工作负荷可以在任何时间竞争相关资源。云工作量可以容易地扩展,因为执行工作负载的服务按定义是无状态的。取决于推送至云计算环境的工作负载,云提供方可能不得不扩展服务实例并调度诸如虚拟机(“VM”)的资源。传入工作负载的典型分发机制可以是选择具有最小当前工作负载的目标资源(例如,底层硬件系统)。

[0004] 无服务器、基于云的计算是执行模型,其中云提供方提供物理或虚拟计算资源。此类场景中的资源分配还可以取决于所选择的执行运行时程序,其可以是单线程或多线程类型。客户端可以订阅和使用无服务器计算环境的资源,而云提供方可管理底层基础设施的分配。

[0005] 虽然对于某些执行要求是有利的,但基于云的计算也可能具有一些缺点。一个缺点是可能需要不同的编解码概念,例如,程序代码可能必须适应无服务器框架原语的概念以在这样的环境中可运行。一个示例可以是将代码分解成表示单个执行循环的切片并且将这些切片与其他函数链接在一起。通常,现有程序不能使用无服务器计算模型来不经修改地运行。

[0006] 美国专利申请公开号2019/0339955A1描述了用于为位于第一服务上的程序代码函数的实例创建函数检查点并且使用函数检查点将程序代码功能的实例加载在第二服务器上的技术。

[0007] 此外,美国专利申请公开号2019/0384655A1描述了一种用于在无服务器计算环境中处理数据的方法和相关系统。在一个实施例中,从数据源接收与多个无服务器计算平台相关联的函数相关的信息和待处理的数据。此后,基于数据和与数据相关的域的特性来确定处理数据所需的处理操作。基于评估置信源来标识用于执行处理操作的目标函数,并且通过向与该目标函数相关联的无服务器计算平台转发数据来部署该目标函数。在一个实施例中,该要求可以通过动态地选择用于处理数据的最佳无服务器计算平台来消除创建和维护服务器的需要。

[0008] 然而,仍然需要一种允许在无服务器计算框架中不经修改地使用以单线程的基于异步I/O的语言编写的程序代码的技术。

发明内容

[0009] 根据本公开的一些方面,可以提供用于在无服务器云运行时平台上执行单线程的基于异步I/O的程序代码的计算机实现的方法。该方法可以包括执行程序代码的异步操作,其中异步操作将利用完成函数而被完成。该方法可以包括:将与异步操作相关的进程休眠,以及释放进程的除了相关系统存储器之外的云运行时平台相关资源。该方法还可以包括:在进程的休眠期间执行异步操作,在异步操作完成之后拦截对该进程发起的完成函数,以及将附加程序代码和数据中的至少一项注入到完成函数中。

[0010] 此外,方法可以包括:将进程取消休眠,并且重新分配进程的被释放的云运行时平台相关资源;以及执行完成函数,向进程返回异步操作的结果数据以及附加的程序代码和数据中的至少一项,由此使得能够在无服务器云运行时平台上执行单线程的基于异步I/O的程序代码。

[0011] 根据本公开的另一方面,可以提供用于在无服务器云运行时平台上执行单线程的基于异步I/O的程序代码的运行时系统。该系统可以包括存储器和与存储器通信的处理器。处理器可以被配置为执行包括以下的操作:执行程序代码的异步操作,其中,异步操作将利用完成功能被完成;将与异步操作相关的进程休眠;以及释放进程的除了相关系统存储器之外的云运行时平台相关资源。

[0012] 在一些实施例中,处理器可以被配置为执行包括以下的操作:在进程的休眠期间执行异步操作,在异步操作完成之后拦截对进程发起的完成函数,以及将附加程序代码和数据中的至少一项注入到完成函数中。

[0013] 在一些实施例中,处理器可以被配置为执行包括以下的操作:将进程取消休眠,并且重新分配与进程的被释放的云运行时平台相关资源。在一些实施例中,处理器可以被配置为执行包括以下的操作:执行完成函数,向进程返回异步操作的结果数据以及附加程序代码和数据中的至少一项,由此使得能够在无服务器云运行时平台上执行单线程的基于异步I/O的程序代码。

[0014] 所提出的用于在无服务器云运行时平台上执行单线程的基于异步I/O的程序代码的计算机实现方法可以提供多个优点、贡献和技术效果。

[0015] 作为示例,任何类型的单线程程序代码(例如,Node.JS程序)可以通过透明地将执行切片映射到无服务器调用来自动地映射到无服务器计算模型。所以,本发明可以允许未经修改的JavaScript程序在该程序最初未针对其被设计的无服务器环境中执行。本公开还给更传统的单线程和更多或更少线性设计的程序代码带来无服务器计算的优点。

[0016] 已经消除了针对无服务器计算环境重写和重新设计单线程程序代码的要求,使得从用户的角度来看,程序代码现在可以无限期地运行,而从无服务器计算的角度来看,它仍然是纯粹的无服务器应用。这提供了两个优点:第一,单线程编程语言(诸如Node.JS);以及第二,需要较少资源的无服务器计算的优点。

[0017] 本公开的(多个)方法还可允许将任意数据和程序逻辑注入到原始程序代码中,而无需重新编译或重新构造原始单线程程序。所注入的数据的示例可以包括(但不限于):对当前数据堆和剖面数据进行签名的密码学签名,其帮助程序代码的性能分析,并且进而优化取决于针对特定无服务器运行时和集群的所得加载剖面的放置。所注入的程序代码的示例还可包括(但不限于):向审计日志服务发送程序执行状态。

[0018] 本公开的(多个)方法的另一优点是取决于计算资源的可用性而在另一云群集或区域上恢复中断的单线程程序的能力。

[0019] 在一些实施例中,单线程、异步的基于I/O的程序代码可以用选自Node.JS、JavaScript、TypeScript和Golang的编程语言或相当的编程语言来编写。因此,本公开可以与支持单线程的基于异步I/O的编程样式的几乎任何编程语言一起使用。

[0020] 在一些实施例中,取消休眠(或恢复)可以包括执行针对用于进程的资源的位置例程。因此,可以重新建立发起异步操作的发起程序进程所需的资源,从而支持无服务器计算的概念。

[0021] 在一些实施例中,放置例程可以是云运行时平台的资源的地理可用性的函数。所以,通过选择用于放置例程的有用配置参数,可以影响其中将执行相关服务的地理区域。这使得可以减少由于调用和被调用服务之间的长距离而导致的网络时延。

[0022] 在一些实施例中,放置例程还可以包括基于异步操作的位置确定要被取消休眠的进程资源放置。可以基于地理坐标确定位置,例如,可以“靠近”选择资源放置,以便避免相关服务之间的长距离网络连接。

[0023] 在一些实施例中,该方法还可以包括基于休眠的进程的位置来确定针对异步操作所需的资源的资源放置。地理坐标也可以起到主要作用。这可以保证两个服务(具体地,调用和被调用)可以在同一云计算中心中运行。

[0024] 在一些实施例中,无服务器云运行时平台可以基于以下技术之一: Docker、Containerd、Linux Containers、Podman、Kubernetes、Knative、AWS Lambda、谷歌云功能、Azure功能和阿里巴巴云功能计算。另外,可以使用其他类似的无服务器云运行时平台。所以,所提出的概念可以基本上支持任何无服务器云运行时框架。

[0025] 在一些实施例中,异步操作可以是长期运行的异步输入/输出(I/O)操作(例如,对文件系统、数据库等的调用)、定时器操作、或针对输入操作的等待操作。然而,应注意,这可能不适用于直接在无服务器计算框架中的承诺,而是适用于包括输入/输出的那些承诺。这同样适用于异步等待操作;它们通常被映射到承诺。

[0026] 在一些实施例中,该方法还可以包括由已经发起(例如,触发)异步操作的执行的放置例程基于与异步操作相关联的数据分类和相应的参数来确定要被取消休眠的进程的资源放置。因此,不仅代码本身和潜在元数据,而且与异步操作相关联的数据类型也可以用于控制资源放置的运作。当在单线程的基于异步I/O的程序代码的上下文中使用时,这可允许对无状态计算框架的细粒度管理。

[0027] 此外,实施例可以采取从计算机可用或计算机可读介质可访问的相关计算机程序产品的形式,该计算机可用或计算机可读介质提供用于由计算机或任何指令执行系统使用或与其结合使用的程序代码。为了本描述的目的,计算机可用或计算机可读介质可以是包含用于存储、通信、传播或传输程序以供指令执行系统、装置或设备使用或与其结合使用的部件的任何装置。

[0028] 以上概述并不旨在描述本公开的每个所示实施例或每个实现方式。

附图说明

[0029] 包括在本公开中的附图被结合到说明书中并且形成说明书的一部分。它们示出了

本公开的实施例,并且与描述一起用于解释本公开的原理。附图仅说明某些实施例,而并不限制本公开。

[0030] 图1是根据本公开的方面的用于在无服务器云运行时平台上执行单线程的基于异步I/O的程序代码的计算机实现的方法的框图。

[0031] 图2是根据本公开的方面的用于在无服务器云运行时平台上执行单线程的基于异步I/O的程序代码的计算机实现的方法的组件的框图。

[0032] 图3是根据本公开的方面的用于在无服务器云运行时平台上执行单线程的基于异步I/O的程序代码的运行时系统的框图。

[0033] 图4A示出了根据本公开的方面的云计算环境。

[0034] 图4B示出了根据本公开的方面的抽象模型层。

[0035] 图5示出了根据本公开的方面的可以用于实现本文所描述的方法、工具和模块和任何相关功能中的一项或多项的示例计算机系统的高级框图。

[0036] 虽然在此描述的实施例对不同修改和备选形式可修改,但是其细节已经通过举例在附图中示出并且将被详细描述。然而,应当理解,所描述的特定实施例不应被视为限制性的。相反,本发明旨在覆盖落入本公开的精神和范围内的所有修改、等同物和替代物。

具体实施方式

[0037] 本公开的各方面总体涉及管理云计算环境中的异步操作,并且更具体地涉及在无服务器云运行时上执行单线程的基于异步I/O的程序代码。虽然本公开不必限于这样的应用,但是可以通过使用该上下文对不同示例的讨论来理解本公开的各个方面。

[0038] 在本说明书的上下文中,可以使用以下惯例、术语和/或表达:

[0039] 术语“单线程的基于异步I/O的程序代码”可以用于指已经在允许部分功能的有状态执行的不同编程概念下写入的程序代码。即,可以为相关的单线程的基于异步I/O的程序代码底层的完整进程维护变量的状态。这对于使用管理程序上的完整虚拟机的编程模型可以是典型的。

[0040] 术语“无服务器云运行时平台”可以用于指云计算执行模型,其中云提供方运行服务器并且动态地管理分配机器资源。该云计算执行模型可以不需要维护完整的虚拟机,并且无服务器代码有时可以与使用微服务部署的代码结合使用。然而,通常,应用的程序代码可以被写入到纯粹无服务器范例,并且可以根本不依赖于完整供应的服务器。

[0041] 术语“异步操作”可以用于指(具体地,异步I/O)输入/输出处理的形式,其可以允许其他处理在传输或事务完成之前继续。这种形式的输入/输出处理可以允许智能操作系统控制的程序相关活动的部分并行性。异步操作可以有助于将通常缓慢的I/O设备(诸如磁盘或类似设备)的时间用于不依赖于I/O数据的程序代码。

[0042] 术语“程序代码”可以用于指在计算机系统、虚拟计算机系统或无服务器计算环境中执行的数字代码。程序代码可以是以人类可理解的语言编写的程序的结果。程序可被编译、解译或翻译成中间代码并且以作为程序代码执行。

[0043] 术语“完成函数”可以用于指在与所调用的异步操作相关的程序代码结束时发起的系统调用,如I/O服务或I/O调用。

[0044] 术语“将进程休眠”(有时也称为挂起至盘)可以用于指使计算机或计算机的相关

部分子系统掉电,同时保持其状态。在休眠时,计算机或相关资源可以将它们的状态保存到非易失性存储装置中。当接收或取消休眠时,计算机如在进入休眠之前一样工作。还可以注意到,与传统的休眠进程相比,在本文所公开的概念中,随机存取存储器(例如,与实际进程相关的系统存储器)的内容可以不被挂起到盘。

[0045] 术语“注入附加程序代码和数据中的至少一项”可以意味着不仅异步操作的结果可以被返回给调用例程,而且附加数据,特别是真实数据形式的数据或程序代码形式的数据可以被返回给调用例程。

[0046] 术语“取消休眠”可以用于指恢复具有与休眠之前相同的状态信息的计算机操作。

[0047] 术语“Node.JS”可以用于指在Web浏览器外部执行JavaScript代码的已知的开源、过程和平台JavaScript运行时环境。Node.JS让开发者使用JavaScript来编写命令行工具并且将其用于服务器侧脚本,例如,在将页面发送给用户的Web浏览器之前在服务器侧运行脚本以产生动态网页内容。因此,Node.JS可表示统一围绕单个编程语言的Web应用开发的“到处JavaScript”范例,而不是用于服务器和客户端侧脚本的不同语言。

[0048] 术语“JavaScript”(典型地缩写为“JS”)可以用于指符合ECMA脚本规范的编程语言。它是高级的、通常即时编译的编程语言。

[0049] 术语“TypeScript”可以用于指由微软开发和维护的开源编程语言。该术语表示JavaScript的超集,并且将可选的静态类型添加到编程语言。

[0050] 术语“Golang”(或简称为“Go”)可以用于指由谷歌设计和维护的已知静态类型的编译编程语言。该编程语言类似于C,但具有存储器安全性、垃圾收集、结构类型和CSP样式并发性。

[0051] 术语“放置例程”可以用于指潜在地利用配置参数开始异步操作的系统调用。

[0052] 术语“Docker”、“Containerd”、“LXC”、“Podman Kubernetes”、“Knactive”、“AWS Lambda”、“谷歌云功能”、“Azure功能”和“阿里巴巴云功能计算”可以用于指用于无服务器计算的框架。这些无服务器计算框架的版本可以用于所提出的概念的上下文中。上述无服务器计算框架的名称中的大多数是基于“容器化的”服务执行。

[0053] 在此上下文中,术语“云计算”和等价地术语“云服务环境”、“云计算环境”或“云计算框架”可以被解释为用于实现对可配置计算资源(例如,网络、网络带宽、服务器、处理、存储器、存储、应用、和服务)的共享池的方便、按需的网络访问,该可配置计算资源可以用最小的管理工作量或与服务提供方的交互来快速供应和释放。该云模型促进可用性并且可以包括至少五个特性、三个服务模型和四个部署模型。

[0054] 特性如下:

[0055] 按需自助服务:云消耗方可按需自动地单方面供应计算能力,诸如服务器时间和网络存储,而无需与服务的提供方的人类交互。

[0056] 广泛的网络接入:能力通过网络可用并且通过标准机制被访问,该标准机制促进由异构的瘦或厚客户端平台(例如,移动电话、膝上型计算机和PDA)使用。

[0057] 资源池化:提供方的计算资源被池化以使用多租户模型服务于多个消耗方,其中不同的物理和虚拟资源根据需要被动态地指派和重新指派。存在位置独立性的意义,因为消耗方通常对所提供资源的确切位置不具有控制或知识,但是可以能够在较高抽象层(例如,国家、州或数据中心)处指定位置。

[0058] 快速弹性:可以快速且弹性地、在一些情况下自动地供应能力以快速缩小、并且快速释放以快速放大。对于消耗方,可用于供应的能力通常显得不受限制,并且可以在任何时间以任何数量购买。

[0059] 测量的服务:云系统通过利用适于服务类型(例如,存储、处理、带宽和活动用户账户)的某个抽象层的计量能力来自动控制和优化资源使用。可以监视、控制和报告资源使用,从而为所利用的服务的提供方和消耗方两者提供透明度。

[0060] 服务模型如下:

[0061] 软件即服务(SaaS):提供给消耗方的能力是使用在云基础设施上运行的提供方的应用。应用通过诸如web浏览器(例如,基于web的电子邮件)的瘦客户端接口从不同客户端设备可访问。消耗方不管理或控制包括网络、服务器、操作系统、存储装置或甚至个体应用能力的底层云基础设施,可能的例外是有限的用户特定的应用配置设置。

[0062] 平台即服务(PaaS):提供给消耗方的能力是将消耗方创建或获取的应用部署到云基础设施上,该应用是使用提供方所支持的编程语言和工具来创建的。消耗方不管理或控制包括网络、服务器、操作系统或存储装置的底层云基础设施,但是具有对所部署的应用以及可能的应用托管环境配置的控制。

[0063] 基础设施即服务(IaaS):提供给消耗方的能力是供应消耗方能够部署和运行可以包括操作系统和应用的任意软件的处理、存储、网络和其他基本计算资源。消耗方不管理或控制底层云基础设施,但是具有对操作系统、存储装置、所部署的应用的控制,以及可能地对选择联网组件(例如,主机防火墙)的有限的控制。

[0064] 部署模型如下:

[0065] 私有云:云基础设施仅为组织被操作。它可以由组织或第三方管理,并且可以存在于场所内或场所外。

[0066] 社区云:云基础设施由若干组织共享并且支持具有共享的关注(例如,任务、安全性要求、策略和合规性考虑)的特定社区。它可由组织或第三方管理,并且可以存在于驻地或驻地外。

[0067] 公共云:使云基础设施对公众或大型产业组可用并且由销售云服务的组织拥有。

[0068] 混合云:云基础架构是两个或更多个云(私有、社区或公共的)的组合,这些云保持独特的实体,但是通过标准化或专有技术被绑定在一起,这些技术实现数据和应用可移植性(例如,用于云之间的负载平衡的云爆发)。

[0069] 注意,云软件通过面向服务以专注于无状态性(具有例外)、低耦合、模块性和语义互操作性来充分利用云范例。

[0070] 根据本公开的一些方面,可以提供用于在无服务器云运行时平台上执行单线程的基于异步I/O的程序代码的计算机实现的方法。该方法可以包括执行程序代码的异步操作,其中异步操作将利用完成函数而被完成。该方法可以包括:将与异步操作相关的进程休眠,以及释放进程的除了相关系统存储器之外的云运行时平台相关资源。该方法还可以包括:在进程的休眠期间执行异步操作,在异步操作完成之后拦截对该进程发起的完成函数,以及将附加程序代码和数据中的至少一项注入到完成函数中。

[0071] 此外,方法可以包括:将进程取消休眠,并且重新分配进程的被释放的云运行时平台相关资源;以及执行完成函数,向进程返回异步操作的结果数据以及附加的程序代码和

数据中的至少一项,由此使得能够在无服务器云运行时平台上执行单线程的基于异步I/O的程序代码。

[0072] 根据本公开的另一方面,可以提供用于在无服务器云运行时平台上执行单线程的基于异步I/O的程序代码的运行时系统。该系统可以包括存储器和与存储器通信的处理器。处理器可以被配置为执行包括以下的操作:执行程序代码的异步操作,其中,异步操作将利用完功能被完成;将与异步操作相关的进程休眠;以及释放进程的除了相关系统存储器之外的云运行时平台相关资源。

[0073] 在一些实施例中,处理器可以被配置为执行包括以下的操作:在进程的休眠期间执行异步操作,在异步操作完成之后拦截对进程发起的完成函数,以及将附加程序代码和数据中的至少一项注入到完成函数中。

[0074] 在一些实施例中,处理器可以被配置为执行包括以下的操作:将进程取消休眠,并且重新分配与进程的被释放的云运行时平台相关资源。在一些实施例中,处理器可以被配置为执行包括以下的操作:执行完成函数,向进程返回异步操作的结果数据以及附加程序代码和数据中的至少一项,由此使得能够在无服务器云运行时平台上执行单线程的基于异步I/O的程序代码。

[0075] 所提出的用于在无服务器云运行时平台上执行单线程的基于异步I/O的程序代码的计算机实现方法可以提供多个优点、贡献和技术效果。

[0076] 作为示例,任何类型的单线程程序代码(例如,Node.JS程序)可以通过透明地将执行切片映射到无服务器调用来自动地映射到无服务器计算模型。所以,本发明可以允许未经修改的JavaScript程序在该程序最初未针对其被设计的无服务器环境中执行。本公开还给更传统的单线程和更多或更少线性设计的程序代码带来无服务器计算的优点。

[0077] 已经消除了针对无服务器计算环境重写和重新设计单线程程序代码的要求,使得从用户的角度来看,程序代码现在可以无限期地运行,而从无服务器计算的角度来看,它仍然是纯粹的无服务器应用。这提供了两个优点:第一,单线程编程语言(诸如Node.JS);以及第二,需要较少资源的无服务器计算的优点。

[0078] 本公开的(多个)方法还可以允许将任意数据和程序逻辑注入到原始程序代码中,而无需重新编译或重新构造原始单线程程序。所注入的数据的示例可以包括(但不限于):对当前数据堆和剖面数据进行签名的密码学签名,其帮助程序代码的性能分析,并且进而优化取决于针对特定无服务器运行时和集群的所得加载剖面的放置。所注入的程序代码的示例还可包括(但不限于):向审计日志服务发送程序执行状态。

[0079] 本公开的(多个)方法的另一优点是取决于计算资源的可用性而在另一云群集或区域上恢复中断的单线程程序的能力。

[0080] 在一些实施例中,单线程、异步的基于I/O的程序代码可以用选自Node.JS、JavaScript、TypeScript和Golang的编程语言或相当的编程语言来编写。因此,本公开可以与支持单线程的基于异步I/O的编程样式的几乎任何编程语言一起使用。

[0081] 在一些实施例中,取消休眠(或恢复)可以包括执行针对用于进程的资源放置例程。因此,可以重新建立发起异步操作的发起程序进程所需的资源,从而支持无服务器计算的概念。

[0082] 在一些实施例中,放置例程可以是云运行时平台的资源的地理可用性的函数。所

以,通过选择用于放置例程的有用配置参数,可以影响其中将执行相关服务的地理区域。这使得可以减少由于调用和被调用服务之间的长距离而导致的网络时延。

[0083] 在一些实施例中,放置例程还可以包括基于异步操作的位置确定要被取消休眠的进程资源放置。可以基于地理坐标确定位置,例如,可以“靠近”选择资源放置,以便避免相关服务之间的长距离网络连接。

[0084] 在一些实施例中,该方法还可以包括基于休眠的进程的位置来确定针对异步操作所需的资源的资源放置。地理坐标也可以起到主要作用。这可以保证两个服务(具体地,调用和被调用)可以在同一云计算中心中运行。

[0085] 在一些实施例中,无服务器云运行时平台可以基于以下技术之一: Docker、Containerd、Linux Containers、Podman、Kubernetes、Knative、AWS Lambda、谷歌云功能、Azure功能和阿里巴巴云函数计算。另外,可以使用其他类似的无服务器云运行时平台。所以,所提出的概念可以基本上支持任何无服务器云运行时框架。

[0086] 在一些实施例中,异步操作可以是长期运行的异步输入/输出(I/O)操作(例如,对文件系统、数据库等的调用)、定时器操作、或针对输入操作的等待操作。然而,应注意,这可能不适用于直接在无服务器计算框架中的承诺,而是适用于包括输入/输出的那些承诺。这同样适用于异步等待操作;它们通常被映射到承诺。

[0087] 在一些实施例中,该方法还可以包括由已经发起(例如,触发)异步操作的执行的放置例程基于与异步操作相关联的数据分类和相应的参数来确定针对要被取消休眠的进程的资源放置。因此,不仅代码本身和潜在元数据,而且与异步操作相关联的数据类型也可以用于控制资源放置的运作。当在单线程的基于异步I/O的程序代码的上下文中使用时,这可允许对无状态计算框架的细粒度管理。

[0088] 此外,实施例可以采取从计算机可用或计算机可读介质可访问的相关计算机程序产品的形式,该计算机可用或计算机可读介质提供用于由计算机或任何指令执行系统使用或与其结合使用的程序代码。为了本描述的目的,计算机可用或计算机可读介质可以是包含用于存储、通信、传播或传输程序以供指令执行系统、装置或设备使用或与其结合使用的装置的任何装置。

[0089] 在下文中,将给出附图的详细描述。附图中的所有指令是示意性的。首先,给出了用于在无服务器云运行时平台上执行单线程的基于异步I/O的程序代码的计算机实现的方法的实施例的框图。之后,将描述进一步的实施例、以及用于在无服务器云运行时平台上执行单线程的基于异步I/O的程序代码的运行时系统的实施例。

[0090] 参考图1,其示出用于在无服务器云计算运行时平台(例如,Docker)上执行单线程的基于异步I/O的程序代码(例如,Node.JS、JavaScript、TypeScript或Golang)的计算机实现方法100的实施例的框图。程序代码可能已经被编译,并且其可以是可解译的代码或基于中间代码的程序代码。

[0091] 在一些实施例中,方法100在操作102开始。在操作102,执行程序代码的异步操作。由此,利用完功能完成异步操作。备选地,回调函数也是可能的。

[0092] 在一些实施例中,方法100进行到操作104。在操作104,与异步操作有关的进程被休眠。在一些实施例中,方法100进行到操作106,其中进程的云运行时平台相关资源(相关系统存储器(例如,RAM)除外)被释放(free-up)或释放(release)。所以,被存储在相关系统

存储器中的数据将保持活跃,并且在可执行代码的情况下将被执行。注意,进程的云运行时平台相关资源的至少一部分可以被释放或释放。不需要自动释放所有资源。

[0093] 在一些实施例中,方法100进行到操作108,其中在进程的休眠期间执行异步操作。异步操作可以是例如潜在地处于不同地理位置的另一群集的另一节点(例如,通常是不同的云计算中心)上的系统调用。

[0094] 在一些实施例中,方法100进行到操作110。在操作110,在异步操作的完成之后对进程的发起的完成函数或完成函数调用被拦截。在一些实施例中,方法100进行到操作112。在操作112处,拦截可于用于将附加程序代码或数据或两者注入到完成函数中,使得其也被返回到调用程序。

[0095] 在一些实施例中,方法100进行到操作114。在操作114处,该进程被取消休眠并且该进程的释放的与云运行时平台相关的资源被重新分配。在一些实施例中,方法100进行到操作116。在操作116,执行将异步操作的结果数据以及附加程序代码和数据中的至少一项返回给进程的完成函数,从而使得能够在无服务器云运行时平台上执行单线程的基于异步I/O的程序代码。这样,有利地,“旧的”单线程程序代码(具体地,用于长事务的程序代码)可以在无状态云计算框架中在没有任何修改的情况下操作。

[0096] 现在参考图2,其示出了所涉及的组件及其之间的交互的实施例200的框图。在左内侧,框202表示具有开始和结束的单线程程序代码。然而,其还可以是循环程序,其仅可以通过用户输入或通过基于操作系统的服务来终止。

[0097] 在程序202的操作期间一借助于放置例程208一开始异步操作,如异步I/O操作204。可以调整几个参数值以控制异步操作的执行。除了系统存储器(例如,相关的RAM)之外的与单线程程序代码202相关的操作系统进程(未示出)可以在异步操作204的执行期间休眠。

[0098] 在异步操作204结束时,可以例如以另一系统调用的形式来执行完成函数206。然而,异步操作204的结果不直接返回给单线程程序代码204以及相应的进程。而是,当尝试将结果数据从另一交叉操作204返回给程序代码202时,完成函数被拦截,并且在返回路径212中注入附加程序代码和/或数据210。

[0099] 这样,在将单线程程序代码202取消休眠之前,或者更好地,将相关操作系统进程取消休眠之前,可以在单线程程序代码继续其执行之前注入附加的函数点。

[0100] 现在参见图3,其示出了用于在无服务器云运行时平台上执行单线程的基于异步I/O的程序代码的运行系统300的实施例的框图。该运行时系统包括第一执行模块302,适于执行程序代码的异步操作。由此,利用完成功能完成异步操作。运行时间系统还包括休眠模块304和释放单元306,休眠模块304用于将与异步操作相关的进程休眠,释放单元306用于释放进程的除了相关系统存储器之外的云运行时平台相关资源。

[0101] 此外,运行时系统300还包括:第二执行模块308,适于在进程的休眠期间执行异步操作;拦截器310,适于在异步操作完成之后拦截对该进程发起的完成函数;以及注入器模块312,适于将附加程序代码和数据中的至少一项注入到完成函数中。

[0102] 运行时系统300还包括取消休眠模块314,用于将进程取消休眠,并且重新分配、释放、或释放进程的云运行时平台相关资源。运行时系统300还包括第三执行模块316,适于执行将异步操作数据的结果、以及附加程序代码和数据中的至少一项返回给进程的完成函

数,由此使得能够执行单线程异步I/O。

[0103] 注意,运行时系统300的单元和模块通信地耦合以用于信号交换。因此,第一执行模块302、休眠模块304、释放单元306、第二执行模块308、拦截器310、注入器模块312、取消休眠模块314和第三执行模块316可以以这样的方式彼此链接。备选地,模块中的这些单元可以连接到运行时系统内部总线系统318。

[0104] 现在参考图4A,其示出了云计算环境410。如图所示,云计算环境410包括由云消耗方使用的本地计算设备可以与其通信的一个或多个云计算节点400,本地计算设备诸如例如个人数字助理(PDA)或蜂窝电话400A、台式计算机400B、膝上型计算机400C和/或汽车计算机系统400N。节点400可以彼此通信。它们可以物理地或虚拟地分组(未示出)在一个或多个网络中,诸如如上所述的私有云、社区云、公共云或混合云,或者其组合。

[0105] 这允许云计算环境410提供基础架构、平台和/或软件作为云消耗方不需要为其维护本地计算设备上的资源的服务。应当理解,图4A中示出的计算设备400A-N的类型仅旨在是说明性的,并且计算节点400和云计算环境410可以通过任何类型的网络和/或网络可寻址连接(例如,使用Web浏览器)与任何类型的计算机化设备通信。

[0106] 图4B示出了由云计算环境410(图4A)提供的一组功能抽象层。应提前理解,图4B中所示的组件、层和功能仅旨在是说明性的,并且本公开的实施例不限于此。如下所述,提供以下层和对应功能:硬件和软件层415包括硬件和软件组件。硬件组件的示例包括:大型机402;服务器406;基于RISC(精简指令集计算机)架构的服务器404;刀片式服务器408;存储设备411;网络和网络组件412。在一些实施例中,软件组件包括网络应用服务器软件414和/或数据库软件416。

[0107] 虚拟化层420提供抽象层,从该抽象层可以提供虚拟实体的以下示例:虚拟服务器422;虚拟存储装置424;虚拟网络426,包括虚拟专用网络;虚拟应用和操作系统428;以及虚拟客户端430。

[0108] 在一个示例中,管理层440可以提供以下描述的功能。资源供应442提供用于在云计算环境内执行任务的计算资源和其他资源的动态采购。计量和定价444在云计算环境内利用资源时提供成本跟踪,并为这些资源的消费开账单或发票。在一个示例中,这些资源可以包括应用软件许可证。安全性为云消耗方和任务提供身份验证,以及为数据和其他资源提供保护。用户门户446为消耗方和系统管理员提供对云计算环境的访问。服务水平管理448提供云计算资源分配和管理,使得满足所需的服务水平。服务水平协议(SLA)规划和履行450提供根据SLA预期未来需求的云计算资源的预安排和采购。

[0109] 工作负载层460提供可以利用云计算环境的功能的示例。可以从该层提供的工作负载和功能的示例包括:地图和导航462;软件开发和生命周期管理464;虚拟课堂教育交付466;数据分析处理468;事务处理470;和运行时系统472(比较图3,运行时系统300)。

[0110] 本公开的实施例可以与几乎任何类型的计算机一起实现,而不管平台适合于存储和/或执行程序代码。

[0111] 现在参照图5,其示出根据本公开的实施例的可以用于实现本文所描述的方法、工具和模块中的一种或多种以及任何相关功能(例如,使用计算机的一个或多个处理器电路或计算机处理器)的另一示例性计算机系统500的高级框图。在一些实施例中,计算机系统500的主要组件可以包括一个或多个CPU 502、存储器子系统504、终端接口512、存储接口

516、I/O (输入/输出) 设备接口514、和网络接口518,所有这些可以直接或间接地通信地耦合,以用于经由存储器总线503、I/O总线508、以及I/O总线接口单元510的组件间通信。

[0112] 计算机系统500可以包含一个或多个通用可编程中央处理单元 (CPU) 502A、502B、502C和502D,这里统称为CPU502。在一些实施例中,计算机系统500可以包含对于相对大的系统典型的多个处理器;然而,在其他一些实施例中,计算机系统500备选地是单个CPU系统。每个CPU 502可以执行被存储在存储器子系统504中的指令,并且可以包括板载高速缓存的一个或多个级。

[0113] 系统存储器504可以包括易失性存储器形式的计算机系统可读介质,诸如随机存取存储器 (RAM) 522或高速缓存存储器524。计算机系统501可以进一步包括其他可移除/不可移除、易失性/非易失性计算机系统存储介质。仅作为示例,存储系统526可以被提供用于从不可移除、非易失性磁介质 (诸如,“硬盘驱动器”) 读取和向其写入。尽管未示出,可以提供用于从可移除非易失性磁盘 (例如,“软盘”) 读取或向其写入的磁盘驱动器,或用于从可移除非易失性光盘 (诸如CD-ROM、DVD-ROM或其他光学介质) 读取或向其写入的光盘驱动器。此外,存储器504可以包括闪存,例如闪存棒驱动器或闪存驱动器。存储器设备可以通过一个或多个数据介质接口连接到存储器总线503。存储器504可以包括具有一组 (例如,至少一个) 程序模块的至少一个程序产品,这些程序模块被配置为执行各种实施例的功能。

[0114] 各自具有至少一组程序模块530的一个或多个程序/实用程序528可以被存储在存储器504中。程序/实用程序528可以包括管理程序 (也称为虚拟机监视器)、一个或多个操作系统、一个或多个应用程序、其他程序模块、以及程序数据。操作系统、一个或多个应用程序、其他程序模块、以及程序数据中的每一项或它们的一些组合可以包括联网环境的实现。程序528和/或程序模块530一般执行各种实施例的功能或方法。

[0115] 尽管存储器总线503在图5中示出为提供CPU 502、存储器子系统504和I/O总线接口510之间的直接通信路径的单个总线结构,但是在一些实施例中,存储器总线503可以包括多个不同的总线或通信路径,其可以以各种形式中的任一种布置,诸如分层、星形或web配置中的点对点链路、多个分级总线、并行和冗余路径、或任何其他适当类型的配置。此外,虽然I/O总线接口510和I/O总线508被示为单个相应单元,但是在一些实施例中,计算机系统500可以包含多个I/O总线接口单元510、多个I/O总线508或两者。此外,虽然示出了将I/O总线508与运行到不同I/O设备的不同通信路径分开的多个I/O接口单元,但是在其他一些实施例中,I/O设备中的一些或全部I/O设备可以直接连接到一个或多个系统I/O总线。

[0116] 在一些实施例中,计算机系统500可以是多用户大型计算机系统、单用户系统、或者服务器计算机或具有很少或没有直接用户界面但从其他计算机系统 (客户端) 接收请求的类似设备。此外,在一些实施例中,计算机系统500可以实现为台式计算机、便携式计算机、膝上型或笔记本计算机、平板计算机、袖珍计算机、电话、智能电话、网络交换机或路由器、或者任何其他合适类型的电子设备。

[0117] 要注意的是,图5旨在描述示例性计算机系统500的代表性主要组件。然而,在一些实施例中,单个组件可以具有比图5中所表示的更大或更小的复杂度,可以存在不同于图5中所示出的那些组件或者除图5中所示出的那些组件之外的组件,并且此类组件的数目、类型和配置可以变化。

[0118] 如在此更详细地讨论的,预期本文描述的方法的一些实施例的一些或全部操作可

以按备选顺序执行或可以根本不执行;此外,多个操作可以同时发生或作为较大过程的内部部分发生。

[0119] 已经出于说明的目的呈现了不同实施例的描述,但并不旨在是详尽的或限于所公开的实施例。在不脱离所描述的实施例的范围和精神的情况下,许多修改和变化对本领域普通技术人员将是明显的。这里使用的术语被选择来最好地解释实施例的原理、实际应用或对在市场中的找到的技术的技术改进,或者使得本领域普通技术人员能够理解这里公开的实施例。

[0120] 本技术方案可以是任何可能的技术细节集成水平的系统、方法和/或计算机程序产品。计算机程序产品可以包括其上具有计算机可读程序指令的一个或多个计算机可读存储介质,该计算机可读程序指令用于使处理器执行本公开的各方面。

[0121] 该介质可以是用于传播介质的电子、磁性、光学、电磁、红外或半导体系统。计算机可读介质的示例可以包括半导体或固态存储器、磁带、可移除计算机磁盘、随机存取存储器(RAM)、只读存储器(ROM)、刚性磁盘和光盘。光盘的当前示例包括致密盘-只读存储器(CD-ROM)、致密盘-读/写(CD-R/W)、DVD和蓝光盘。

[0122] 计算机可读存储介质可以是保留和存储指令以供指令执行设备使用的有形设备。计算机可读存储介质可以是例如但不限于电子存储设备、磁存储设备、光存储设备、电磁存储设备、半导体存储设备或前述各项的任何合适的组合。计算机可读存储介质的更具体示例的非穷举列表包括以下:便携式计算机盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦除可编程只读存储器(EPROM或闪存)、静态随机存取存储器(SRAM)、便携式压缩盘只读存储器(CD-ROM)、数字通用盘(DVD)、记忆棒、软盘、机械编码的设备(诸如穿孔卡片或者具有记录在其上的指令的凹槽中的凸起结构),以及前述各项的任何合适的组合。如本文中所使用的,计算机可读存储介质不应被解释为瞬态信号本身,诸如无线电波或其他自由传播的电磁波、通过波导或其他传输介质传播的电磁波(例如,通过光纤电缆的光脉冲)、或者通过电线传输的电信号。

[0123] 本文所述的计算机可读程序指令可以从计算机可读存储介质被下载到相应的计算/处理设备,或者经由网络(例如,互联网、局域网、广域网和/或无线网络)被下载到外部计算机或者外部存储设备。网络可以包括铜传输电缆、光传输光纤、无线传输、路由器、防火墙、交换机、网关计算机和/或边缘服务器。每个计算/处理设备中的网络适配器卡或网络接口从网络接收计算机可读程序指令,并且转发计算机可读程序指令以存储在相应计算/处理设备内的计算机可读存储介质中。

[0124] 用于执行本公开的操作的计算机可读程序指令可以是汇编指令,指令集架构(ISA)指令、机器指令、机器相关指令、微码、固件指令、状态设置数据、集成电路系统的配置数据,或者以一种或多种编程语言的任何组合编写的源代码或目标代码,包括诸如Java、Smalltalk、C++等面向对象的编程语言,以及过程式编程语言,诸如“C”编程语言或类似的编程语言。计算机可读程序指令可以完全在用户的计算机上执行、部分在用户的计算机上执行、作为独立软件包执行、部分在用户的计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在后一种场景中,远程计算机可以通过包括局域网(LAN)或广域网(WAN)的任何类型的网络连接到用户的计算机,或者可以连接到外部计算机(例如,通过使用互联网服务提供方的互联网)。在一些实施例中,包括例如可编程逻辑电路装置、现

场可编程门阵列 (FPGA) 或可编程逻辑阵列 (PLA) 的电子电路可以通过利用计算机可读程序指令的状态信息来执行计算机可读程序指令以使电子电路装置个性化, 以便执行本公开的方面。

[0125] 本文中参考根据本公开的实施例方法、装置(系统)和计算机程序产品的流程图说明和/或框图描述本公开的方面。应当理解, 流程图和/或框图的每个框以及流程图和/或框图中各个框的组合, 都可以由计算机可读程序指令来实现。

[0126] 这些计算机可读程序指令可以被提供给通用计算机的处理器, 专用计算机或者其他可编程数据处理装置以产生机器, 使得经由计算机的处理器或其他可编程数据处理装置执行的指令创建用于实现在流程图和/或框图的一个或多个框中指定的功能/动作的装置。这些计算机可读程序指令还可以被存储在可以引导计算机、可编程数据处理装置, 和/或以特定方式起作用的其他设备的计算机可读存储介质中, 使得具有存储在其中的指令的计算机可读存储介质包括制品, 该制品包括实现流程图和/或框图中的一个或多个框中指定的功能/动作的各方面的指令。

[0127] 计算机可读程序指令还可以被加载到计算机、其他可编程数据处理装置、或者另一设备上, 以使一系列操作步骤在计算机、其他可编程装置或者其他设备产生计算机实现的过程, 使得在计算机、其他可编程装置或者其他设备上执行的指令实现流程图和/或框图中的一个或多个框中指定的功能/动作。

[0128] 图中的流程图和/或框图示出了根据本公开的不同实施例的系统、方法和计算机程序产品的可能的实现的架构、功能和操作。对此, 流程图或框图中的每个框可以表示指令的模块、段或部分, 其包括用于实现(多个)指定的逻辑功能的一个或多个可执行指令。在一些备选实现中, 框中标注的功能可以不按照图中标注的顺序发生。例如, 连续示出的两个方框实际上可以作为一个步骤完成, 同时、基本上同时、以部分或完全时间上重叠的方式执行, 或者框有时可以以相反的顺序执行, 这取决于所涉及的功能。也要注意的, 框图和/或流程图中的每个框、以及框图和/或流程图中的框的组合, 可以用执行指定的功能或动作或执行专用硬件与计算机指令的组合的专用的基于硬件的系统来实现。本文中使用的术语仅用于描述具体实施例的目的, 而并非旨在限制本公开。如本文中使用的, 除非上下文另有明确指示, 否则单数形式“一”、“一个”和“该”旨在也包括复数形式。还应当理解, 当在本说明书中使用术语“包括(comprises)”和/或“包含(comprising)”时, 其指定所述特征、整体、步骤、操作、元件和/或组件的存在, 但不排除一个或多个其他特征、整体、步骤、操作、元件、组件和/或其组合的存在或添加。所附权利要求中的所有装置或步骤加上功能元件的对应结构、材料、动作和等效物旨在包括用于结合如具体要求保护的其他要求保护的元件执行该功能的任何结构、材料或动作。已经出于说明和描述的目的呈现了对本公开的描述, 但并不旨在是详尽的或限于所公开形式的本公开。在不背离本公开的范围和精神的情况下, 许多修改和变化对于本领域普通技术人员来说是明显的。选择和描述这些实施例是为了最好地解释本公开的原理和实际应用, 并且使本领域的其他普通技术人员能够理解具有不同修改的不同实施例(如适合于所预期的特定用途)的本公开。

[0129] 虽然已经根据具体实施例描述了本公开, 但预期其变更和修改将对本领域技术人员变得清楚。因此, 以下权利要求旨在被解释为覆盖落入本公开的真实精神和范围内的所有这样的改变和修改。

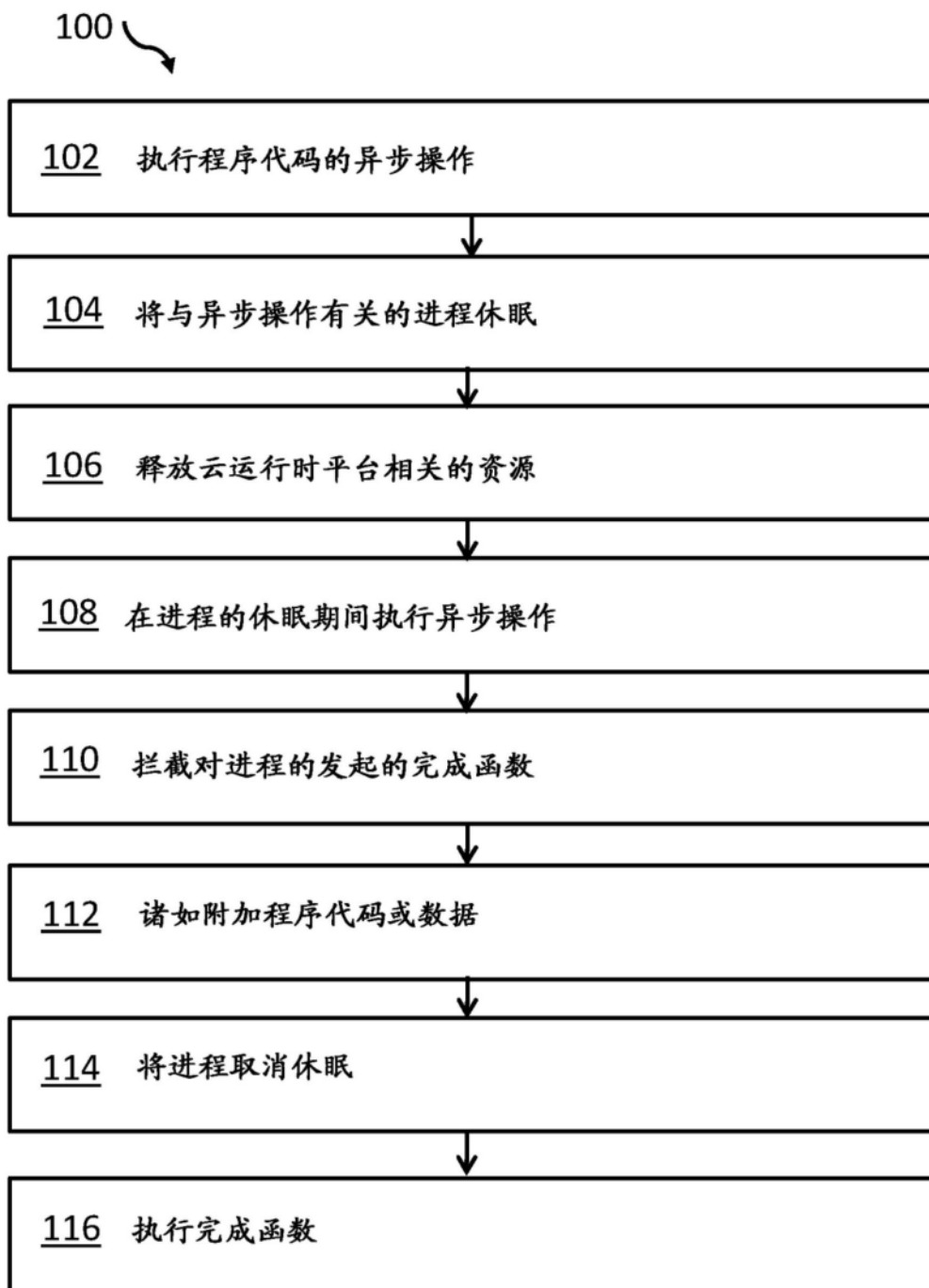


图1

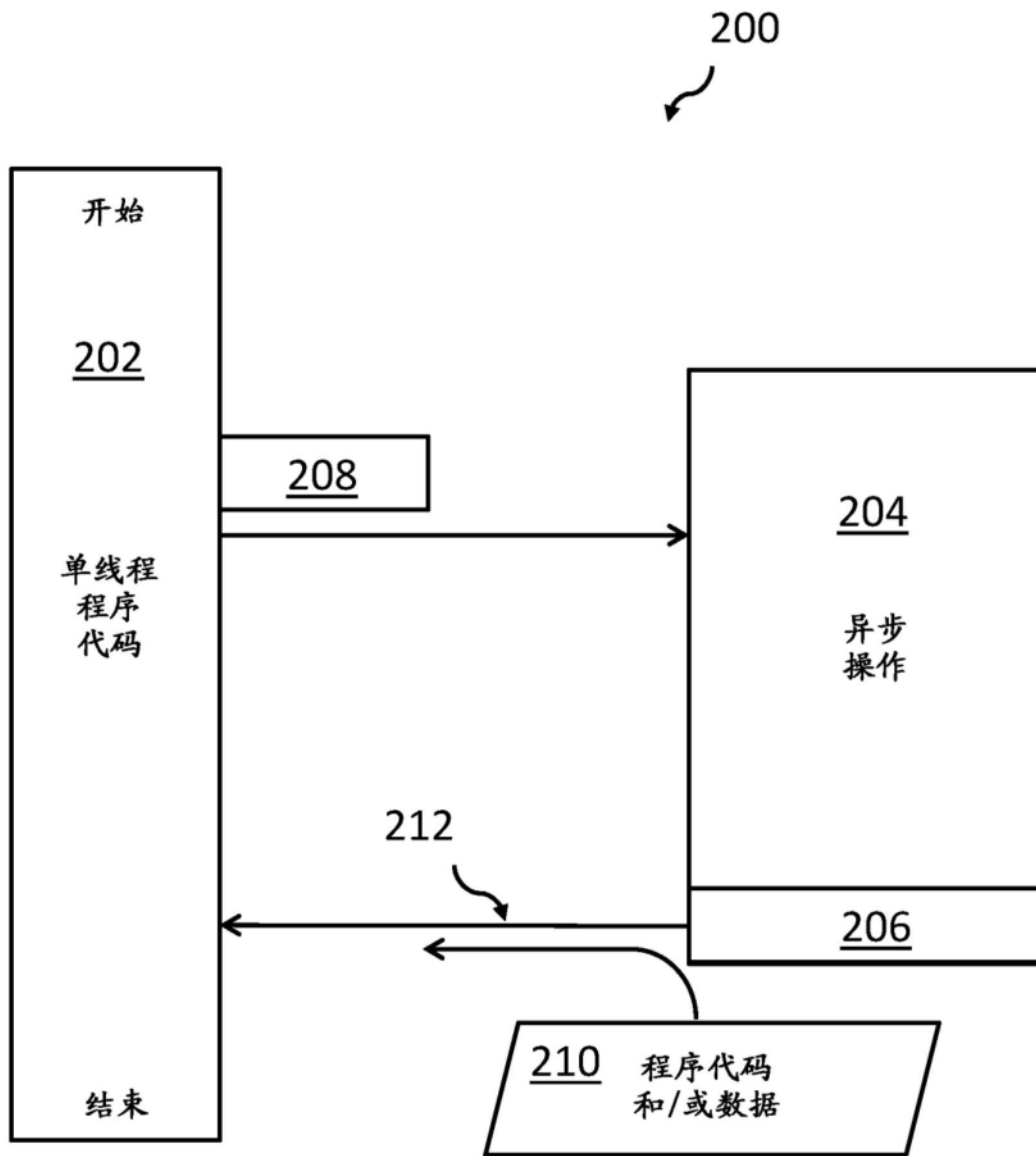


图2

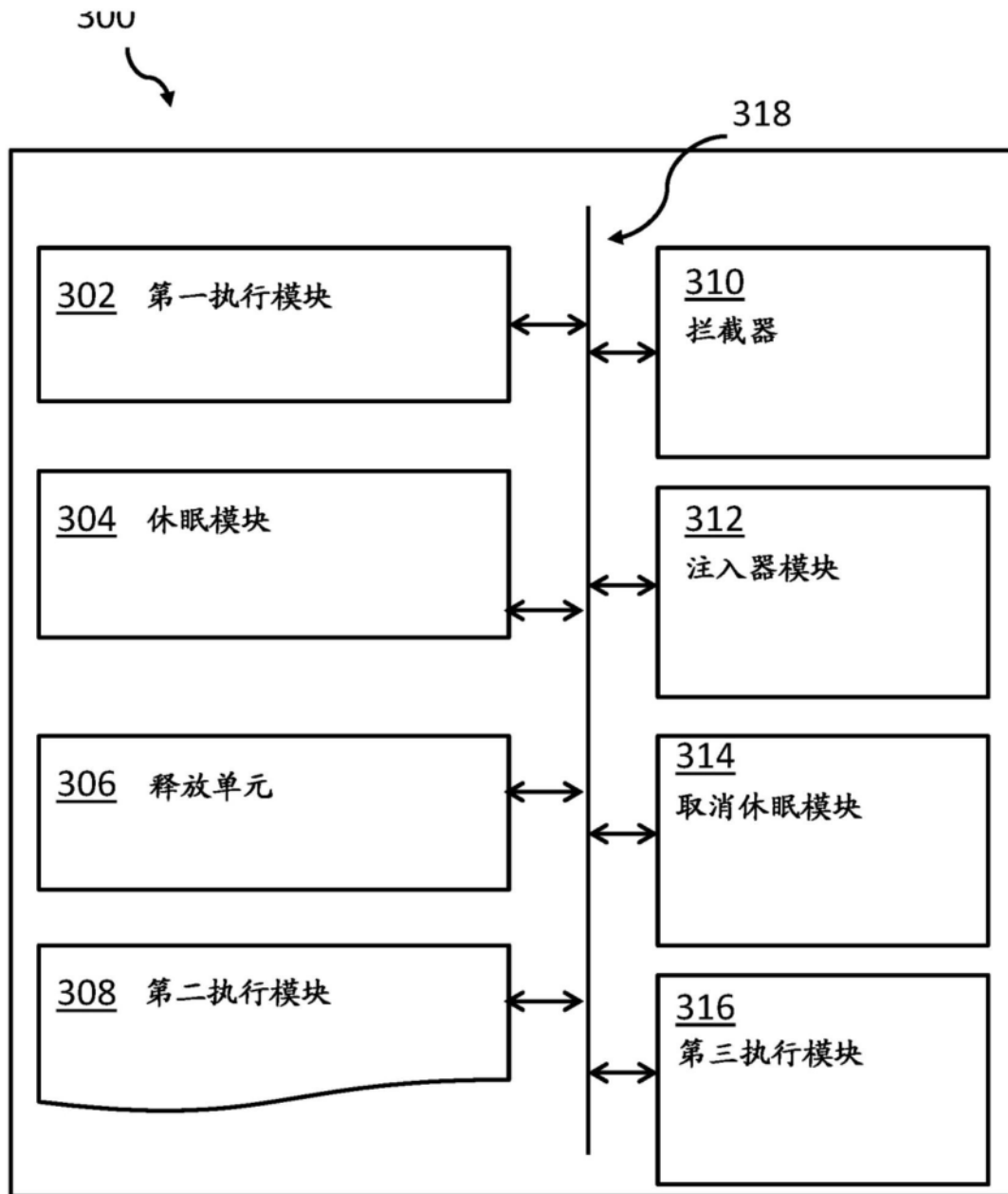


图3

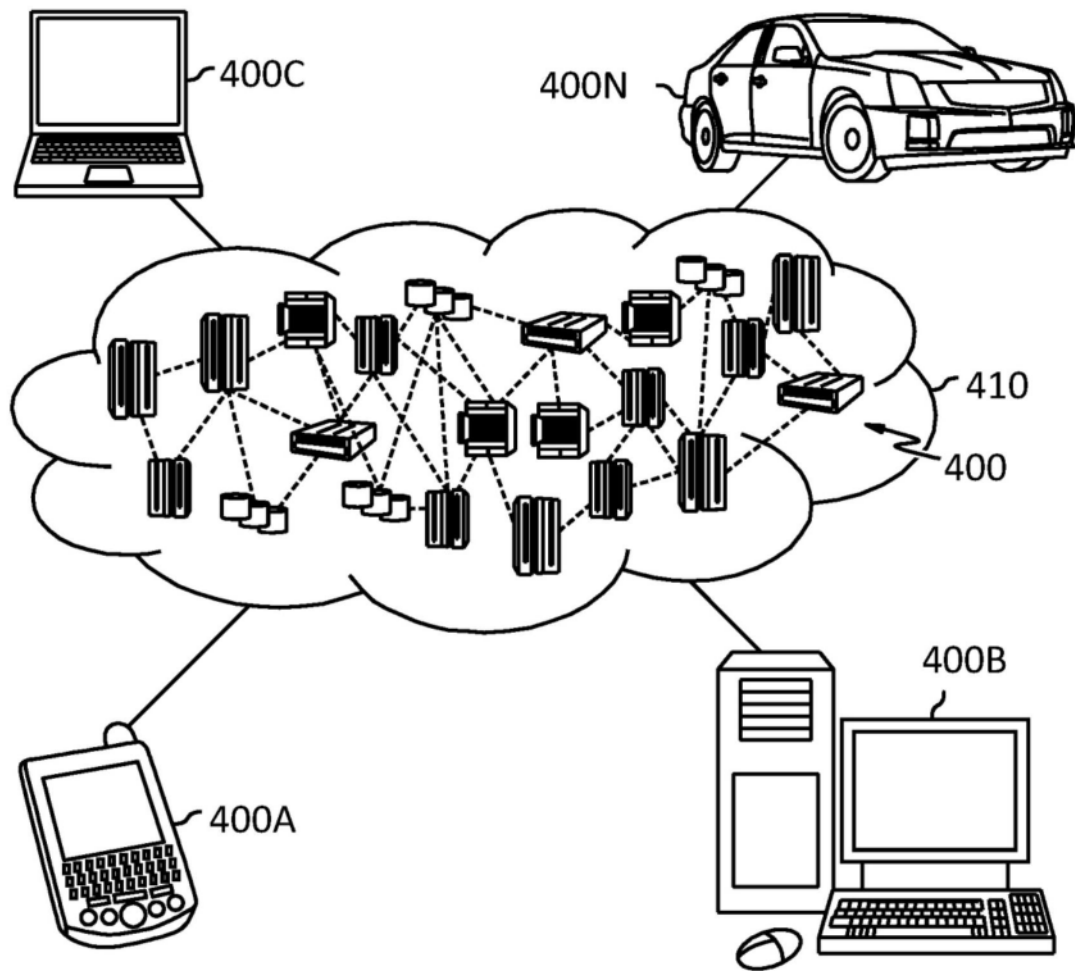


图4A

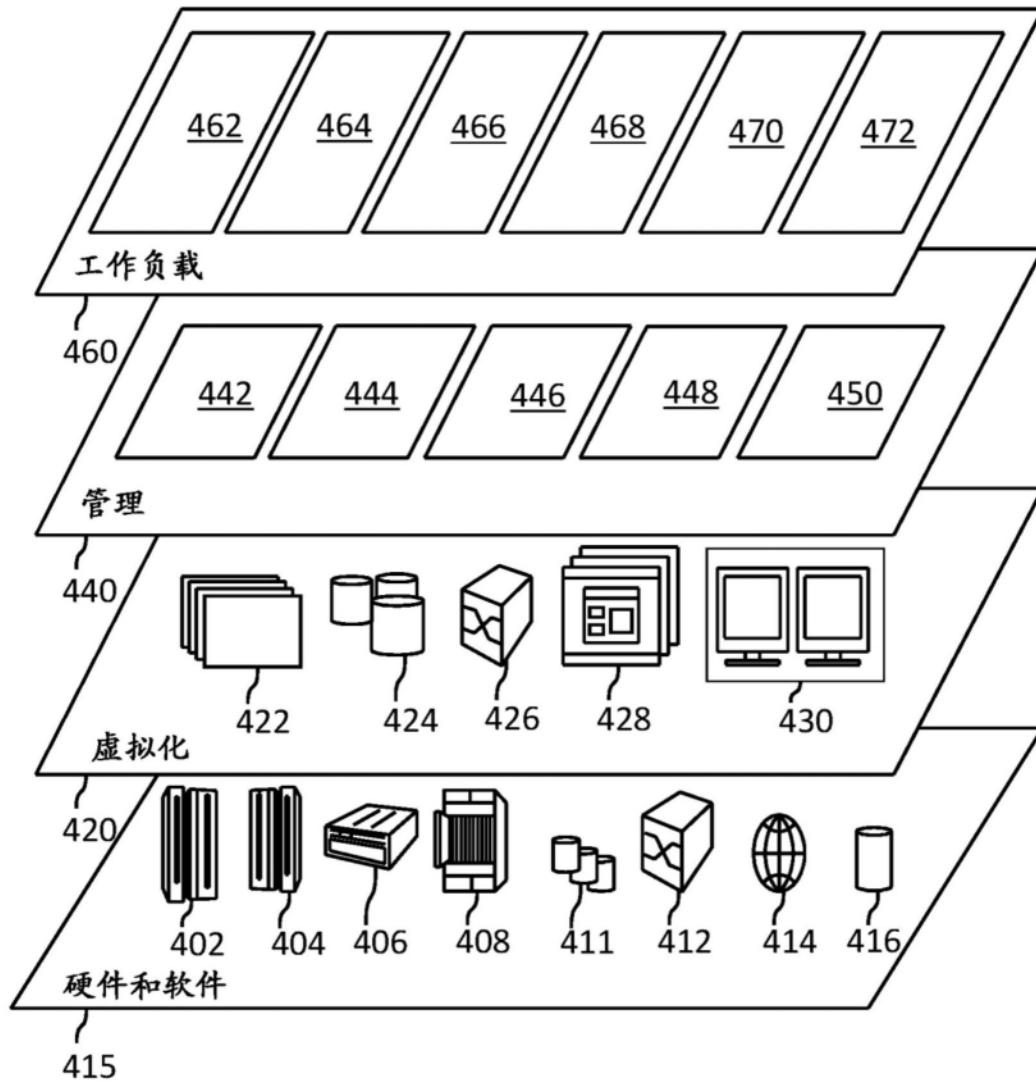


图4B

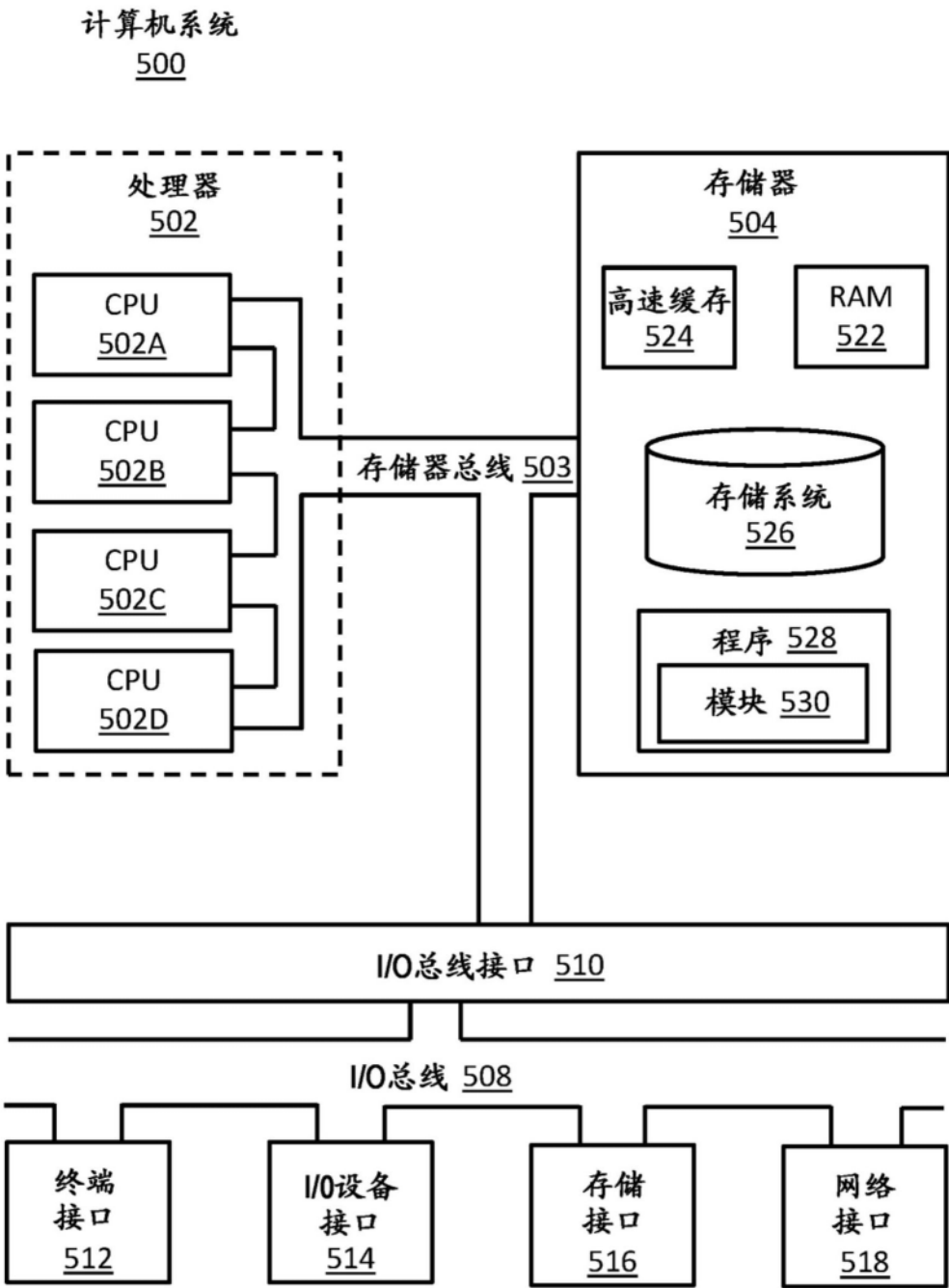


图5