



(12) 发明专利申请

(10) 申请公布号 CN 120051784 A

(43) 申请公布日 2025. 05. 27

(21) 申请号 202480003592.4

(22) 申请日 2024.07.22

(30) 优先权数据

63/516,966 2023.08.01 US

(85) PCT国际申请进入国家阶段日

2025.01.27

(86) PCT国际申请的申请数据

PCT/US2024/039031 2024.07.22

(87) PCT国际申请的公布数据

W02025/029514 EN 2025.02.06

(71) 申请人 欧特克公司

地址 美国加利福尼亚州

(72) 发明人 P·R·彼得森

(74) 专利代理机构 北京市柳沈律师事务所

11105

专利代理师 胡琪

(51) Int.Cl.

G06Q 10/0631 (2023.01)

G06N 20/00 (2019.01)

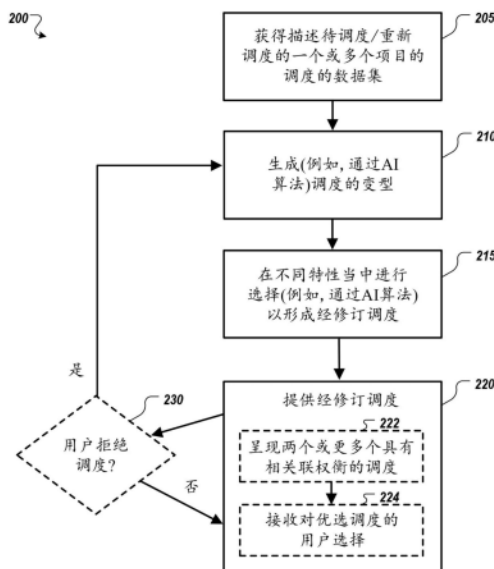
权利要求书2页 说明书17页 附图8页

(54) 发明名称

计算机辅助生成式任务调度

(57) 摘要

包括介质编码的计算机程序产品在内的用于计算机辅助生成式任务调度的方法、系统和设备包括：获得描述待重新调度的一个或多个项目的调度的数据集，所述数据集包括待调度的任务、资源需求以及任务之间的相依性；生成所述调度的变型，所述变型中的每一个具有不同特性，所述不同特性针对所述调度中的所述任务确定何时及时地调度每个任务，并且所述变型中的每一个满足所述资源需求和所述相依性；在所述调度的所述变型中针对所述任务的所述不同特性当中进行选择以形成所述一个或多个项目的经修订调度；以及提供所述一个或多个项目的所述经修订调度以供显示给用户或者供输出以管理所述一个或多个项目。



1. 一种方法,所述方法包括:

通过调度计算机程序获得描述待重新调度的一个或多个项目的调度的数据集,所述数据集包括待调度的任务的工作分解结构、资源需求以及所述任务之间的相依性;

通过所述调度计算机程序生成所述调度的变型,其中所述调度的所述变型中的每一个具有不同特性,所述不同特性针对所述调度中的所述任务确定何时及时地调度每个任务,并且所述变型中的每一个都满足所述资源需求和所述相依性;

通过所述调度计算机程序在所述调度的所述变型中针对所述任务的所述不同特性当中进行选择以形成所述一个或多个项目的经修订调度;以及

通过所述调度计算机程序提供所述一个或多个项目的所述经修订调度以供显示给用户或者供输出以管理所述一个或多个项目。

2. 根据权利要求1所述的方法,其中所述数据集除了所述相依性之外还包括一个或多个调度约束,并且所述变型中的每一个除满足所述相依性之外还满足所述一个或多个调度约束。

3. 根据权利要求1所述的方法,其中所述生成和所述选择由进化人工智能算法执行。

4. 根据权利要求1、2和3中任一项所述的方法,其中所述一个或多个项目为共享至少一个资源类别的两个或更多个项目,所述一个或多个调度约束包括针对所述两个或更多个项目中的每一个的至少一个时间约束,并且所述生成和所述选择通过使所述至少一个资源类别的利用率最大化同时还满足针对所述两个或更多个项目中的每一个的所述至少一个时间约束来形成所述经修订调度。

5. 根据权利要求1、2和3中任一项所述的方法,所述方法包括对所述数据集采用层状数据模型,其中所述层状数据模型包括

第一层,其指定表示所述调度的图的拓扑;

第二层,其至少指定对所述第一层做出局部改变的编辑操作;以及

第三层,其指定表示所述调度的所述图,包括所述调度中的所述任务之间的所述相依性的所有细节以及针对所述调度中的所述任务的所有经调度开始和结束时间。

6. 根据权利要求5所述的方法,其中所述调度计算机程序至少在远离由所述用户操作的客户端计算机的服务器计算机上运行,所述层状数据模型的所述第一层完全加载到所述客户端计算机的存储器中,并且响应于编辑操作对所述第二层的更新是在所述客户端计算机本地并发执行的并持久化到所述服务器计算机。

7. 根据权利要求1、2和3中任一项所述的方法,所述方法包括:

在图形用户界面中呈现所选择的资源类别的当前经规划使用率和所选择的资源类别的计算出的理想使用率;以及

经由所述图形用户界面接收用户输入,所述用户输入改变所选择的资源类别的所述计算出的理想使用率的形状,以创建所选择的资源类别的用户指定的理想使用率;

其中所述生成和所述选择通过以下来形成所述经修订调度:基于针对工作分布的目标函数来修改所选择的资源类别的所述当前经规划使用率以近似所述用户指定的理想使用率,所述目标函数被表达为所选择的资源类别的利用率与所选择的资源类别的所述用户指定的理想使用率的偏差。

8. 根据权利要求7所述的方法,其中所述图形用户界面在第一轴上示出时间并且在第

二轴上示出工作负荷,并且所述用户输入改变使用所选择的资源类别的开始日期、使用所选择的资源类别的结束日期、所述工作负荷的针对所选择的资源类别的斜升曲线、所述工作负荷的针对所选择的资源类别的斜降曲线或者它们的组合。

9. 根据权利要求7所述的方法,其中针对工作分布的所述目标函数使用非均匀和不连续的自变量,所述非均匀和不连续的自变量将时钟时间表示为自开始时间以来单调递增的累积可用工作时间量。

10. 根据权利要求9所述的方法,所述方法包括提供并发安全的缓式构建模式和用于评定所述时钟时间的高性能无锁只读模式两者。

11. 根据权利要求7所述的方法,其中所述生成和所述选择通过以下来形成所述经修订调度:基于(i)针对工作分布的第一目标函数以及(ii)针对工作分布的第二目标函数来修改所选择的资源类别的所述当前经规划使用率,所述第一目标函数被表达为指定的资源类别的当前利用率与所述指定的资源类别的理想利用率的偏差,所述第二目标函数被表达为使用所述指定的资源类别的所述任务的当前时间定序与使用所述指定的资源类别的所述任务的理想时间定序的偏差,其中所述任务的所述理想时间定序是从分配给使用所述指定的资源类别的所述任务的优先级值确定的。

12. 根据权利要求1所述的方法,其中所述生成和所述选择分两个阶段执行,在所述两个阶段中的初始阶段中,按项目或按子项目应用针对调度变型的相对浮动,并且在所述两个阶段的后续阶段中,按活动应用针对调度变型的相对浮动。

13. 一种非暂时性计算机可读介质,其对计算机辅助设计程序进行编码,所述计算机辅助设计程序可操作以使一个或多个数据处理设备执行根据权利要求1至12中任一项所述的操作。

14. 一种系统,所述系统包括:

一个或多个数据处理设备;以及

一个或多个非暂时性计算机可读介质,其对指令进行编码,所述指令可由所述一个或多个数据处理设备执行,以执行根据权利要求1至12中任一项所述的操作。

计算机辅助生成式任务调度

[0001] 相关申请的交叉引用

[0002] 本申请要求2023年8月1日提交的名称“计算机辅助生成式任务调度”的美国专利申请号63/516,966的优先权的权益。

背景技术

技术领域

[0003] 本说明书涉及计算机辅助任务调度。

[0004] 相关技术的描述

[0005] 以现实世界的尺度创建和修订调度(由于大小、复杂性或两者)是耗时且很大程度上手动过程,通常会阻止调度者探索替代场景并导致他们忽视节省时间、劳动力和成本的潜在机会。项目的初始调度通常是手工创建的,需要大量的时间和精力来构建,因为调度者会努力手动实现待完成的任务的水平资源利用率,同时适应他们的项目的相依性和约束。

[0006] 虽然调度的第一轮次可能相当合适,但由于许多原因,规划不可避免地会发生改变。当回应于改变而调整他们的调度太过耗时且负担过重时,相反,调度者会尝试继续使用他们的原始调度。因此,项目必须做出其他牺牲才能坚持其初始规划,所述其他牺牲通常呈以下形式:项目的工作人员加班,或者更糟糕的是,超出项目的预算或错过重要期限。

发明内容

[0007] 本说明书描述了与计算机辅助生成式任务调度相关的技术,并且特别描述了任务的调度,以便基于完成任务的要求来优化资源类别的利用率。生成式调度提供解决方案和 workflows,所述解决方案和 workflows 聚焦于以下问题:在高效且优化地使用资源的情况下调度或重新调度复杂项目(或两个或更多个互相关项目),并促进对替代调度场景的快速迭代(无论是在初步规划中还是响应于使原始调度不适合的改变),以探索在不同的时间和资源目标及约束的条件下的权衡。生成式调度为算法方法,其可以创建包含分层任务结构、相依性、时间约束和资源需求的基于时间的任务调度。这一方法可以经专门设计以根据所施加的结构来生成可行的调度,并且更重要的是,可以针对资源利用率来优化。此外,可以生成调度以(a)增加(或优化)任务与项目和/或子项目的连续性,和/或(b)基于经分配的优先级来改善(或优化)针对任务的定时,同时针对资源利用率进行并发优化。

[0008] 在一些实现方式中,生成式调度利用机器学习(诸如进化算法)来探索可能可行调度的广阔搜索空间。系统可以基于调度质量的一个或多个综合目标测度(也称为目标函数)来适应朝着期望的解决方案指导调度。需注意,由于问题的NP困难(非确定性多项式时间)复杂性,穷举或甚至迭代的细化搜索和优化方法通常不适用于许多现实世界调度问题。此外,依赖于基于规范数据集的训练的机器学习方法通常也不可用,因为一个调度问题与另一调度问题之间存在很差的(如果有的话)相关性。

[0009] 可以实现本说明书所描述的主题的特定实施例以达到以下优点中的一个或多个。

通过实现一个或多个项目中的任务的快速重新调度,会改善计算机的功能性,即使当没有类似的现有项目调度可以用于指导重新调度时。使用调度变型生成(例如,任务调度特性的随机变型)和选择会改善任务调度的技术领域,这解决以下技术问题:在没有类似的先前项目调度可以用于指导重新调度时如何使任务重新调度自动化。此外,还描述了用户界面控制,其提供持续和受指导的人机交互,以促进重新调度复杂项目(或两个或更多个互相关项目)的技术任务,例如,通过采用资源利用率整形用户界面控制。

[0010] 在一些实现方式中,使用进化人工智能(AI)算法以使用基本基因过程的简化模拟以及对良好调度的获悉贡献者从现有调度进化出更好的调度,如按目标所测量,所述目标可使用用户易于理解的高级控制来定义。可以按照考虑现实世界尺度和/或可用性的方式来将最先进的多目标进化技术应用于调度问题。此外,可以解决调度问题,而不必就时间如何表示或处理做出简化假设。

[0011] 如本申请所描述,生成式调度可以将创建针对复杂项目的调度从繁琐的手动过程在根本上转变为交互式的且富有成效的体验,实现对动态改变的更快反应以及更主动的策略以降低风险,因为必须根据新的发展来改变调度。生成式调度并不聚焦于单目标问题,诸如如对“资源约束调度问题”(RCSP)的研究中典型的使“完工时间”(调度的总持续时间)最小化,而是可聚焦于以下更一般的多目标问题:参考对多个可能竞争性的资源类别的理想目的利用率来优化资源利用率。此外,可以快速生成多个替代调度,允许系统的用户响应于出现使原始调度不适合或甚至不可能的一些改变而快速探索重新调度项目的替代方法,从而使调度改变能够在需要改变时尽快完成,这可以是每天或甚至每小时,而使用传统调度软件重新调度项目将需要两天或周或更多天或周。此外,由于现有方法通常或必然将潜在调度解决方案的集合限制为整体可能性的极小子集,因此得到的调度通常展现出比使用传统调度软件实际可能的更高的优化程度。

[0012] 在附图和以下描述中阐述了本说明书所描述的主题的一个或多个实施例的细节。本发明的其它特征、方面和优点将依据描述、附图和权利要求而变得显而易见。

附图说明

[0013] 图1示出了可用于促进计算机辅助生成式任务调度的系统的示例。

[0014] 图2示出了生成式任务调度的过程的示例。

[0015] 图3示出了生成式任务调度的过程的另一示例。

[0016] 图4A至图4C示出了用于资源类别整形的用户界面的示例。

[0017] 图5示出了层状数据模型的示例,所述层状数据模型可供本文档所描述的系统 and/或过程使用。

[0018] 图6是包括数据处理设备的数据处理系统的示意图,所述数据处理设备可被编程为客户端或编程为服务器并实现本文档所描述的技术。

[0019] 图7示出了生成式任务调度的过程的另一示例。

[0020] 在各种附图中,相似附图标记和名称指示相似元素。

具体实施方式

[0021] 图1示出了可用于促进计算机辅助生成式任务调度的系统100的示例。计算机110

包括处理器112和存储器114,并且计算机110可连接到网络140,所述网络可以是私有网络、公共网络、虚拟私有网络等。处理器112可以是一个或多个硬件处理器,所述一个或多个硬件处理器可各自包括多个处理器核心。存储器114可包括易失性存储器 and 非易失性存储器两者,诸如随机存取存储器 (RAM) 和闪速RAM。计算机110可包括各种类型的计算机存储介质和装置,其可以包括存储器114以存储运行在处理器112上的程序的指令,所述程序包括生成式任务调度器116,所述生成式任务调度器为实现任务调度例如以便基于完成任务的资源需求来优化资源类别的利用率的一个或多个程序116。

[0022] 程序116可在计算机110上在本地运行,在一个或多个远程计算机系统150 (例如,由计算机110经由网络140可访问的一个或多个第三方提供商的一个或多个服务器系统) 的计算机上远程地运行,或既在本地又远程地运行。在一些实现方式中,生成式调度技术可通过从网络浏览器访问的基于云的软件即服务获得,所述网络浏览器可包括一个或多个代码插件。此外,生成式任务调度器116可针对调度的数据集使用开放调度格式,例如以JSON (JavaScript对象表示法) 模式调度数据存储器,其中数据集描述待重新调度的一个或多个项目。在一些实现方式中,生成式任务调度器116包括一个或多个用户界面,所述一个或多个用户界面使得用户能够创建描述初始调度的数据集。在一些实现方式中,生成式任务调度器116从另一项目或调度或跟踪系统和/或程序 (诸如由加利福尼亚州旧金山的欧特克公司提供的SHOTGRID[®] 生产管理产品 (最近重命名为流程生成管理[™])) 导入描述初始调度的数据集。

[0023] 在任何情况下,都可在计算机110的显示装置120上的用户界面 (UI) 122中呈现当前调度132 (如数据集所定义),所述用户界面可使用计算机110的一个或多个输入装置118 (例如,键盘和鼠标) 来操作。在一些实现方式中,UI 122提供针对一个或多个项目的当前调度中的数据的通用甘特图视图。不论UI 122的呈现格式如何,UI 122都使得用户160能够查看、自省和调整生成式任务调度器116中打开的一个或多个项目的资源利用率。因此,生成式任务调度器116可用作场景规划程序,而不论生成式任务调度器116是否也可用作主要资源跟踪系统。

[0024] 需注意,虽然在图1中示出为单独的装置,但显示装置120和/或输入装置118还可彼此集成和/或与计算机110集成,诸如在平板计算机中 (例如,触摸屏可以为输入/输出装置118、120)。而且,计算机110可包括虚拟现实 (VR) 和/或增强现实 (AR) 系统或者为其一部分。举例来说,输入/输出装置118和120可包括VR/AR输入控制器、手套或其它手动操纵工具118a,和/或VR/AR头戴式耳机120a。在一些实例中,输入/输出装置可包括基于传感器的手部跟踪装置,所述传感器跟踪移动并重新创建交互,就像使用物理输入装置执行一样。在一些实施方案中,VR装置和/或AR装置可以是可能不需要连接到计算机110的独立装置。VR和/或AR装置可以是具有处理能力的独立装置和/或集成计算机诸如计算机110,例如具有输入/输出硬件部件诸如控制器、传感器、检测器等。

[0025] 在任何情况下,用户160都可与生成式任务调度器116交互以从基线调度创建多个调度场景。在每个场景中,用户160可添加、删除或修改时间约束,指定相对优先级,以及控制资源利用率目标和约束。对于每个场景,可例如使用基于生成式调度人工智能 (AI) 的引擎来调用生成式任务调度器116,以自动生成可行且经资源优化的调度。需注意,如本文所使用,“经优化” (“最优”或“优化”) 并不意味着在所有情况下都实现所有可能调度中的最佳

调度,而是意味着从根据调度的多个目标接近理想目的利用率的可能调度的有限集合选择最佳(或几乎最佳)调度。

[0026] 然后,一个或多个经选择的场景可:将其经优化的调度数据保存到调度文档130(本地地在计算机110处和/或远程地在计算机150处),所述调度文档可呈现在显示屏上;和/或将其经优化的调度数据135导出以用于生产管理或其他应用170。其他应用170可包括物理结构(例如,办公楼)建造任务调度、制造(例如,增材和/或减材)机器任务调度、动画/图形渲染任务调度(例如,电影制作项目)和计算机资源任务调度(例如,预测利用率,包括预取计算机资源和平衡用于计算的竞争性资源)。如将理解的,不同应用领域对任务调度可能有不同具体约束,包括潜在地基于位置的约束。在任何情况下,核心工作流程都是以下中的一者:通过提供简单、直接和交互式的高级控制来操纵调度特性,对多个替代调度(场景)进行迭代和探索;以及让计算机辅助生成式调度(例如,AI调度引擎)完成优化所述调度和确保可行性被保留两者的繁重差事。

[0027] 图2示出了生成式任务调度的过程200的示例。过程200的目标是在针对项目的调度中最优地利用资源。在205处,通过调度计算机程序(例如,生成式任务调度器116和/或调度程序604)获得描述待调度/重新调度的一个或多个项目的调度的数据集。获得205可包括生成数据集以定义调度或者从项目调度或跟踪系统接收/导入数据集(定义先前指定的调度)。一般来说,项目调度涉及满足约束、高效地使用资源和对改变做出反应。调度约束可包括相依性(定义的前导关系)、时间约束(与时间点的定义的关系)和界限(其中活动可以开始或结束的定义的间隔)。资源可具有类别(与项目相关的定义的资源类型)、要求(针对各种资源类别的定义的工作负荷)和目标(各种资源类别的随时间的定义的理想利用率)。所获得205的数据集可包括待调度的任务的工作分解结构、资源需求(例如,每个资源类别的工作负荷)、任务之间的相依性以及任选的除了相依性之外的一个或多个调度约束(例如,至少一个时间约束,其可由两个或更多个项目共享)。

[0028] 工作分解结构可以是基于标准调度术语的描述(或衍生自),所述描述详述按分层结构布置的待调度的活动的集合。活动可以是任务、里程碑或概要。任务可表示持续指定的工作持续时间需要某一资源集合(资源需求)的活动,其中每个资源需求由所需的资源类别和所需的所述类别的单位数目。里程碑可指示调度中的关键点,所述关键点指示整个调度中某个子区段的开头或结尾。概要可由活动群组组成。每个活动可描述对任何其他活动的相依性(前导关系)以及特定的时间约束、适用的工作时间日历和其他参数。前导关系不需要严格地指示活动遵循其时间上的相依性。相反,前导关系可描述预期活动如何在时间上相关的关系。在一些实现方式中,输入数据集为以开放调度格式模式(例如,描述模型被定义为生成式调度产品开发的一部分定义的基于JSON的清单)描述的数据模型。在一些实现方式中,提供Python库以实现开放调度格式的程序化构建。

[0029] 由于过程200的目标是在针对项目的调度中最优地利用资源,并且由于针对调度的可能替代方案的数目可能非常大(对于典型项目诸如电影制作项目,大约数百万或数十亿个可能调度,其中多个项目可同时运行、共享资源,并且项目中的每一个可包括数万个活动和数十万个任务),因此对整个调度空间的穷举搜索是不可行的。因此,所产生的“经优化”的调度是在竞争性的多个资源类别中的每一个的最佳利用率之间提供良好折衷的调度,并且具有快速生成新调度的能力可促进用户探索具有不同资源可用性(定时和/或数

量)或时间约束的调度之间的权衡,以确定“大小适度化”,使风险最小化,并管理成本。

[0030] 在210处,通过调度计算机程序(例如,生成式任务调度器116和/或调度程序604)生成调度的变型。调度的变型中的每一个可具有不同特性,所述不同特性针对调度中的任务来确定何时及时地调度每个任务,并且变型中的每一个可满足资源需求、相依性以及任选的除了相依性之外的一个或多个调度约束。在一些实现方式中,所生成210的两个或更多个变型的集合是随机变型,其可包括伪随机变型。在一些实现方式中,两个或多个变型的集合是使用预定义的规则和/或启发式方法来确定性地生成210的。

[0031] 所生成210的不同可能变型可基于浮动的概念。任何给定的活动都可在不导致违反任何可行的指定的约束的时间界限内浮动。在一些实现方式中,数据集的内部表示(例如,层状数据模型,其可以包括如下文所描述的有向无环图(DAG))经预处理以确定每个活动的最大浮动界限。基于这一点,可通过将(例如,DAG的)前导可行遍历中的一个与每个活动的值(例如,在范围[0..1]内)组合来生成调度变体,定义待使用的相对自由浮动量,但要根据遍历次序对所有前导进行调度。在一些实现方式中,会每个活动编码两个或更多个数值。此外,在生成210期间,可累积每个资源类别的利用率。

[0032] 在215处,通过调度计算机程序(例如,生成式任务调度器116和/或调度程序604)在调度的变型中针对任务的不同特性当中进行选择,以形成一个或多个项目的经修订调度。在一些实现方式中,生成210和选择215由进化AI算法执行。此类实现方式的进一步细节如下所描述,但在各种实现方式中可使用其他算法,包括其他AI算法或迭代启发式算法,诸如分支定界算法。一般来说,可使用处理多目标优化问题的任何适合算法(多目标优化算法)。此外,生成210和选择215的结果可以是单个经修订调度或多于一个经修订调度。

[0033] 在任何情况下,都可根据一个或多个目标测度来评估生成的调度的集合(例如,可每个目标评估一个或多个数值),并且基于目的测度来选择子集,尝试既包含最佳解决方案同时又保持多样性。然后,这一子集可用于例如使用伪基因繁殖算子(交叉和变异)的基因算法概念来生成新集合。可贯穿后续的代重复此过程,直到满足某些停止准则为止,留下进化的解决方案集。在一些实现方式中,可使用相对于先前的代有意义的改善的测度作为停止准则与根据调度复杂性或可接受的时间确定的固定最大值组合地产生结果。

[0034] 生成式过程可学习有助于生成良好调度的值,其中“良好性”是根据目标来分类的。最终的解决方案集可以为输出结果,展现根据测度优化的解决方案的集合。在一些情况下,输出结果为针对调度的单个解决方案,但在其他情况下,可呈现多个解决方案,由于调度的多目标问题而展现不同的权衡。在一些实现方式中,最终解决方案中的每一个都具有与其相关联的简单度量,所述简单度量指示(每个目标测度,例如,每个资源类别)其与期望的最优利用率目标的相似性。

[0035] 需注意,用于评估调度的质量的核心标准可包括多目标(通常为多个目标)公式。具体地,目标是优化调度所需的每个资源类别的资源利用率。对于每个资源类别,可在调度生成期间将这一资源利用率计算为时间序列。这形成与(离散)概率分布函数具有高度相似性的数据的集合。可通过创建目的理想分布函数(包括可能通过用户控制来对此分布进行整形,如下文进一步描述的)来形成单目标测度,然后可针对相似性将实际分布与理想分布进行比较,从而产生目标测度/函数。需注意,可引入与顺序连续性、相对优先级和其他因素相关的附加目标测度作为多目标公式中的附加维度。例如,顺序连续性的目标测度可寻求

使任务开始与相关任务完成之间引入的延迟量最小化。相对优先级的目标测度可量化不同优先级的两个或更多个任务的经调度时间(与这些任务的期望的相对优先级相关),而不管相依性如何。

[0036] 在220处,通过调度计算机程序(例如,生成式任务调度器116和/或调度程序604)提供一个或多个项目的经修订调度。这可以涉及呈现经修订调度以供显示给用户(例如,在显示装置120上的UI 122上供用户160查阅)和/或以供输出以管理一个或多个项目(例如,输出到调度文档130和/或经由调度数据导出135输出以用于生产管理或其他应用170)。

[0037] 在一些实现方式中,会呈现222两个或更多个经修订调度,任选地在针对不同调度示出相关联的权衡(例如,上面引用的简单度量)的情况下,并且会在输出特定经修订调度以管理一个或多个项目之前接收224对优选调度的用户选择。在一些实现方式中,会向用户呈现拒绝230所有提供的调度的选项,并且然后,过程200返回到新变型的生成210(例如,利用其新选择的参数)。

[0038] 图3示出了生成式任务调度的过程300的另一示例。在305处,通过调度计算机程序(例如,生成式任务调度器116和/或调度程序604)导入描述待重新调度的两个或更多个项目的调度的数据集。在此示例中,两个或更多个项目共享至少一个资源类别,但通常,两个或更多个项目可各自具有部分或全部不相关的资源需求。此外,所导入305的数据集可包括待调度的任务的工作分解结构、资源需求(例如,每个资源类别的工作负荷)、任务之间的相依性以及针对两个或更多个项目中的每一个的至少一个时间约束。

[0039] 在一些实现方式中,通过调度计算机程序(例如,生成式任务调度器116和/或调度程序604)呈现310所选择的资源类别(例如,用户选择的资源和/或由两个或更多个项目共享的至少一个资源类别)的当前经规划使用率以及所选择的资源类别的计算出的理想使用率。然后,通过调度计算机程序(例如,生成式任务调度器116和/或调度程序604)接收315用户输入,所述用户输入改变所选择的资源类别的计算出的理想使用率的形状,以创建所选择的资源类别的用户指定的理想使用率。然后,可通过修改所选择的资源类别的当前经规划使用率来形成经修订调度,以基于针对工作分布的目标函数来近似用户指定的理想使用率,所述目标函数表达为所选择的资源类别的利用率与所选择的资源类别的用户指定的理想使用率的偏差。例如,在320处,使至少一个资源类别的利用率最大化(例如,通过在生成式任务调度器116和/或调度程序604中执行生成210和选择215的AI算法),同时还满足针对两个或更多个项目中的每一个的至少一个时间约束。

[0040] 在一些实现方式中,会提供220一个或多个项目的经修订调度,如上文所描述。在一些实现方式中,经修订调度中的至少一个自动地变成新调度并且过程300返回到310。由于可非常快速地生成经修订调度,因此可实时地向用户提供经修订调度,并且过程300可充当有效的场景规划程序。

[0041] 在一些实现方式中,进化AI算法(也称为基因算法)用于使一个或多个资源类别的利用率最大化320。基因算法繁殖算子(交叉和变异)需要可变调度参数的表示,所述表示服从这些算子,并且是以鼓励有利特点的方式,同时维持多样性(避免过早收敛或过度繁殖)。可用于影响调度生成的参数的两个集合为(1)图(例如,结合图5讨论的DAG)遍历次序和(2)每个活动的相对浮动。浮动可被参数化为长度为n的向量,其中n为图中的节点数目,并且每个值都在范围[0..1]内,表示相对浮动的量。

[0042] 也可按照相同的方式利用第二向量来对遍历次序进行编码,其中每个 $[0..1]$ 值表示遍历次序中节点的优先级,以前导满足为准。以这一方式进行编码可保证所有可能的变型都存在,并且可使用适合问题域的算子。此外,可使这些算子保证产生的后代都会构成可行的调度。在一些实现方式中,模拟二进制交叉和多项式变异因其在这方面的良好性质而被使用。

[0043] 此外,在一些实现方式中,在进化过程期间使用的具体排序和选择准则基于适应性几何估计技术(AGE-MOEA),所述适应性几何估计技术使用非欧几里得几何来估计多目标问题的帕累托最优前沿。对于每一代,此算法使用基于多值目标测度(每个资源类别)的非支配排序方法对调度的集合进行排序,并且然后估计帕累托最优前沿并基于复杂存活评分来选择用以形成下一代的亲代的候选者,以利用多样性覆盖对此前沿进行目的取样。

[0044] 最后,在各种实现方式中,可在调度生成和进化过程两者中交错加入附加的优化和元启发式方法。这些可以包括:局部迭代搜索,用以尝试满足对最大利用率的严格限制;基于双向对齐和细化的对结果的局部改进;以及预调节分布函数,用以基于促进工作分解结构分层结构内的连续性的启发式方法来播种调度的初始集合。

[0045] 图4A示出了用户界面(UI) 450的示例,所述用户界面呈现与待重新调度的一个或多个项目相关联的多个资源类别。图4B示出了已经选择资源类别455之后的UI 450,其中UI 450通过在第一轴470上示出时间以及在第二轴472上示出工作负荷来示出当前调度中资源类别455的实际利用率。此外,UI 450示出UI元素460(图4B中的框),所述UI元素表示理论上的理想资源利用率,所述理想资源利用率可基于利用此资源类别的一个或多个项目中的一个或多个活动(各自包括多个任务)的最早可能开始日期和最晚可能结束日期(即,基于调度约束)来自动地计算。然后,可使用资源整形来允许用户通过UI 450以交互方式修改理论上的理想资源利用率。

[0046] 为了促进这一过程,UI元素460可包括一个或多个部分,用户可直接修改所述一个或多个部分以改变使用所选择的资源类别的开始日期和/或使用所选择的资源类别的结束日期。例如,用户可选择并拖动开始日期UI元素462A和/或结束日期UI元素462B来调整使用所选择的资源类别的开始和结束日期。作为另一示例,用户可选择并斜升曲线UI元素464A和/或斜降曲线UI元素464B来修改工作负荷的针对所选择的资源类别的斜升和斜降曲线。在一些实现方式中,使用贝塞尔曲线来定义斜升和斜降曲线,但可使用各种类型的UI元素。

[0047] 图4C示出了开始日期已提前并且斜升和斜降曲线已调整之后的UI 450。如将理解的,这仅仅是用户可进行的修改类型的一个示例。在一些实现方式中,使用UI元素460来改变开始或结束日期会使得系统根据所有工作调度日历(其可以为不连续的工作日)和调度约束自动地重新计算UI元素460下的总面积(资源利用率目的)。此外,在一些实现方式中,使用UI元素460来改变斜升或斜降会使得系统(例如,AI算法)自动地优化调度以达到指定的目的工作负荷斜升和/或斜降。

[0048] 因此,经由UI元素460的资源整形会完成两件事:(1) 它提供高级控制,以创建实际的现实目标函数,以供调度计算机程序在优化调度时使用资源利用率;以及(2) 它方便使用,因为可为用户提供图形用户界面(GUI),以提供关于资源利用率应是什么样的的一般描述,并且然后让调度计算机程序弄清楚如何得出正确的目标函数以及实时地更新资源利用率目的两者,并且然后使用这两者来确定针对这一新资源利用率目的的经修订调度。需注

意,经由UI 450的这一调整不仅仅改变调度计算机程序可使用来完成优化的目标函数;所述调整还将所述目标函数用作对调度的控制形式。

[0049] 理想分布定义可与时间约束相组合,作为用于高级控制和操纵的方法。此外,不是简单地将模型分布函数表示和/或得出为固定量,而是可以利用用户定义的单调曲线来表示和/或得出模型分布函数。还需注意,根据工作调度,对在给定日的工作可用的时间量每天都是变化的。因此,当计算面积时,面积不是简单的总和/积分,而是在不连续的时间内计算的。计算具有非均匀和不连续的自变量的分布函数对于具有不同工作调度(每天包括变化的工作时间)的现实世界问题非常重要。

[0050] 再次参考图1,在一些实现方式中,针对工作分布的目标函数使用非均匀和不连续的自变量,所述非均匀和不连续的自变量将时钟时间116a表示为自开始时间以来单调递增的累积可用工作时间量。执行调度图评估所需的计算机资源受支配于按照某个时钟向前和向后在时间上进行索引。然而,在现实世界调度中,时钟可以既是不连续的,又是非均匀的(但是是单调的)。此外,同一调度内可存在许多不同时钟,并且针对调度计算机程序的关键操作可能需要从一个时钟空间变换到另一时钟空间。使用表示时钟时间116a的非均匀和不连续的因变量有利于在时间上向前和向后进行索引(因为这可以在共享的时钟时间116a内完成)以及从一个时钟空间变换到另一时钟空间(因为每个不同时钟空间可轻松地转换为共享的时钟时间116a或从其转换)两者。

[0051] 在一些实现方式中,系统提供访问模式116b,包括并发安全的缓式构建模式和用于评定时钟时间116a的高性能无锁只读模式两者。对于这两种模式,均可对照可用工作时间的单调累积表示来完成索引和增量计算,所述单调累积表示允许高效二元搜索方法在不连续和非均匀的时间内递增。此外,时间可以与归一化的一天的开始达成一致,以从计算析出时区信息。一天的时钟可表示为工作秒数(相对时间)对绝对时间。这对于特定的调度情况来说是高效的(虽然对于一般的时间/时钟表示来说不是),并且从而减少执行如本文档所描述的生成式调度所需的计算资源。

[0052] 在一些实现方式中,为了每个资源类别(其可被进一步限制为每个项目)定义一目标函数,考虑待调度的总工作负荷。可将理想目标函数描述为某一时间范围内工作负荷的离散分布。理想目标函数的积分应等于总工作负荷。根据给定的调度变型,可计算出工作的实际离散分布。实际与理想的散度的测度充当目标函数的值。

[0053] 可通过缩放单位分布来得出理想分布,使得离散积分(或仅仅是分布曲线下的面积)等于总工作负荷。可利用以下来定义常见分布:时间上的开始和完成点以及分布在其处“斜”升至其峰值并从其峰值“斜”降的任意的点。开始/完成点和斜坡点之间的插值可采用如上文所描述的贝塞尔曲线,或者插值可以为线性的或某个其他适当的非线性定义。需注意,单位分布的缩放因子的计算应将以下列为因素:所述缩放因子在时间维度上是非均匀的(就可以假设的可用工作负荷而言)或不连续的。此外,除了定义目的分布之外,理想曲线的开始和结束点还充当约束,使得开始和结束点形成针对需要此资源类别的任务所具有的最早允许开始时间和最晚允许完成时间的界限。

[0054] 再次参考图4B,对理想曲线460的控制点462A、464A、462B、464B的调整可导致实时计算工作负荷分布的尺度,使得根据上文定义的计算要求,其积分与总工作负荷相匹配。另外,还可以参考针对资源类别的可用容量的时间序列。然后,可对理想曲线460进行裁剪(如

果需要) 以确保理想曲线不超过可用的可变容量, 并且所得理想曲线的尺度会补偿任何容量裁剪。在一些实现方式中, 调度计算机程序响应于对资源类别的理想曲线460的用户编辑基于待完成的工作以及工作调度(包括非工作日和经调度的加班) 来连续地计算所需的资源数目。

[0055] 调度定义可包括一个或多个工作调度(或工作时间日历), 其中每个工作调度定义在给定日期可用的工作时间量。每个任务具有相关联的工作调度。在生成调度时, 每个任务将基于其前导的经调度时间(根据相依性) 和任选的约束来得到最早可能开始时间。为了计算实际经调度开始时间, 应将最早开始时间变换为待调度的任务的工作时间日历。并且, 为了计算任务的完成时间, 应将工作时间内的持续时间添加到任务的工作时间日历中的开始时间。这涉及在非均匀和不连续的空间中计算时间增量/偏移(本质上是加法)。高效地完成这一点(如本文档所描述) 实现达到期望的(例如, 实时) 性能, 因为受影响操作会在现实世界项目的重新调度期间被执行大量次数。另外, 支持成为与时区无关的日期/时间的概念也很有用。在没有这一高效方法的情况下, 将需要迭代前向搜索和递增, 这在持续时间不是很短的情况下需要向前的许多步骤, 同时工作量会相对于持续时间的长度而增加。

[0056] 为了达到期望的效率, 可将感兴趣范围的工作调度变换为从给定开始点开始的累积工作时间量。具体地, 将开始日定义为通用世界时钟(UTC) 中从某个时期开始的秒数。对于随后的每一日, 在针对时钟的条目(称为“时间表”) 中, 存储累计工作时间量(包括当日的结束)。这定义不受时区影响的时钟(即, 相对时间), 所述时钟可能是非均匀的和不连续的。效率来自于利用时钟的这一分布。因此, 给定任何时间点, 可使用二元搜索找到所述时间点存在的间隔并且然后从累积时间转换为相对时间来得出实际日期时间, 所述相对时间进而可简单地变换为实际日期/时间。例如, 即使在总调度持续时间为大约三年的情况下, 对于长达三年的任何量的持续时间, 都最多需要十个增量。使用纯前向搜索策略, 任何十天或更长(也可能更短) 的持续时间都将超过这一时间, 并且缩放效率低。另外, 可贪婪地构造这些时间表(包括在多线程上下文中), 并且然后使用只读无锁实现方式116b对其进行持续查询, 以达到甚至进一步提高的性能。

[0057] 为了完成项目的重新调度, 多次评估与多个资源相关联的日历, 并且这些日历中的每一个可具有不同工作日并且在每个工作日具有不同可用小时数。自定义时钟实现允许以通用时间表示每个不同工作日历, 其中时钟表示自项目的时间的开始以来已过去多少可用工作时间。因此, 针对资源的单个时钟时间可对应于许多不同现实时钟时间, 并且针对此资源的时钟时间仅在针对此资源的工作小时期间推进。时钟的这种实现使得能够在重新调度期间对实际时间点进行高效搜索(例如, 使用二元搜索算法), 并且由此提供大规模的高性能。需注意, 不仅可在单独资源级别, 还可在单独任务级别实现独特时钟。每个任务都可以具有其自己独特的时钟时间, 因为任务可能需要多于一个资源, 并且这种任务将具有涵盖此任务所需的所有资源的复合的日历。

[0058] 此外, 图5示出了层状数据模型500的示例, 所述层状数据模型可由本文档所描述的系统和/或过程采用, 以便减少所使用的计算资源和/或减少由本地计算机110上的程序116与远程计算机150上的程序116之间的计算机通信消耗的带宽, 从而通过减少用户在UI上执行操作与显示执行所述操作的结果之间的操作时延, 使计算机能够在合理的时间量内对复杂的现实世界项目实际执行调度过程。层状数据模型500可理解为三个层(虽然这三个

层在每一层之间不需要具有严格的边界),其中每一层在此层建模的内容以及此层的持久化(保存以供长期存储)和输送(通过网络传送)方式方面有所不同。

[0059] 虽然在各种实现方式中可使用不同数目的层,但层状数据模型500的设计以解决以下问题。调度图模型的基本特性是局部改变可对图中一些(或全部)节点产生传播效应。在数据集驻留的桌面应用中,这通常不是问题。然而,在基于云的服务中,于服务器侧存在持久性,同时数据通过某个网络协议递送到客户端。随着数据集大小的扩大,这会引起问题,因为单个局部改变可能需要(a)整个数据模型的重新加载和重新持久化;和/或(b)整个数据模型的重新传输。这一问题以服务或微服务架构中的REST API(代表性状态转移应用程序编程接口)为典型。当使用多租户和/或无服务器架构时,在高速缓存一致性较难的情况下,也会加重这些问题。层状数据模型500可用于达到可复合性(即,程序实现的子部件,例如模块化软件例程,可轻松组合以形成复杂系统),而不是使用REST机制来传播本地编辑,这可能会导致网络连接的过载,并且由于从云发送和接收数据的高级别而导致操作时延的显著增加。

[0060] 在一些实现方式中,三层数据模型的第一层505指定表示调度的图的拓扑。这一第一层表示图的大部分不可变的数据拓扑,也可表示可变属性的时间点视图。本质上,会建立跨一定数目的调度相对固定的调度拓扑(编码工作分解结和调度拓扑是基本规划,如图所示)。例如,可将所生成或所导入的数据集(例如,基于JSON的清单)变换(针对内部系统表示)为单分区有向无环图(DAG)。此DAG基于来自输入模型的前导关系,但也可包括将所有分层构建体和约束的变换为单个统一DAG模型以供高效评估。内部表示可专门针对调度生成来优化,并且可包括实现去重复的不连续单调时钟(时钟时间116a),以用于计算具有多个工作时间日历的时间偏移。

[0061] 三层数据模型的第二层510至少指定对第一层进行局部改变的编辑操作。如图所示,稀疏编辑操作可用作底层调度拓扑的修改器。这些编辑操作可嵌入并复合到可在运行时工作的评估图515(网络)中。此外,调度变型编码可进入单独的数据流,所述单独的数据流被复合和分层在评估图515之上,这促进高性能高速缓存、数据输送和执行编辑操作,所述编辑操作可能会导致在整体所得复合评估数据结构中传播大量改变,但可以按照受保证可再现的方式通过层状数据模型中的小局部改变来描述。

[0062] 三层数据模型的第三层520指定表示调度的图,包括调度中的任务之间的相依性的所有细节以及调度中的任务的所有经调度开始和结束时间。此外,第三层520可在表示调度的图的规范中包括定义调度变体的特性以及确定针对每个任务的潜在变化范围的其他参数。第三层可理解为包含传播效应的整个图的高度可变性质的扁平化层。

[0063] 此外,在一些实现方式中,调度计算机程序至少在远离由用户操作的客户端计算机110的服务器计算机150上运行,层状数据模型500的第一层505完全加载到客户端计算机110的存储器中,并且响应于编辑操作对第二层510的更新是在客户端计算机110本地并发执行的并持久化到服务器计算机150。在一些实现方式中,除了服务器侧的存储和操作之外,客户端110还在存储器中针对维护当前和最近的工作集的3层高速缓存,所述3层高速缓存可实现为在不同线程上运行的嵌入式服务。高速缓存可用于通过从高速缓存加载第一层505、应用第二层510、以及然后进行评估以生成第三层520来补充(填充或充填)完整的评估数据结构。

[0064] 组合产生高效完整的实时数据模型。在编辑操作时,唯一需要传输到网络的数据是本地经改变属性(第二层的元素)。可从客户端高速缓存并发到服务器侧将改变持久化(并可能对其进行操作)乐观且确定性地评估操作的结果。

[0065] 在一些实现中,形成分层数据模型实现的公共组件550可在整个平台架构中被利用。在服务器计算机(其可包括短暂的无服务器运行时)上,可从操作的母语(例如,Go开源编程语言)并任选地使用GraphQL(或类似的API查询语言)来部署和/或编译对层状数据模型的该操作。对于较大规模的计算需求(所运行的优化过程,例如,当用户完成做出改变并且触发对调度的优化时),可部署相同的实现方式,但可使用多线程来更快地生成结果,例如,每次求解在许多(50到100个,或64到100、500、1000或更多个)中央处理单元(CPU)150上多线程地运行,并且在一些实现方式中可使用可缩放的工人池。需注意,与以完全序列化方式运行求解相比,在相应线程中使用具有独立进化(例如,进化AI算法中的生成)的多线程可提供对处理时间的显著改善(高达100倍的改善)。最后,客户端计算机上的应用可包含嵌入式微服务,并且可能包含多层高速缓存(如上文所描述),其中对编译为WASM(WebAssembly)部件的层状数据模型具有相同的操作。

[0066] 在此类实现方式中,可利用完全相同的代码使可复合评估图可用,所述完全相同的代码在计算引擎中运行,在来自API的无服务器可缩放部件中运行,或者在编译到WebAssembly中并嵌入为在Web客户端本身内具有通信协议的微服务下运行。因此,尽管有基于云的服务模型,但仍提供一致的图结构和用于数据访问的一致性的接口,其中可轻松实时地完成优化,使得对影响大量数据的约束的编辑可在客户端应用程序内(例如,在浏览器程序内)近乎实时地(例如,实时地)得到计算,并产生向用户示出的结果。

[0067] 图6是包括数据处理设备600的数据处理系统的示意图,所述数据处理设备可被编程为客户端或编程为服务器并实现本文档所描述的技术。数据处理设备600通过网络680与一个或多个计算机690连接。虽然在图6中仅一个计算机被示出为数据处理设备600,但可使用多个计算机。数据处理设备600包括可分布在应用层与操作系统之间的各种软件模块。这些可包括可执行和/或可解译的软件程序或库,包括实现资源类别利用率优化的一个或多个调度程序604的工具和服务。调度程序604可以为项目调度程序或项目跟踪程序,或者调度程序604可以为项目调度程序或项目跟踪程序的附加组件。在任何情况下,调度程序604都可提供场景规划功能性来评定与各种可能调度相关联的权衡。所使用的软件模块的数目可因实现方式而变化。此外,软件模块可以分布在由一个或多个计算机网络或其他合适的通信网络连接的一个或多个数据处理设备上。

[0068] 数据处理设备600还包括硬件或固件装置,包括一个或多个处理器612、一个或多个附加装置614、计算机可读介质616、通信接口618和一个或多个用户接口装置620。每个处理器612能够处理用于在数据处理设备600内执行的指令。在一些实现方式中,处理器612是单线程或多线程处理器。每个处理器612能够处理存储在计算机可读介质616或存储装置(诸如附加装置614中的一者)上的指令。数据处理设备600使用通信接口618与一个或多个计算机690通信,例如,通过网络680进行通信。用户接口装置620的示例包括显示器、相机、扬声器、麦克风、触觉反馈装置、键盘、鼠标以及VR和/或AR装置。数据处理设备600可将实现与上述程序相关联的操作的指令存储在例如计算机可读介质616或一个或多个附加装置614(例如硬盘装置、光盘装置、磁带装置和固态存储器装置中的一者或多者)上。

[0069] 图7示出了生成式任务调度的过程700的另一示例。在705处,通过调度计算机程序(例如,生成式任务调度器116和/或调度程序604)定义描述待调度/重新调度的一个或多个项目的调度的数据集。定义705可包括获得205、导入305、呈现310和/或接收315,如上文所描述。此外,呈现310可包括呈现分配给共享一个或多个指定的资源类别的任务的优先级值的当前(或默认)集合,并且接收315可包括接收对分配给任务的优先级值的用户指定的改变。

[0070] 这允许用户定义针对调度的软时间约束,所述软时间约束鼓励尽早调度重要的项目和/或任务。如上文所指出,可引入与相对优先级相关的附加目标测度作为多目标公式中的附加维度。在许多情况下,期望能够指定需要资源类别的任务比另一也需要同一资源类别的任务相对更重要。具体地,如果可能的话,相对较不太重要的任务应被调度为在相对更重要的任务的经调度开始时间处或之后开始。然而,不应以牺牲优化资源利用率为代价来严格实施相对优先级。因此,相对优先级可被定义为目标函数而不是约束。

[0071] 因此,用户界面可被设计为允许用户添加优先级值(例如,0至100的整数)来指导多目标优化算法(例如,如上文所描述的进化AI算法)按照用户喜欢的方式对任务进行定序。因此,可使用除了针对工作分布的目标函数之外的一个或多个附加目标函数来完成生成和选择710以形成经修订调度,所述目标函数被表达为指定的资源类别的当前利用率与指定的资源类别的理想利用率的偏差(如上文结合生成210、选择215和使利用率320最大化所描述)。在一些实现方式中,一个或多个附加目标函数包括针对工作分布的目标函数,所述目标函数被表达为使用指定的资源类别的任务的当前时间定序与使用指定的资源类别的任务的理想时间定序的偏差,其中任务的理想时间定序是从分配给使用指定的资源类别的任务的优先级值确定的。因此,在710处,可通过调度计算机程序(例如,生成式任务调度器116和/或调度程序604)使用与待调度的各种任务的相对优先级相关的附加目标函数来完成生成和选择710以形成经修订调度。

[0072] 在一些实现方式中,这是使用向量以表示需要共享的资源类别的每个任务集合来完成的,其中任务的针对此共享的资源的当前时间定序是通过向量中引用的任务的当前位置定义的,并且任务的理想时间定序是通过基于经分配的优先级值对向量中的任务进行重新定序而确定的。然后,相对优先级目标的运行时评估涉及向量的简单比较,以评定任务的理想时间定序与任务的当前时间定序之间的偏差。需注意,相对优先级的这一基于偏差的目标测度与上文描述的指定的资源类别的当前利用率的基于偏差的目标测度非常相似。

[0073] 例如,假设向需要给定资源类别的任务分配的 n 个唯一但任意的整数优先级值的经定义集合,我们可以将相对优先级的集合归一化为整数的序列 $[1..n]$ 。给定需要同一资源类别的 m 个任务的集合,我们则可以得出长度为 m 的目标向量,每个元素都为任务的归一化优先级值,并且向量按优先级次序排序。例如,对于9个任务的集合,利用3个唯一的优先级值,这会产生向量 $[1,1,1,2,2,2,3,3,3]$ 。可以针对每一资源类别得出类似的目标向量。

[0074] 当评估调度时,会针对每一资源类别创建类似的向量,但现在将其根据任务的经调度开始时间进行排序。例如,产生一个可能的向量 $[1,1,1,2,2,3,2,3,3]$ 。可以得出目标函数,其对优先级与目标优先级定序的散度进行测量。例如使用测量两个向量之间的距离的L2范数,或另一散度量,诸如Kullback-Leibler或Jensen-Shannon散度。使用此方法会确保向量的各个元素的散度越大,目标测度就会呈指数级缩放。例如,以上示例的目标值将

为 $2^{0.5}$ (1.4142), 而 $[1, 1, 3, 2, 2, 2, 1, 3, 3]$ 的散度的值将为 $8^{0.5}$ (2.8284)。此外, 在逐资源类别的基础上, 还可缩放相对优先级和利用率目标的权重, 以给予相等或优先的权重。例如, 对于给定资源类别, 经评估的目标函数值可被视为单个组合和归一化的目标函数, 其中如果利用率的目标值为 u , 优先级的目标值为 p 并且优先级的重要度为 w , 其中 w 为 $[0..1]$, 则复合的目标函数 $r = w * p + (1 - w) * u$, 使得 $w = 0.5$ 的值给予优先级和利用率两者相等重要度, 而 $w = 0.8$ 的值相比利用率给予优先级相当大的重要度。

[0075] 此外, 在一些实现方式中并且当定义705两个或更多个项目时, 使用两轮次方法进行调度/重新调度。两个或更多个项目可共享一个或多个资源类别, 并且因此需要作为全包式项目的一部分被一起调度/重新调度。对于这种全包式项目调度, 用户可能希望每个经定义子项目中的任务在时间上保持紧密, 即使这在存在其他子项目竞争相同资源时可能并不合适。为了达到此目标, 生成和选择可分两个阶段执行, 第一阶段的生成和选择710将相对浮动(如上文所描述)以按项目(或按子项目)的级别应用于调度变型, 并且第二阶段的生成和选择715以活动(或任务)级别应用相对浮动。需注意, 在一些实现方式中, 生成和选择715还可采用针对相对优先级的第二目标函数, 如上文所描述。此外, 现在所描述的两阶段方法可与使用或不使用与顺序连续性相关的上述附加目标测度(寻求使任务的开始与相关任务的完成之间引入的延迟量最小化)组合地实现为多目标公式中的附加维度。

[0076] 如上文所指出, 两阶段方法可包括第一阶段的生成和选择710, 所述第一阶段在按项目或按子项目的级别将相对浮动应用于调度变型。此外, 每个子项目可以为单个项目中或两个或更多个需要协调的项目中工作分解结构(WBS)的预定义和/或用户定义的块。因此, 可在第一阶段中在子项目级别在单个项目内完成使用两阶段方法。

[0077] 当对任务进行调度以优化资源利用率时, 各个任务可能会比其最早开始时间有所延迟。对需要同一资源类别的任务的给定集合应用延迟量的选择可能会导致工作分解结构(WBS)的公共子集不期望地缺乏连续性。例如, 考虑WBS的各自由三个相关任务组成的两个不相关子树, A_1 、 B_1 、 C_1 以及 A_2 、 B_2 、 C_2 。并且, 其中任务 A_1 和 A_2 需要资源类别A, 对于B和C也是如此。下表1所示的调度展现了每一资源类别的经优化使用。

[0078]

项目 1								
A1								
						B1		
							C1	
项目 2								
		A2						
				B2				
						C2		

[0079] 表1

[0080] 然而, 与下表2所示的替代调度相比, 上文示出的第一工作分解(A1、A2、A3)在连续性方面具有不期望中断。

[0081]

项目 1								
A1								

[0082]

		B1						
				C1				
项目 2								
		A2						
				B2				
						C2		

[0083] 表2

[0084] 虽然这是简单示例,但在具有WBS的数千个子树的调度中,确保合理的连续性同时又优化资源使用率会得到天文数字的可能的布置。

[0085] 使用多分辨率方法的概念(概念上类似于数值算法中的多网格方法),可在两阶段方法的第一阶段中对进化算法中的调度的群体进行预调节。首先,对初始种群进行约束,使得WBS的子树以块为单位被延迟,即,以子树为单位被延迟,其中 $A_1 \rightarrow B_1 \rightarrow C_1$ (呈前导关系),只有 A_1 可以被延迟,而 B_1 和 C_1 则会遵照其前导关系和其他约束尽快被调度。这可称为块约束(其中相对浮动被应用于整个块而不是按活动或任务来应用)。在此阶中运行的进化算法具有较粗(较低分辨率)结果作为其结果,但连续性得到严格维持。接下来,去除块约束,并进行进化算法的第二阶。然而,群体现在被预调节,以便获得高分辨率结果,但倾向于保留期望的连续性。此外,在此第二阶中,可引入连续性的目标测度(所使用的任选延迟量)来添加附加压力以保留连续性。

[0086] 一般来说,在第一阶段,项目(或子项目)会被来回移动以生成不同的调度选项以供选择,并且在第二阶段,活动(或任务)会被来回移动以生成不同的调度选项以供选择。因此,当产生初始粗略调度时,在第一阶段会将每个项目(或子项目)内的任务保持在一起,并且然后第二阶段不需要在调度中进行太多的任务移动来达到最优解决方案。最终所得调度如用户所期望在每个项目(或子项目)内提供更多的顺序连续性,而不要求使用顺序连续性目标函数作为多目标公式中的附加维度。

[0087] 本说明书中描述的主题和功能操作的实施例可在数字电子电路中实现,或在计算机软件、固件或硬件(包括本说明书所公开的结构及其结构等同物)中实现,或以它们中的一者或多者的组合实现。本说明书中所描述的标的物的实施例可使用编码在非暂时性计算机可读介质上以供数据处理设备执行或用于控制数据处理设备的操作的一个或多个计算机程序指令模块来实施。所述计算机可读介质可为制造产品,诸如计算机系统 中的硬盘驱动器,或通过零售渠道销售的光盘,或嵌入式系统。可单独获取计算机可读介质,并且稍后利用一个或多个计算机程序指令模块来编码,例如,在通过有线或无线网络递送一个或多个计算机程序指令模块之后。计算机可读介质可以是机器可读存储装置、机器可读存储衬底、存储器装置或它们中的一者或多者的组合。

[0088] 术语“数据处理设备”涵盖用于处理数据的所有设备、装备和机器,包括例如可编程处理器、计算机或多个处理器或计算机。除了硬件之外,所述设备还可包括为正在讨论中的计算机程序产生执行环境的代码,例如,构成处理器固件、协议栈、数据库管理系统、操作系统、运行时环境或它们中的一者或多者的组合的代码。此外,所述设备可以采用各种不同计算模型基础设施,诸如web服务、分布式计算和网格计算基础设施。

[0089] 可以任何适合形式的编程语言来编写计算机程序(还被称为程序、软件、软件应用

程序、脚本或代码),所述编程语言包括经编译或经解译语言、声明性或程序性语言,并且可以任何适合形式来部署所述计算机程序,包括部署为独立程序或适合供在计算环境中使用的模块、组件、子例程或其它单元。计算机程序不一定与文件系统中的文件相对应。程序可存储在保存其它程序或数据(例如,存储在标记语言文档中的一个或多个脚本)的文件的一部分中、存储在专用于所讨论的程序的单个文件中,或者存储在多个协同文件(例如,存储一个或多个模块、子程序或代码的部分的文件)中。计算机程序可被部署成在一个计算机上执行,或者在位于一个场所或跨越多个场所分布并且通过通信网络互连的多个计算机上执行。

[0090] 本说明书中所描述的过程和逻辑流可以由一个或多个可编程处理器执行,所述一个或多个可编程处理器执行一个或多个计算机程序以通过对输入数据进行操作并且产生输出来执行功能。所述过程和逻辑流还可以由专用逻辑电路执行,并且还可将设备实施为专用逻辑电路系统,所述专用逻辑电路系统,例如,FPGA(现场可编程门阵列)或ASIC(专用集成电路)。

[0091] 适合于执行计算机程序的处理器包括例如通用和专用微处理器两者,以及任何种类的数字计算机的任何一个或多个处理器。一般来说,处理器将从只读存储器或随机存取存储器或两者接收指令和数据。计算机的基本元素是用于执行指令的处理器和用于存储指令和数据的一个或多个存储器装置。一般来说,计算机还将包括用于存储数据的一个或多个大容量存储装置,例如,磁盘、磁光盘或光盘,或者操作性地耦合以从所述一个或多个大容量存储装置接收数据或向其传递数据,或两者。然而,计算机不需要具有此类装置。而且,计算机可嵌入于另一装置中,例如移动电话、个人数字助理(PDA)、移动音频或视频播放器、游戏控制台、全球定位系统(GPS)接收器或便携式存储装置(例如,通用串行总线(USB)闪存驱动器),仅举几例。适合于存储计算机程序指令和数据的装置包括所有形式的非易失性存储器、介质和存储器装置,包括例如:示例性半导体存储器装置,例如,EPROM(可擦除可编程只读存储器)、EEPROM(电可擦除可编程只读存储器)和闪存存储器装置;磁盘,例如,内部硬盘或可移除盘;磁光盘;以及CD-ROM和DVD-ROM盘。可通过专用逻辑电路系统来补充处理器和存储器,或者处理器和存储器可并入专用逻辑电路系统中。

[0092] 为了提供与用户的交互,本说明书中所描述的标的物的实施例可以在计算机上实施,所述计算机具有:显示装置,例如,LCD(液晶显示器)显示装置、OLED(有机发光二极管)显示装置或用于向用户显示信息的另一监视器;以及键盘和指向装置,例如,鼠标或轨迹球,用户可以借此向计算机提供输入。还可使用其它种类的装置来提供与用户的交互;例如,向用户提供的反馈可以是任何适合形式的感官反馈,例如,视觉反馈、听觉反馈或触觉反馈;并且可接收来自用户的呈任何适合形式的输入,包括声学、语音或触觉输入。

[0093] 计算系统可包括客户端和服务端。客户端和服务端一般彼此远离并且通常通过通信网络进行交互。客户端和服务端的关系凭借在相应的计算机上运行并且彼此具有客户端-服务端关系的计算机程序而产生。本说明书中描述的主题的实施方案可以在计算系统中实施,所述计算系统包括:后端部件,例如,作为数据服务器;或包括中间件部件,例如,应用服务器;或包括前端部件,例如,具有图形用户界面或浏览器用户界面的客户端计算机,用户可以借此与本说明书中描述的主题的实施方式交互;或一个或多个此类后端部件、中间件部件或前端部件的任何组合。系统的部件可以通过任何合适形式或介质的数字数据通

信(例如,通信网络)互连。通信网络的示例包括局域网(“LAN”)和广域网(“WAN”)、互联网络(例如,互联网)和对等网络(例如,专用对等网络)。

[0094] 虽然本说明书含有许多实施细节,但这些不应被解释为对所要求保护的范围或可能要求保护的范围的限制,而是应解释为对所公开标的物的特定实施例特有的特征的描述。本说明书中在单独实施例的上下文中描述的某些特征也可在单个实施例中以组合方式实施。相反地,在单个实施例的上下文中描述的各种特征也可单独地在多个实施例中实施或以任何适合子组合方式实施。而且,尽管特征在上文可被描述为以某些组合方式起作用并且甚至最初如此要求受到保护,但是来自要求保护的组合的一个或多个特征在一些情形中可与所述组合分离,并且要求保护的组合可涉及子组合或子组合的变体。

[0095] 类似地,虽然在图式中以特定次序描绘操作,但是这不应被理解为要求此类操作以所展示特定次序或以连续次序执行,或要求执行所有所图解说明操作,以实现所期望结果。在某些情况中,多任务和并行处理可能是有利的。此外,上文所描述的实施例中的各种系统组件的分离不应被理解为在所有实施例中都需要此种分离,并且应理解,所描述程序组件和系统可大体上一起集成在单个软件产品中或封装到多个软件产品中。

[0096] 因此,已经描述本发明的特定实施例。其它实施例在所附权利要求书的范围内。此外,权利要求中记载的动作可按不同的次序执行并且仍然实现期望的结果。

[0097] 实例:虽然本申请在所附权利要求中受到限定,但应当理解,本发明还(附加地或替代地)可根据以下实例来限定:

[0098] 实例1.一种方法,所述方法包括:通过调度计算机程序获得描述待重新调度的一个或多个项目的调度的数据集,所述数据集包括待调度的任务的工作分解结构、资源需求以及所述任务之间的相依性;通过所述调度计算机程序生成所述调度的变型,其中所述调度的所述变型中的每一个具有不同特性,所述不同特性针对所述调度中的所述任务确定何时及时地调度每个任务,并且所述变型中的每一个都满足所述资源需求和所述相依性;通过所述调度计算机程序在所述调度的所述变型中针对所述任务的所述不同特性当中进行选择以形成所述一个或多个项目的经修订调度;以及通过所述调度计算机程序提供所述一个或多个项目的所述经修订调度以供显示给用户或者供输出以管理所述一个或多个项目。

[0099] 实例2.根据实例1所述的方法,其中所述数据集除了所述相依性之外还包括一个或多个调度约束,并且所述变型中的每一个除满足所述相依性之外还满足所述一个或多个调度约束。

[0100] 实例3.根据实例1至2中任一项所述的方法,其中所述生成和所述选择由进化人工智能算法执行。

[0101] 实例4.根据实例1至3中任一项所述的方法,其中所述一个或多个项目为共享至少一个资源类别的两个或更多个项目,所述一个或多个调度约束包括针对所述两个或更多个项目中的每一个的至少一个时间约束,并且所述生成和所述选择通过使所述至少一个资源类别的利用率最大化同时还满足针对所述两个或更多个项目中的每一个的所述至少一个时间约束来形成所述经修订调度。

[0102] 实例5.根据实例1至4中任一项所述的方法,所述方法包括对所述数据集采用层状数据模型,其中所述层状数据模型包括:第一层,其指定表示所述调度的图的拓扑;第二层,其至少指定对所述第一层做出局部改变的编辑操作;以及第三层,其指定表示所述调度的

所述图,包括所述调度中的所述任务之间的所述相依性的所有细节以及针对所述调度中的所述任务的所有经调度开始和结束时间。

[0103] 实例6.根据实例5所述的方法,其中所述调度计算机程序至少在远离由所述用户操作的客户端计算机的服务器计算机上运行,所述层状数据模型的所述第一层完全加载到所述客户端计算机的存储器中,并且响应于编辑操作对所述第二层的更新是在所述客户端计算机本地并发执行的并持久化到所述服务器计算机。

[0104] 实例7.根据实例1至6中任一项所述的方法,所述方法包括:在图形用户界面中呈现所选择的资源类别的当前经规划使用率和所选择的资源类别的计算出的理想使用率;以及经由所述图形用户界面接收用户输入,所述用户输入改变所选择的资源类别的所述计算出的理想使用率的形状,以创建所选择的资源类别的用户指定的理想使用率;其中所述生成和所述选择通过以下来形成所述经修订调度:基于针对工作分布的目标函数来修改所选择的资源类别的所述当前经规划使用率以近似所述用户指定的理想使用率,所述目标函数被表达为所选择的资源类别的利用率与所选择的资源类别的所述用户指定的理想使用率的偏差。

[0105] 实例8.根据实例7所述的方法,其中所述图形用户界面在第一轴上示出时间并且在第二轴上示出工作负荷,并且所述用户输入改变使用所选择的资源类别的开始日期、使用所选择的资源类别的结束日期、所述工作负荷的针对所选择的资源类别的斜升曲线、所述工作负荷的针对所选择的资源类别的斜降曲线或者它们的组合。

[0106] 实例9.根据实例7至8中任一项所述的方法,其中针对工作分布的所述目标函数使用非均匀和不连续的自变量,所述非均匀和不连续的自变量将时钟时间表示为自开始时间以来单调递增的累积可用工作时间量。

[0107] 实例10.根据实例9所述的方法,所述方法包括提供并发安全的缓式构建模式和用于评定所述时钟时间的高性能无锁只读模式两者。

[0108] 实例11.根据实例1至10中任一项所述的方法,其中所述生成和所述选择通过以下来形成所述经修订调度:基于(i)针对工作分布的第一目标函数以及(ii)针对工作分布的第二目标函数来修改所选择的资源类别的所述当前经规划使用率,所述第一目标函数被表达为指定的资源类别的当前利用率与所述指定的资源类别的理想利用率的偏差,所述第二目标函数被表达为使用所述指定的资源类别的所述任务的当前时间定序与使用所述指定的资源类别的所述任务的理想时间定序的偏差,其中所述任务的所述理想时间定序是从分配给使用所述指定的资源类别的所述任务的优先级值确定的。

[0109] 实例12.根据实例1至11中任一项所述的方法,其中所述生成和所述选择分两个阶段执行,在所述两个阶段中的初始阶段中,按项目或按子项目应用针对调度变型的相对浮动,并且在所述两个阶段的后续阶段中,按活动应用针对调度变型的相对浮动。

[0110] 实例13.一种非暂时性计算机可读介质,其对计算机辅助设计程序进行编码,所述计算机辅助设计程序可操作以使一个或多个数据处理设备执行根据实例1至12中任一项所述的操作。

[0111] 实例14.一种系统,所述系统包括:一个或多个数据处理设备;以及一个或多个非暂时性计算机可读介质,其对指令进行编码,所述指令可由所述一个或多个数据处理设备执行,以执行根据实例1至12中任一项所述的操作。

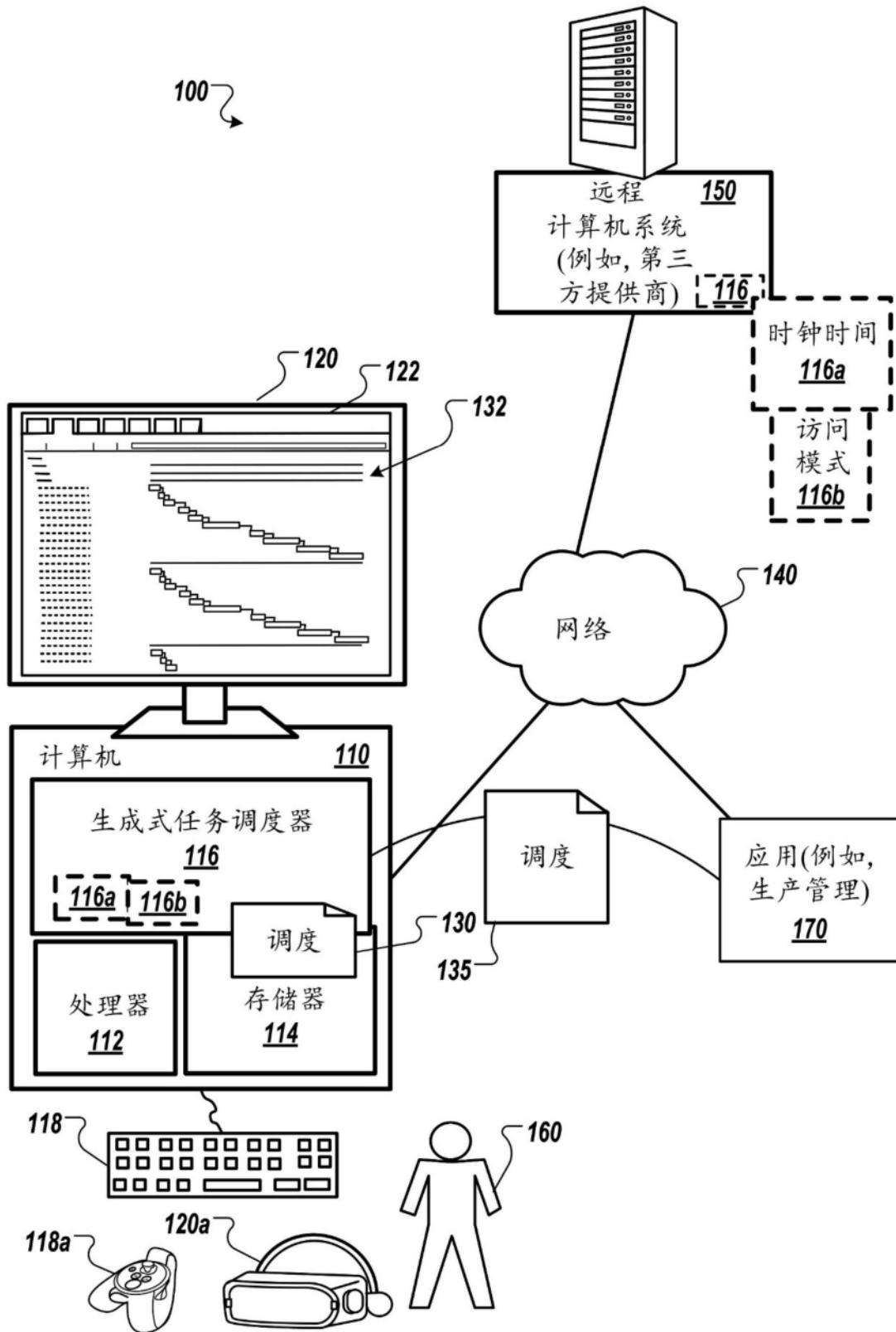


图1

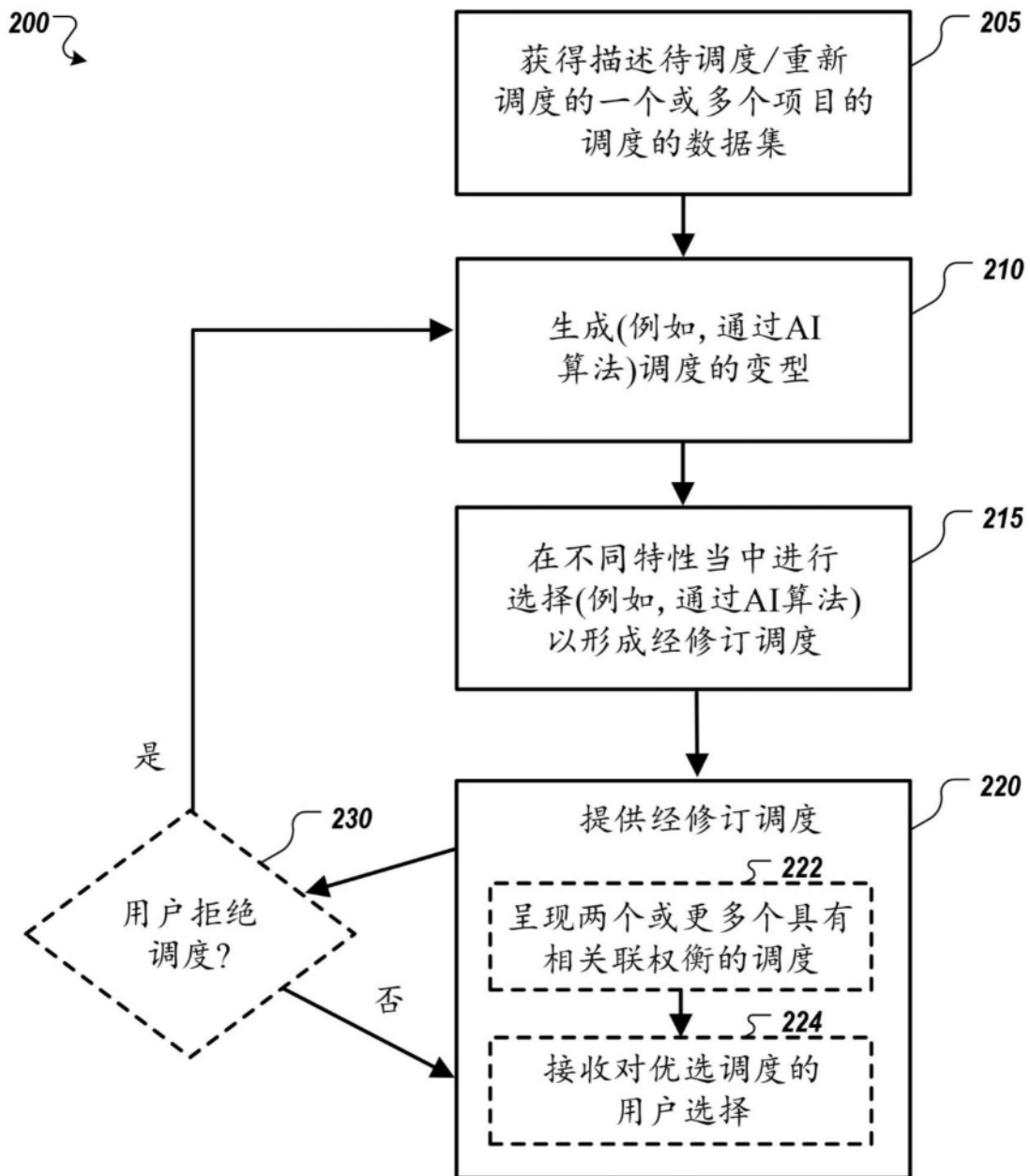


图2

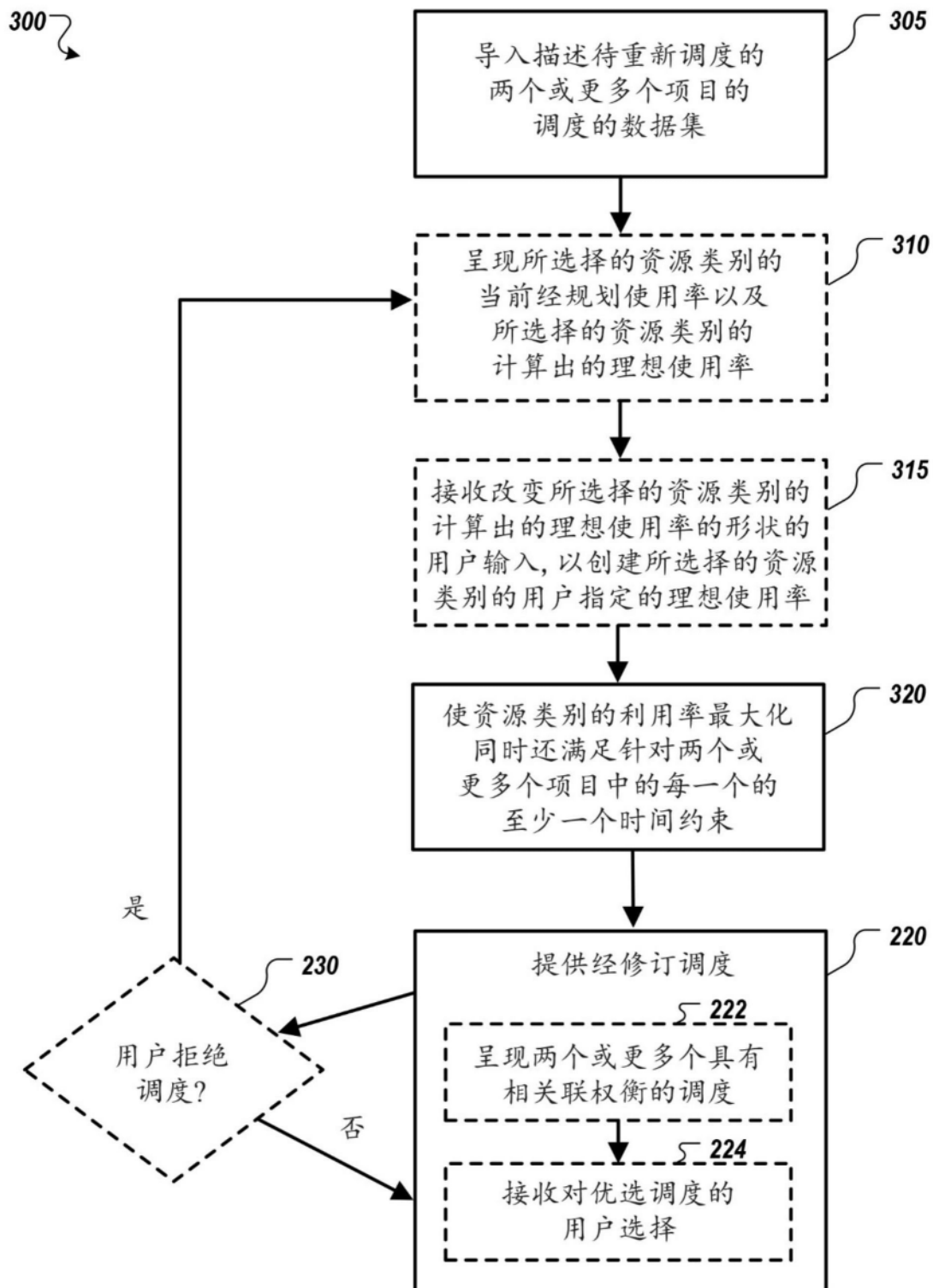


图3

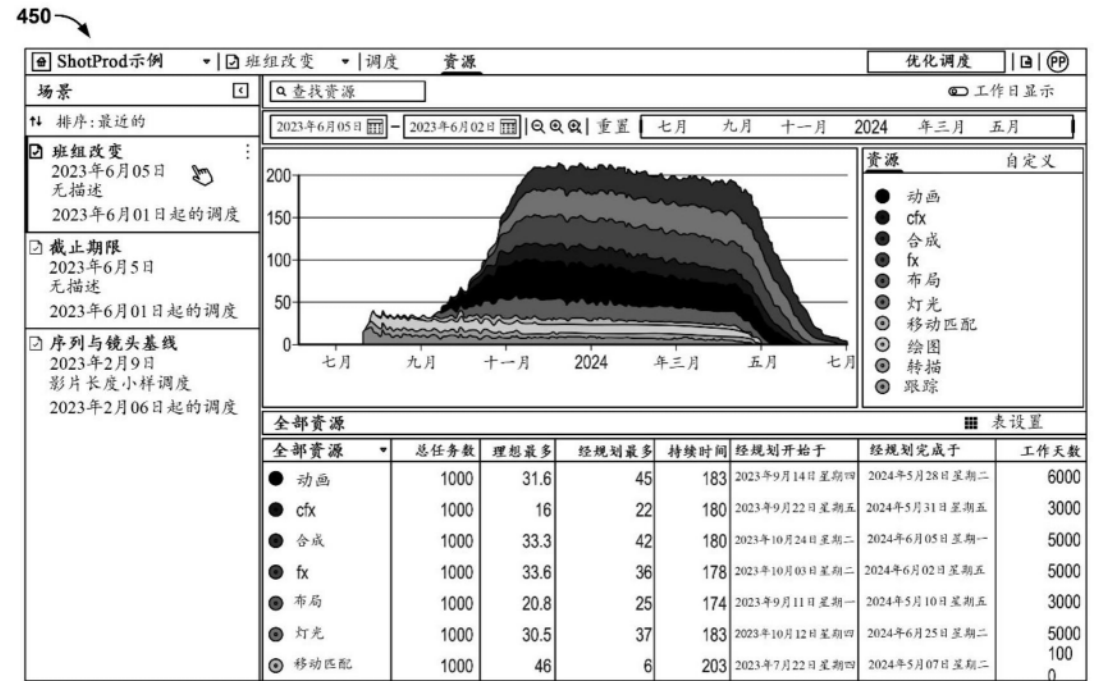


图4A

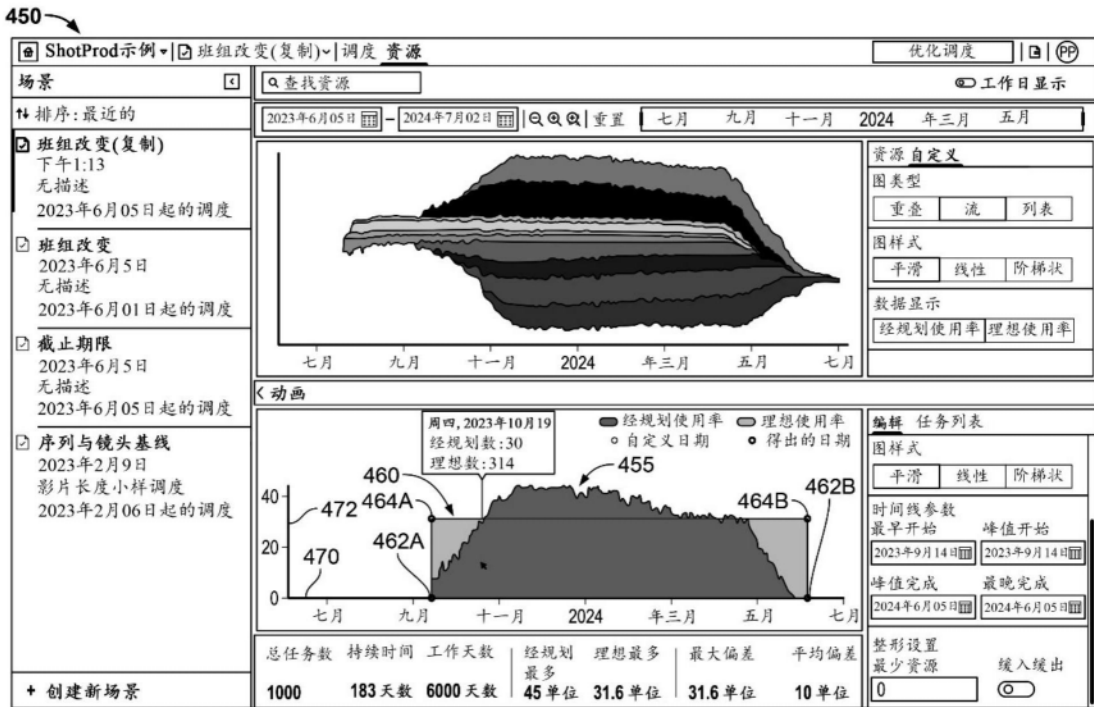


图4B

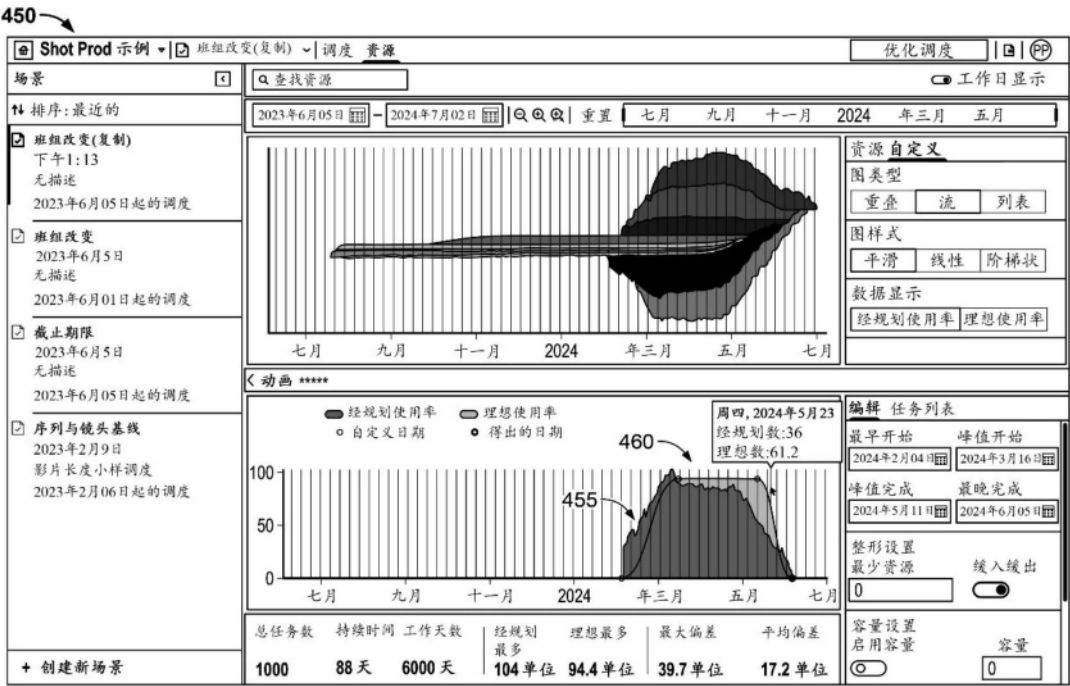


图4C

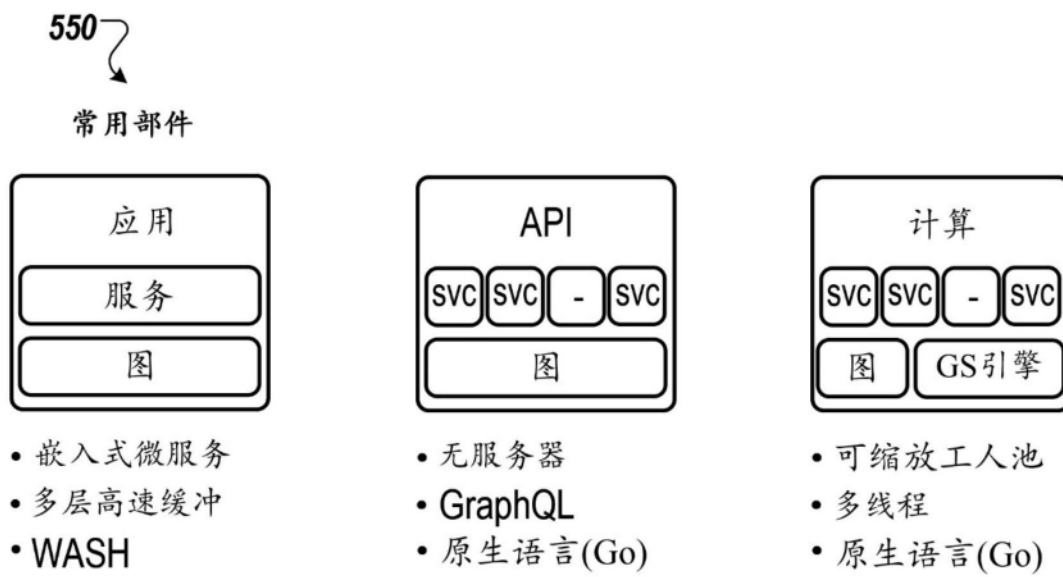
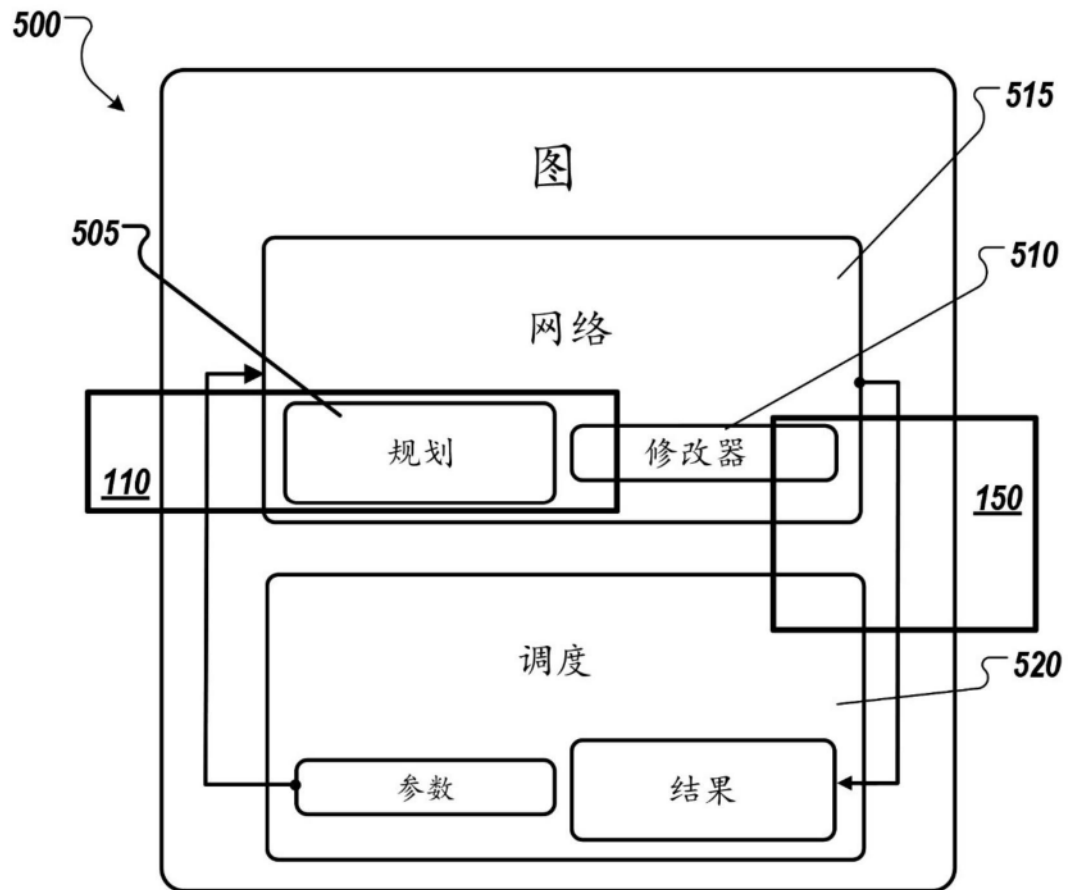


图5

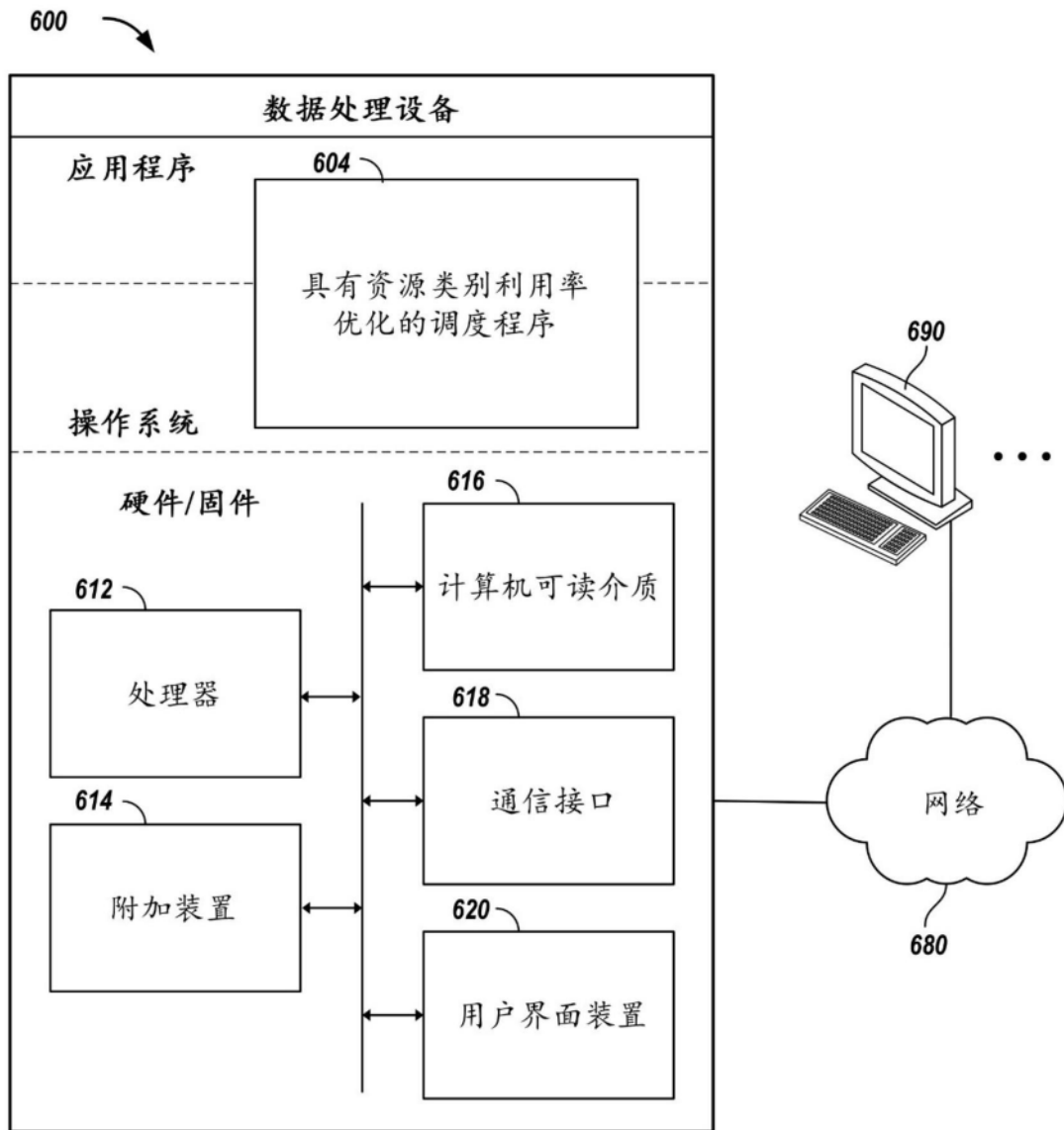


图6

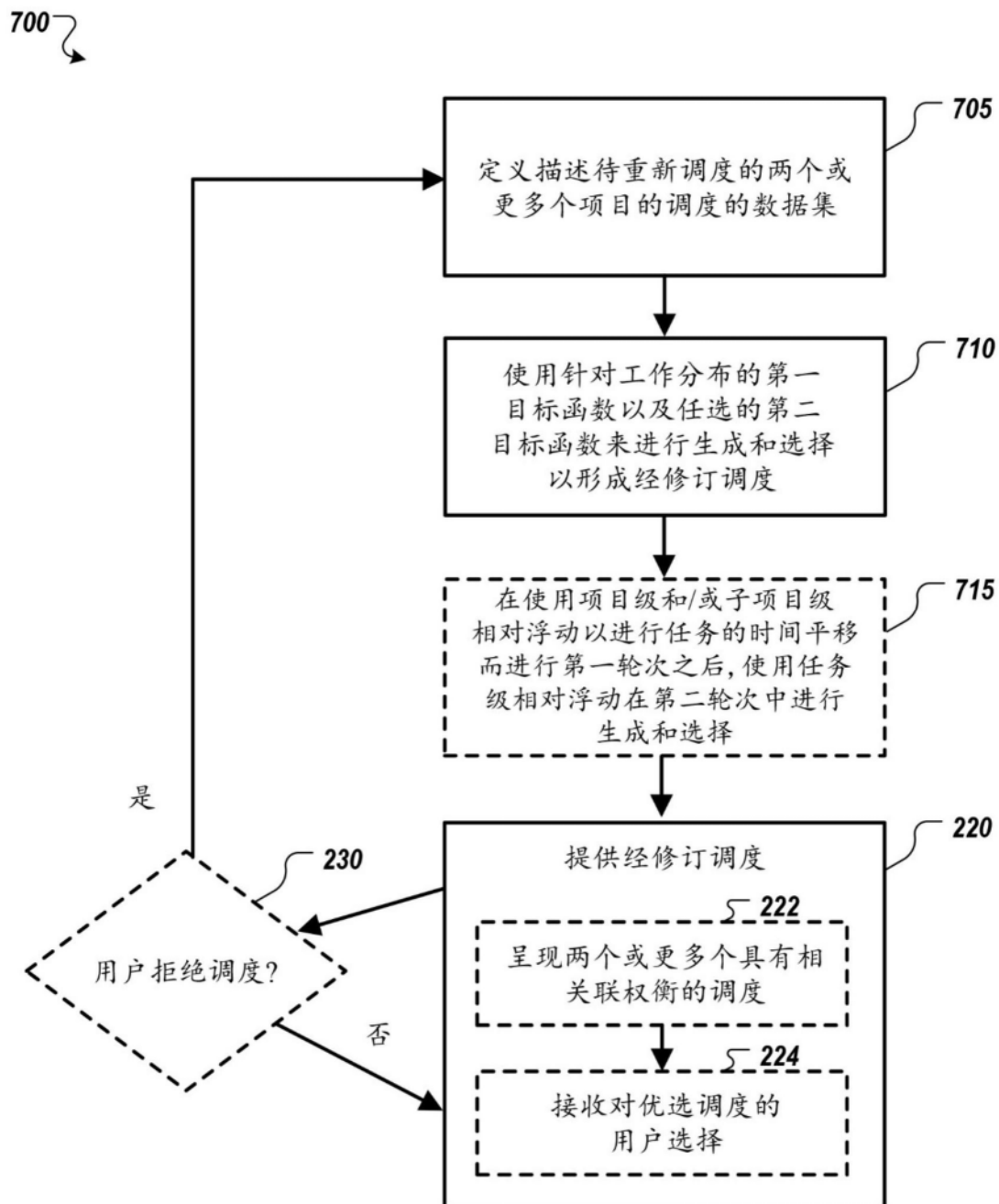


图7