



(12) 发明专利申请

(10) 申请公布号 CN 116860399 A

(43) 申请公布日 2023. 10. 10

(21) 申请号 202211666449.2

(22) 申请日 2022.12.23

(30) 优先权数据

17/705,959 2022.03.28 US

(71) 申请人 霍尼韦尔国际有限责任公司

地址 捷克霍多夫

(72) 发明人 帕维尔·扎伊科夫

拉里·詹姆斯·米勒 H·卡瓦略

斯里瓦赞·瓦拉达拉詹

(74) 专利代理机构 中国专利代理(香港)有限公

司 72001

专利代理师 吕传奇

(51) Int.Cl.

G06F 9/48 (2006.01)

G06F 9/50 (2006.01)

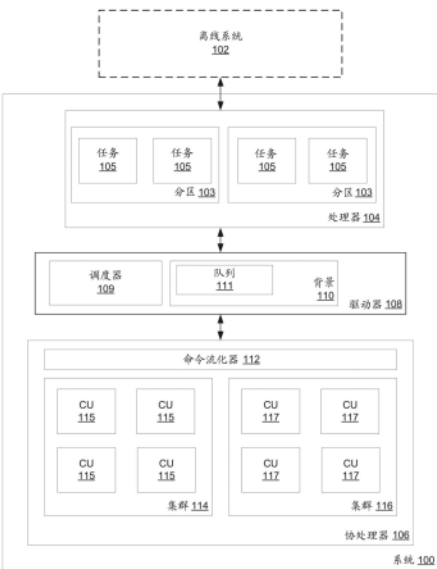
权利要求书1页 说明书28页 附图20页

(54) 发明名称

用于时间-空间分区处理系统的处理引擎调度度

(57) 摘要

本发明公开了用于处理器和至少一个协处理器之间的改进处理效率的实施方案。一些示例涉及处理器-协处理器调度,其中工作负载基于该处理器的定时窗口被调度到协处理器。在附加的或另选的示例中,工作负载基于该处理资源和/或优先级顺序被分配到该协处理器。结合所公开的实施方案,该协处理器可由图形处理单元(GPU)、硬件处理加速器、现场可编程门阵列(FPGA)、专用集成电路(ASIC)或其他处理电路实现。该处理器可由中央处理单元(CPU)或其他处理电路实现。



1. 一种处理系统,包括:

处理器;

协处理器,所述协处理器被配置为实现处理引擎;

处理引擎调度器,所述处理引擎调度器被配置为调度工作负载以用于在所述协处理器上执行;

其中所述处理引擎调度器被配置为从在所述处理器上执行的一个或多个任务接收一个或多个工作负载发动请求,并且作为响应,基于协处理器调度策略来生成用于提交到所述协处理器的至少一个发动请求;

其中基于所述协处理器调度策略,所述处理引擎调度器选择激活哪些协处理器集群以基于所述至少一个发动请求执行由队列识别的工作负载;以及

其中所述协处理器调度策略限定以下中的至少一者:

紧密耦接协处理器调度,其中调度由所述至少一个发动请求识别的工作负载以在其中所述一个或多个任务在所述处理器上执行的定时窗口的时间内立即在所述协处理器上执行;或者

紧密耦接协处理器调度,其中调度由所述至少一个发动请求识别的工作负载以基于优先级顺序并且通过以下中的任一者在所述协处理器上执行:

相对于所述处理器和所述协处理器两者共同的外部事件,或

在其中所述一个或多个任务在所述处理器上执行的所述定时窗口的所述时间之后的后续定时窗口期间。

2. 根据权利要求1所述的处理系统,其中所述协处理器调度策略包括限定耦接协处理器调度的预占策略,其中被调度用于在所述协处理器上执行的或当前在所述协处理器上执行的一个或多个工作负载被配置为由排队以基于所述优先级顺序进行执行的一个或多个工作负载预占。

3. 一种方法,包括:

从在处理器上执行的一个或多个任务接收一个或多个工作负载发动请求,其中所述一个或多个工作负载发动请求包括被配置用于在协处理器上执行的一个或多个工作负载;

基于协处理器调度策略,响应于所述一个或多个工作负载发动请求而生成至少一个发动请求;

基于所述协处理器调度策略通过以下中的至少一者来调度所述至少一个发动请求中识别的一个或多个工作负载以用于在所述协处理器上执行:

紧密耦接协处理器调度,其中调度由所述至少一个发动请求识别的工作负载以在其中所述一个或多个任务在所述处理器上执行的定时窗口的时间内立即在所述协处理器上执行;或者

紧密耦接协处理器调度,其中调度由所述至少一个发动请求识别的工作负载以基于优先级顺序并且通过以下中的任一者在所述协处理器上执行:

相对于所述处理器和所述协处理器两者共同的外部事件,或

在其中所述一个或多个任务在所述处理器上执行的所述定时窗口的所述时间之后的后续定时窗口期间。

用于时间-空间分区处理系统的处理引擎调度

[0001] 关于非美国赞助研究或开发的声明

[0002] 产生该申请的项目已根据拨款协议第945535号从欧盟地平线2020研究和创新计划下的清洁天空2联合执行体获得资金。

背景技术

[0003] 动态环境中的实时处理需要在非常短的时间框架内处理大量数据。取决于特定背景,此类处理可涉及计算迭代数学计算或执行密集数据分析。快速和准确的数据输出对于避免处理延迟而言是重要的,避免处理延迟对于安全关键或任务关键的应用程序(诸如在航空电子使用的那些应用程序)而言尤其必要。

[0004] 一些实时操作系统利用时间和/或空间分区过程来处理数据。最初,根据来自应用程序的指令在主处理器(本文称为“中央处理单元”或“CPU”)处执行任务。CPU通常负责引导任务的执行以及在CPU执行任务时管理数据输出。针对在CPU处接收的任务的原始数据处理的大部分由与CPU不同的协处理器执行。当CPU执行任务时,它可将与任务相关联的工作负载分配给协处理器以用于处理。“工作负载”在本文中也称为具体应用程序的“作业”、“内核”或“着色器”。由CPU执行的任务可能需要可在协处理器上更快速地执行的处理,因此CPU可发送限定协处理器必须执行以完成由CPU执行的任务的工作负载的一个或多个请求。这些请求在本文中被称为“工作负载发动请求”。

[0005] 协处理器通常接收许多此类请求,有时在短时间段内。每个请求可涉及非常大量的密集计算。以及时方式处理工作负载发动请求的能力不仅取决于协处理器的处理能力,而且还取决于如何利用协处理器以执行由主处理器请求的工作。虽然具有强大处理资源的协处理器可快速处理这些请求,但它们可能实现起来很昂贵并且不保证协处理器能够在短时间框架中处理具有大量处理要求的任务。具有有限处理资源的较不高级的协处理器倾向于处理与用于处理附加请求的不足带宽相关联的延迟并且可导致确定性的损失的损失。在任何情况下,协处理器通过备份工作负载发动请求变得不堪重负。

[0006] 一些协处理器实现其处理资源的时间和/或空间分区,使得可并行执行多个作业。然而,常规协处理器未提供足够的空间隔离、时间确定性和响应性以同时执行多个安全关键应用程序。无法及时处理安全关键应用程序可最终导致确定性的损失的损失。

发明内容

[0007] 以下描述中阐述了一个或多个实施方案的细节。结合一个示例性实施方案示出或描述的特征部可与其它实施方案的特征部进行组合。因此,可组合本文所述的各种实施方案中的任一种以提供另外的实施方案。如有必要,可修改实施方案的各方面,以采用如本文所识别的各种专利、应用和出版物的概念,以提供又另外的实施方案。

[0008] 在一个实施方案中,公开了一种处理系统。该处理系统包括处理器和协处理器,该协处理器被配置为实现处理引擎。该处理系统还包括处理引擎调度器,该处理引擎调度器被配置为调度工作负载以用于在该协处理器上执行。该处理引擎调度器被配置为从在该处

理器上执行的一个或多个任务接收一个或多个工作负载发动请求。作为响应,该处理引擎调度器被配置为基于协处理器调度策略来生成用于提交到该协处理器的至少一个发动请求。基于该协处理器调度策略,该处理引擎调度器选择激活哪些协处理器集群以基于该至少一个发动请求执行由队列识别的工作负载。该协处理器调度策略限定以下中的至少一者:紧密耦接协处理器调度,其中调度由该至少一个发动请求识别的工作负载以在其中该一个或多个任务在该处理器上执行的定时窗口的时间内立即在该协处理器上执行;或紧密耦接协处理器调度,其中调度由该至少一个发动请求识别的工作负载以基于优先级顺序并且通过以下中的任一者在该协处理器上执行:相对于该处理器和该协处理器两者共同的外部事件,或在其中该一个或多个任务在该处理器上执行的该定时窗口的该时间之后的后续定时窗口期间。

[0009] 在另一个实施方案中,公开了一种协处理器。该协处理器被配置为耦接到处理器并且被配置为实现处理引擎。该协处理器包括被配置为执行工作负载的至少一个集群。该协处理器包括处理引擎调度器,该处理引擎调度器被配置为调度工作负载以用于在该协处理器上执行。该处理引擎调度器被配置为从在该处理器上执行的一个或多个任务接收一个或多个工作负载发动请求。作为响应,该处理引擎调度器被配置为基于协处理器调度策略来生成用于提交的至少一个发动请求。基于该协处理器调度策略,该处理引擎调度器被配置为选择激活该至少一个集群中的哪一者以执行由包括该至少一个发动请求的队列识别的工作负载。该协处理器调度策略限定以下中的至少一者:紧密耦接协处理器调度,其中调度由该至少一个发动请求识别的工作负载以在其中该一个或多个任务在该处理器上执行的定时窗口的时间内立即在该协处理器上执行;或紧密耦接协处理器调度,其中调度由该至少一个发动请求识别的工作负载以基于优先级顺序并且通过以下中的任一者在该协处理器上执行:相对于该处理器和该协处理器两者共同的外部事件,或在其中该一个或多个任务在该处理器上执行的该定时窗口的该时间之后的后续定时窗口期间。

[0010] 在另一个实施方案中,公开了一种方法。该方法包括从在处理器上执行的一个或多个任务接收一个或多个工作负载发动请求。该一个或多个工作负载发动请求包括被配置用于在协处理器上执行的一个或多个工作负载。该方法包括基于协处理器调度策略,响应于该一个或多个工作负载发动请求而生成至少一个发动请求。该方法包括基于该协处理器调度策略通过以下中的至少一者来调度该至少一个发动请求中识别的一个或多个工作负载以用于在该协处理器上执行:紧密耦接协处理器调度,其中调度由该至少一个发动请求识别的工作负载以在其中该一个或多个任务在该处理器上执行的定时窗口的时间内立即在该协处理器上执行;或紧密耦接协处理器调度,其中调度由该至少一个发动请求识别的工作负载以基于优先级顺序并且通过以下中的任一者在该协处理器上执行:相对于该处理器和该协处理器两者共同的外部事件,或在其中该一个或多个任务在该处理器上执行的该定时窗口的该时间之后的后续定时窗口期间。

[0011] 在另一个实施方案中,公开了一种处理系统。该处理系统包括处理器和协处理器,该协处理器被配置为实现处理引擎。该处理系统包括处理引擎调度器,该处理引擎调度器被配置为调度工作负载以用于在该协处理器上执行。该处理引擎调度器被配置为从在该处理器上正执行或已执行的一个或多个任务接收一个或多个工作负载发动请求。作为响应,该处理引擎调度器被配置为生成用于提交到该协处理器的至少一个发动请求。该协处理器

包括多个计算单元和与该多个计算单元中的一个或多个计算单元相关联的至少一个命令流化器。基于协处理器分配策略,该处理引擎调度器被配置为经由该至少一个命令流化器针对给定执行分区分配该协处理器的计算单元的集群,以根据工作负载优先级执行由该一个或多个工作负载发动请求识别的一个或多个工作负载。该协处理器分配策略至少限定:排他性分配策略,其中每个工作负载由计算单元的专用集群执行;交错分配策略,其中每个工作负载跨该计算单元集群的所有计算单元专门执行;策略-分布式分配策略,其中每个工作负载在该给定执行分区期间被单独地分配到该计算单元的集群中的至少一个集群和执行持续时间;或共享分配策略,其中每个工作负载不专门由该计算单元的集群执行,该计算单元的集群各自同时执行多个工作负载。

[0012] 在另一个实施方案中,公开了一种协处理器。该协处理器被配置为耦接到处理器并且被配置为实现处理引擎。该协处理器包括各自被配置为执行工作负载的多个计算单元。该协处理器包括处理引擎调度器,该处理引擎调度器被配置为分配工作负载以用于在该协处理器上执行。该处理引擎调度器被配置为从在该处理器上正执行或已执行的一个或多个任务接收一个或多个工作负载发动请求。作为响应,该处理引擎调度器被配置为生成用于提交到该协处理器的至少一个发动请求。该协处理器包括与该多个计算单元中的一个或多个计算单元相关联的至少一个命令流化器。基于协处理器分配策略,该处理引擎调度器被配置为经由该至少一个命令流化器针对给定执行分区分配该协处理器的计算单元的集群,以根据工作负载优先级执行由该一个或多个工作负载发动请求识别的一个或多个工作负载。该协处理器分配策略至少限定:排他性策略,其中每个工作负载由该计算单元的集群中的专用集群执行;交错策略,其中每个工作负载跨该计算单元的集群中的至少一个集群的所有计算单元专门执行;策略-分布式策略,其中每个工作负载在该给定执行分区期间被单独地分配到该计算单元的集群中的至少一个集群和执行持续时间;共享策略,其中每个工作负载不专门由该计算单元的集群执行,该计算单元的集群各自同时执行多个工作负载。

[0013] 在另一个实施方案中,公开了一种方法。该方法包括从在处理器上正执行或已执行的一个或多个任务接收一个或多个工作负载发动请求。该一个或多个工作负载发动请求包括被配置用于在协处理器上执行的一个或多个工作负载。该方法包括响应于该一个或多个工作负载发动请求而生成至少一个发动请求。该方法包括分配该协处理器的计算单元的集群,以基于协处理器分配策略根据工作负载优先级执行该一个或多个工作负载发动请求中识别的一个或多个工作负载。该协处理器分配策略至少限定:排他性策略,其中每个工作负载由该计算单元的集群中的专用集群执行;交错策略,其中每个工作负载跨该计算单元的集群中的至少一个集群的所有计算单元专门执行;策略-分布式策略,其中每个工作负载在给定执行分区期间被单独地分配到该计算单元的集群中的至少一个集群和执行持续时间;共享策略,其中每个工作负载不专门由该计算单元的集群执行,该计算单元的集群各自同时执行多个工作负载。

附图说明

[0014] 应当理解,附图仅示出了示例性实施方案,因此不应认为是限制本发明的范围,将通过附图的使用以附加特征和细节来描述示例性实施方案,其中:

[0015] 图1A至图1B描绘了示出被配置为将发动请求调度和分配到协处理器的示例性系统的框图；

[0016] 图2A至图2B描绘了将与一个或多个队列相关联的工作负载调度和分配到协处理器的集群的框图；

[0017] 图3描绘了根据一个实施方案的在松散耦接协处理器方案中将工作负载调度到协处理器的多个集群的图；

[0018] 图4A至图4B描绘了在紧密耦接协处理器方案中将工作负载调度到协处理器的多个集群的图；

[0019] 图5A至图5B描绘了在CPU与协处理器的一个或多个集群之间同步推断处理的图；

[0020] 图6A至图6B描绘了应用于被调度到图形处理单元 (GPU) 的工作负载的预占策略的图；

[0021] 图7描绘了示出用于将用于执行的工作负载调度到协处理器的示例性方法的流程图；

[0022] 图8A至图8B描绘了协处理器的多个集群之间的数据耦接的图；

[0023] 图9描绘了根据一个实施方案的应用于协处理器的多个集群的协处理器分配策略的图；

[0024] 图10A至图10C描绘了示出被配置为将工作负载分配给协处理器的多个集群的示例性系统的框图；

[0025] 图11描绘了示出用于将工作负载分配到协处理器的处理资源的示例性方法的流程图；

[0026] 图12描绘了示出用于在协处理器上执行工作负载时管理处理资源的示例性方法的流程图；

[0027] 图13描绘了示出用于在背景中优先化工作负载的示例性方法的流程图；并且

[0028] 图14A至图14C描绘了示出用于调度和分配工作负载的示例性方法的流程图。

[0029] 根据惯例，所描述的各种特征未必按比例绘制，而是用于强调与示例性实施方案相关的特定特征。

具体实施方式

[0030] 在以下详细描述中，参考形成其一部分的附图，并且其中通过图示方式示出了特定的例示性实施方案。然而，应当理解可能利用其他实施方案，并且可能进行逻辑、机械和电气更改。此外，附图和说明书中呈现的方法不应被解释为限制可执行各个步骤的顺序。因此，以下详细描述不应被视为具有限制意义。

[0031] 本公开的实施方案提供了对于将工作负载调度和分配到协处理器 (例如，GPU) 以用于执行的改进。本文公开的一些实施方案使得能够基于CPU的定时窗口将工作负载调度到GPU，使得GPU至少部分地与CPU同步。其他实施方案使得GPU能够动态地分配工作负载以优化GPU上的处理资源的使用。除非另有说明，否则工作负载可在本文中以单数“工作负载 (workload)”或复数“工作负载 (workloads)”教学地引用，应当理解该描述适用于单个工作负载或多个工作负载。

[0032] 虽然示出和描述了一些示例以用于专门地将工作负载调度和分配到GPU，但本文

描述的示例也在其他系统的背景中适用。例如,此类技术也适用于具有将工作负载调度和分配到一个或多个协处理器的一个或多个处理器的任何处理系统。协处理器通常可在处理系统中实现为集成或分立处理单元。在各种示例中,协处理器可被实现为图形处理单元(“GPU”)、神经处理单元(“NPU”)、数据处理单元(“DPU”)、专用集成电路(ASIC)、现场可编程门阵列(FPGA)和其他处理电路,或它们的组合。

[0033] 协处理器可使用传统执行或通过人工智能促进执行来加速工作负载处理。对于基于AI的建模,协处理器用于加速与机器学习(ML)/人工智能(AI)应用程序相关联的一些工作负载的执行。另外,协处理器可用于通过推断引擎加速ML/AI应用程序的执行,该推断引擎可用于例如深度神经网络(DNN)处理。在以下各种附图和描述中,协处理器被实现为GPU以用于教学解释。

[0034] 图1A描绘了示出被配置为将工作负载调度和分配到协处理器的示例性系统100的框图。在一些示例中,系统100实现实时操作系统(RTOS),当数据在指定时间约束内进入时,该RTOS有助于执行实时应用程序以对该数据进行处理。系统100包括耦接到一个或多个协处理器106的处理器104。在图1中明确示出了仅一个处理器104和一个协处理器106,但应当理解,任何数量的处理器104可耦接到一个或多个协处理器106。

[0035] 处理器104被配置为从离线系统102(例如,从存储的系统配置)接收系统参数,包括协处理器调度策略和协处理器分配策略,该协处理器调度策略确定何时将工作负载分配到协处理器106,该协处理器分配策略确定在何处将工作负载分配到协处理器106的处理资源。处理器104还被配置为执行从在处理器104(图1A中未示出)的处理资源(处理器、处理电路)上运行的一个或多个应用程序(安全关键应用程序、最佳努力应用程序等)接收的任务105。在处理器104上执行的任务105可能需要利用协处理器106上的处理资源的处理。在一个示例中,处理器被配置为当执行给定任务105时准备一个或多个工作负载发动请求,其中工作负载需要通过协处理器106的数据处理(例如,包括数学计算(诸如矩阵运算等)的DNN处理)。任务105也可能需要其他类型的处理,包括但不限于渲染和/或计算处理。在各种示例中,工作负载被表示为“内核”、“作业”、“线程”、“着色器”或其他处理单元。当任务105由处理器104处理时,协处理器106还处理工作负载发动请求以与处理器104并行地发动和执行与任务105相关联的工作负载。工作负载发动请求包括处理所需的工作负载(例如,内核)的序列,以及其他工作负载参数,诸如输入和输出数据阵列、工作负载代码、完成工作负载所必需的处理负载、包括在发动请求中的工作负载的优先级和/或其他参数。

[0036] 处理器104可包括一个或多个分区103。每个分区103用作独立处理系统(例如,如图1B所示的处理核心103)并且被配置为执行任务105中的一者或多者。分区103可在时间和/或空间中进行分区。时间和空间分区可通过分离处理器104中的硬件电路的常规物理(“硬”)分区技术或通过设置由处理器104执行的处理约束的软件方法(例如,虚拟化技术)来实现。当经由软件进行空间分区时,分区103不会污染另一个分区103的代码、输入/输出(I/O)或数据的存储区域,并且每个分区的消耗不超过其相应的共享处理资源分派。此外,可归因于一个软件分区的硬件的故障不会不利地影响软件分区的性能。当由于减轻在不同核心上托管的分区分之间的时间干扰,因此没有软件分区的消耗超过该软件分区在其上执行的该核心上的该软件分区的执行时间分派时,实现了时间分区,而不管分区不在其他活动核心中的任一者上执行还是在所有其他活动核心上执行。对于分区处理器(诸如图1所

示的处理器104),任务105中的一者或多者在第一分区103上执行,由此实现针对分配到不同分区103的任务105中的至少一些的同时处理。处理器104可包括任何数量的分区103。在一些示例中,分区103彼此物理或虚拟地隔离以防止从一个分区到另一个分区的错误传播。

[0037] 耦接到一个或多个处理器104的每个协处理器106被配置为接收由处理器104卸载的处理中的至少一些。系统100包括驱动器108,该驱动器包括处理引擎调度器(也称为“调度器”)109和一个或多个背景110。背景110包括被配置为提供空间隔离的硬件。多个背景110使得能够并行地执行协处理器上的多个分区以支持时间和/或空间分区。

[0038] 对于诸如神经网络的人工智能处理模型,调度器109可以是利用推断处理来调度工作负载以用于在协处理器106上执行的推断引擎调度器。驱动器108和协处理器106可利用多种类型的处理,包括计算和渲染。在系统100是RTOS的一个示例中,驱动器108驻留在处理器104中并且基于协处理器106的处理资源来调度工作负载以用于执行。在另一个示例中,驱动器108由专门由服务器应用程序访问的软件实现,一个或多个客户端应用程序向该服务器应用程序提交工作负载。服务器通常将保留对驱动器108的专有访问,并且当其从处理器104上执行的任务105接收到工作负载发动请求时,利用驱动器108以调度协处理器106上的工作负载。如图1A所示,驱动器108被实现为独立单元,但在其他示例中(诸如先前描述的那些),驱动器108由协处理器106或处理器104实现或以其他方式为其部分。

[0039] 驱动器108的调度器109被配置为将与由处理器104执行的任务105相关联的工作负载派遣到计算单元115、117,并且在一些示例中,基于处理器104的定时窗口来派遣工作负载。调度器109被配置为从处理器104接收工作负载发动请求并且调度工作负载以用于在协处理器106上执行。在一些示例中,调度器109被配置为基于调度策略根据工作负载发动请求生成至少一个发动请求。相对于图2至图7进一步描述了调度策略的一些示例。

[0040] 对于由调度器109生成的每个发动请求,一个或多个背景110包括将被调度和分配到协处理器的处理资源以用于执行的工作负载。背景110还包括对从一个或多个发动请求识别的工作负载进行分类的一个或多个队列111。每个队列111中的发动请求可基于队列111相对于由背景110组织的其他队列的优先级来按顺序排队和调度或分配。在一些示例中,队列111存储在列出每个队列的优先级的运行列表中。而且,驱动器108可包括任何数量的背景110,并且每个背景110可包括任何数量的队列111。在一些示例中,不同队列中的工作负载发动请求可并行地或以不同的顺序执行,条件是队列中的工作负载在处理期间彼此隔离。

[0041] 协处理器106还包括一个或多个命令流化器112,其被配置为根据协处理器调度策略和协处理器分配策略将由发动请求识别的工作负载调度和分配到可用集群114和/或116。协处理器106可包括任何数量的命令流化器112,并且在一些示例中,一个或多个命令流化器112在队列111之间共享和/或由专用背景110托管。每个集群114、116包括被配置为执行数据处理的相应计算单元115、117的集合。在一些示例中,集群114、116被静态配置(例如,硬连线)在协处理器106中,其中计算单元115永久地与集群114相关联并且计算单元117永久地与集群116相关联。集群114、116被配置为在队列由命令流化器112分配到相应集群时执行与和每个队列111相关联的工作负载中的一者或多者相关联的处理。

[0042] 如本文所用,“计算单元”是指集群的处理资源。每个计算单元115、117可包括如呈现给调度器109的用于执行工作负载的一个处理核心(以其他方式被称为“单核心处理单

元”)或多个处理核心(以其他方式被称为“多核心处理单元”)。核心可以是物理核心或虚拟核心。物理核心包括形成物理地处理已分配工作负载的核心的硬件(例如,处理电路)。然而,也可将虚拟核心呈现给调度器109以用于处理工作负载,其中每个虚拟核心使用基础物理核心来实现。

[0043] 处理器104和协处理器106通常包括处理器、微处理器、数字信号处理器、专用集成电路、现场可编程门阵列和/或它们的其他类似变型的组合。处理器104和协处理器106还可包括用于执行本文所述的方法中使用的各种过程任务、计算和控制功能的软件程序、固件或其他计算机可读指令或以它们起作用。这些指令通常被有形地体现在用于存储计算机可读指令或数据结构的任何存储介质(或计算机可读介质)上。

[0044] 来自工作负载执行的数据以及其他信息可存储在存储器(图1中未示出)中。存储器可包括可由通用或专用计算机或处理器或任何可编程逻辑设备访问的任何可用存储介质(或计算机可读介质)。合适的计算机可读介质可包含存储介质或存储器介质,诸如半导体、磁介质和/或光学介质,并且可体现为将指令存储在非暂态计算机可读介质,诸如高速缓存、随机存取存储器(RAM)、只读存储器(ROM)、非易失性RAM、电可擦除可编程ROM、闪存存储器或其他存储介质中。例如,在当前由集群114、116执行的工作负载被较高优先级工作负载预占的示例中,在预占之前的执行的进展被存储在存储器中并且在稍后时间段期间(例如,当预占工作负载完成执行时)可访问以用于由命令流化器112重新调度。另外,存储器被配置为存储包括由处理器104执行的任务的应用程序。

[0045] 图1B描绘了图1A中描述的系统的特定示例的示例性实施方案。图1B中使用的相同附图标记是指图1A中使用的部件的具体示例。在图1B中,系统100包括耦接到与GPU 106通信的GPU驱动器108的CPU 104。CPU 104被配置为处理器104,GPU驱动器108被配置为驱动器108,并且GPU 106被配置为协处理器106,如图1A所述。CPU 104、GPU驱动器108和GPU 106的所有其他部件类似于图1A所述的那样起作用。

[0046] 图2A至图2B示出了根据本文所述并且如图1A至图1B所述的调度和/或分配技术将队列分配到协处理器的集群。图2A示出了时间(时间分区)隔离耦接配置中的工作负载分配。相比之下,图2B示出了空间隔离耦接配置中的工作负载分配。

[0047] 参考图2A,一个或多个背景201-A包括队列202和队列203,其中每个队列包括待由协处理器处理的一个或多个工作负载(工作)。更具体地,队列202包括工作负载202-A和工作负载202-B,而队列203包括工作负载203-A和工作负载203-B。在图2A所示的时间分配的一个示例中,仅一个队列可在给定时间执行以实现队列202与队列203之间的时间隔离。因此,队列202和队列203按顺序执行。在一些示例中,基于优先级顺序分配队列,其中队列202被给予较高优先级(并且因此首先在协处理器上执行),然后是队列203。然而,与单个队列相关联的工作负载可同时在协处理器的一个或多个可用集群204上执行。

[0048] 在图2A所示的示例中,在第一时间间隔(图2A所示的时间间隔1),工作负载202-A和工作负载202-B被分配到可用集群204以用于处理。当队列202的工作负载202-A和工作负载202-B经历由集群204进行的执行时,队列203在背景201-A中被排队在下一个。一旦与队列202相关联的所有工作负载已经完成执行(或在一些示例中,被预占),在后续时间间隔(在图2A中示出的时间间隔2),将与队列203相关联的工作负载203-A和工作负载203-B分配到可用集群204以用于执行。当背景201-A从驱动器108接收附加队列时,这种工作负载分配

被迭代地重复。

[0049] 相比之下,图2B示出了空间隔离执行系统的示例,其中多个不同队列202和203可同时执行,同时维持队列之间的足够隔离。具体地,队列202和203在隔离背景201-B中排队。该配置使得队列202的工作负载202-A和队列203的工作负载203-A能够在时间间隔1中在集群204上同时执行。类似地,在时间间隔2,当处理工作负载202-A和工作负载203-A时,队列202的工作负载202-B和队列203的工作负载203-B可被装载在集群205上。在一些示例中,集群204(或包括集群204的协处理器分区)与队列202相关联,并且集群205(或包括集群205的协处理器分区)与队列203相关联。

[0050] 如下文进一步详细描述,在一些示例中,驱动器108被配置为根据协处理器调度策略调度工作负载,使得协处理器至少部分地与处理器同步。在其他示例中,驱动器108被配置为根据协处理器分配策略将工作负载分配到协处理器的处理资源以优化协处理器上的处理资源的使用。调度策略和分配策略都可包括基于工作负载的优先级来管理工作负载的预占的策略。尽管出于教学解释分别描述,但工作负载调度、工作负载分配和工作负载预占技术可组合利用。

[0051] 协处理器调度策略

[0052] 如先前相对于图1A至图1B所述,在一些示例中,驱动器108的调度器109被配置为实现协处理器调度策略,该协处理器调度策略根据至少一个发动请求调度与任务105相关的工作负载以用于由协处理器106的集群114、116执行。调度策略限定何时将调度工作负载以用于在协处理器106中的一个或多个指定集群上执行。相对于图1A至图7描述的示例包括在协处理器(诸如GPU)上调度工作负载的各种示例性示例。

[0053] 仍然参考图1A,在一个示例中,由驱动器108的调度器109执行的调度策略任选地首先选择队列111,该队列包含将被调度到集群114、116以用于处理的相关联工作负载。选择可由将队列的优先化初始化到协处理器106的外部事件触发。一些事件包括处理器104与协处理器106之间的公共定时事件,诸如来自处理器104的更新、中断消息,或其他外部事件。在调度由协处理器106接收的工作负载之前选择队列111有助于确保在执行期间的多个队列111之间的足够隔离。在一些示例中,队列111将具有调度器109在确定要选择哪个队列111时考虑的相关联处理参数。示例性参数包括识别参数(例如,分区ID和/或背景ID)、队列和/或队列内的工作负载所需的最小和/或最大集群、先验分配的队列或工作负载的优先级、预算、预占参数和其他参数。在一个示例中,对应于工作负载的预算是指完全执行工作负载将花费的最长预期时间加安全裕度。

[0054] 在调度策略的下一个阶段,调度器109选择将被调度到集群114、116以用于处理的工作负载。类似于上文所描述的相关联分区参数,工作负载可具有相关参数,诸如工作负载ID、分区ID、优先级、预算、集群要求、预占、内核的数量和其他参数。一旦调度器109选择队列111和与选定队列111相关的工作负载,调度器109然后基于处理器104与协处理器106之间的耦接布置来生成与选定任务105相关的一个或多个发动请求。取决于示例,协处理器106可具有与处理器104的不同程度的同步。在一个示例中,协处理器106与处理器104解耦并且与处理器104异步操作。因此,当集群114、116根据相关联工作负载请求的优先级变得在协处理器106上可用时,调度由调度器109生成的工作负载发动请求。在此耦接布置中,几乎没有在已经在协处理器106上执行的工作负载上发生的任何预占。

[0055] 在另一个示例中,协处理器106与处理器104共享松散耦接布置。在此示例中,协处理器106以与处理器104的某种程度的同步操作。例如,在松散耦接布置中,协处理器106在数据帧边界处与处理器104同步,并且在数据帧边界结束时执行的任何未服务的工作负载在后续数据帧的开始时被清除。因此,处理器104和协处理器106都在松散耦接布置中具有相同的输入和输出数据速率。然而,协处理器106通常将在定时窗口期间与处理器104异步地操作,这意味着在处理器104上执行的分区和/或任务可与在协处理器106上执行的不相关的分区和/或工作负载并行执行。松散耦接布置可支持处理器104与协处理器106之间的预占和非预占调度。

[0056] 在又一个示例中,协处理器106与处理器104共享紧密耦接布置。在此示例中,协处理器106以与处理器104的高度同步操作;也就是说,协处理器106基于处理器104的定时窗口使与由处理器104同时执行的对应任务相关联的队列和/或工作负载执行同步。紧密耦接布置可以各种方式体现。在一个具体实施中,协处理器106在相同定时窗口期间与处理器104高度同步,或者换句话说,协处理器106在该定时窗口中执行与当前在处理器104上执行的一个或多个任务相关联的工作负载。当处理器104在后续定时窗口中执行另一个任务时,协处理器106然后执行与由处理器104执行的下一个任务相关联的工作负载。在另一个具体实施中,协处理器106与处理器104同步以用于后续定时间隔,但在协处理器106上执行与和其他优先级规则或处理可用性一致的不同任务相关联的工作负载的方面,协处理器106保持自由度。

[0057] 也可组合耦接布置。例如,协处理器106可相对于一个定时窗口与处理器104松散耦接,但相对于另一个定时窗口与处理器104紧密耦接。因此,调度策略可基于处理器104与协处理器106之间的耦接布置的组合来调度发动请求,并且可随着系统调度参数改变而动态地更新。

[0058] 图3描绘了在松散耦接的协处理器方案中将工作负载调度到协处理器(教学地描述为GPU)的多个集群的图300。在第一定时窗口(TW1)处,CPU 104接收新数据帧(帧1)并且在时间302开始第一CPU任务的执行。然后,CPU 104在时间303向GPU 106的集群1请求“初始化”工作负载以确认是否发起与第一CPU任务相关联的处理。在一些示例中,GPU驱动器108确定对于相关联任务不需要处理。例如,如果数据帧对应于相机图像的不同帧,则GPU驱动器108可在如果先前图像帧足够类似于当前图像帧时确定不需要处理。GPU 106在时间303在集群1上执行“初始化”工作负载,并且在定时窗口1内确认在时间302接收的工作负载需要处理。因此,GPU 106的集群1在时间304开始执行与第一CPU任务相关联的工作负载。

[0059] 虽然GPU 106的集群1继续处理来自在时间302执行的任务的工作负载,但定时窗口2(TW2)在CPU 104处在时间306开始,并且CPU 104在时间306开始处理第二任务。虽然集群1执行与CPU 104相关联的工作负载,但集群2开始执行与下一个CPU任务相关联的工作负载。在时间308,集群1完成与第一CPU任务相关联的工作负载的处理,并且开始处理与第二CPU任务相关联的工作负载。因此,在时间308,集群1和2都调用处理资源以执行与第二CPU任务相关联的工作负载。在此示例中,先前仅在集群2上执行的工作已被扩展并且现在在集群1和2两者上执行。然后在时间310,集群1和2完成处理与定时窗口2内的第二CPU任务相关联的工作负载。因为CPU 104没有需要在定时窗口2内进行调度的附加任务,所以可分派集群1和2以用于在时间310处理低优先级工作负载(如果此类工作负载为可用的)。对于最高

优先级工作负载,驱动器108被配置为对调度进行优先化,使得这些工作负载可在最早可用定时窗口内开始执行。相比之下,驱动器108被配置为每当处理资源变得可用时调度最低优先级工作负载。也就是说,驱动器108做出“最佳努力”以在最早可用定时窗口内调度低优先级工作负载,但一旦由于被较高优先级工作负载预占和/或用于执行较低优先级工作负载的不足处理资源而进行调度,此类低优先级工作负载可能无法开始或完成执行。在航空电子应用中,高优先级工作负载与高设计保证水平(DAL)(例如,A-C)相关联,而低优先级工作负载与低DAL(例如,D-E)相关联。

[0060] 在时间312,定时窗口改变为定时窗口1,并且CPU 104开始执行第三任务。在一些示例中,定时窗口经由时分复用按顺序进行调度。在时间314,GPU驱动器108从CPU 104接收指令以开始执行与第三CPU任务相关联的工作负载。由于第三CPU任务具有比在时间310之后由集群1和2执行的低优先级工作负载更高的优先级,因此GPU驱动器108在时间314停止(或预占)低优先级工作负载的执行并且将与第三CPU任务相关联的工作负载调度到集群1以用于执行。当集群1执行工作负载时,集群2任选地在时间314保持空闲。在时间316,集群1完成执行与第三CPU任务相关联的工作负载,并且集群1和2两者都恢复处理低优先级工作负载。定时窗口1和2可根据需要交替并且可以与或可不与新数据帧的接收同步。当处理帧1时,GPU 106可继续处理低优先级工作负载,直到CPU104执行需要GPU 106上的工作负载的另一个任务。数据帧中的定时窗口的数量可变化,并且在一些示例中独立于协处理器调度策略指定。

[0061] 在时间318,CPU 104接收新数据帧(帧2)。定时窗口1在接收到数据帧2之后不久的时间320开始,并且CPU 104开始执行第四CPU任务。在时间322,GPU 106然后执行工作负载以确定第四CPU任务是否需要处理;GPU驱动器108将此工作负载分配到集群1,如图3所示。当集群1执行工作负载时,其确定对于第四CPU任务不需要附加处理,并且在时间324,集群1和2都开始针对定时窗口1的剩余时间执行低优先级工作负载。

[0062] 在时间325,CPU 104执行第五CPU任务并且随后在时间326向GPU驱动器108发送针对与第五CPU任务相关联的工作负载的工作负载请求。在此情况下,GPU 106和CPU 104并行执行对应的相应工作负载和任务,其中CPU 104等待GPU 106“赶上”。因此,CPU 104和GPU 106并行执行。与第五CPU任务相关联的工作负载预占先前在GPU 106上执行的低优先级工作负载。在时间328,集群1和2完成与第五CPU任务相关联的工作负载并且恢复低优先级工作负载的处理。最后,在时间330,CPU 104在定时窗口1执行第六CPU任务并且确定不需要附加处理来执行第六CPU任务。因此,GPU 106继续针对定时窗口1的剩余时间处理低优先级工作负载,直到接收到新数据帧(帧3)。

[0063] 图4A至图4B描绘了在紧密耦接协处理器方案中将工作负载调度到GPU的多个集群的图。图400A示出了其中GPU执行与相同定时窗口内的CPU任务相关联的工作负载的示例。图400B示出了其中GPU执行与后续定时窗口中的CPU任务相关联的工作负载的示例。尽管图4A教学地示出了CPU的定时窗口与GPU的定时窗口之间的1:1相关性,但在其他示例中,CPU的定时窗口可具有与GPU的定时窗口不同的持续时间。而且,CPU可具有与GPU的定时窗口不同数量的定时窗口,并且与CPU任务相关联的GPU工作负载可具有不同的执行时间要求。

[0064] 首先参考图400A,CPU 104在时间401执行定时窗口1内的第一CPU任务。GPU驱动器108随后确定对于第一CPU任务需要处理,并且在时间402,GPU驱动器108将与第一CPU任务

相关联的工作负载调度到集群1以用于执行。集群1针对定时窗口1的其余部分继续执行工作负载,但在定时窗口2在时间403开始之前无法完成工作负载的执行。当定时窗口2开始时,CPU 104开始执行需要来自GPU 106的处理的第二CPU任务。在定时窗口1期间由集群1处理的工作负载在定时窗口2边界变得被预占。由于在此示例中GPU 106在定时窗口边界与CPU 104同步,因此一旦定时窗口2在时间403开始时,集群1停止与第一CPU任务相关联的工作负载的处理。同时,在定时窗口2的时间期间,集群2开始与第二CPU任务相关联的工作负载的处理。

[0065] 在时间404,定时窗口返回到定时窗口1。此时,在定时窗口2期间由集群2处理的工作负载变得被预占,并且在时间404,集群1恢复与先前已经在定时窗口2的开始时预占的第一CPU任务相关联的工作负载的处理。随着集群1恢复第一CPU任务工作负载的处理,CPU 104还执行第三CPU任务以用于处理。在时间405,集群1完成与第一CPU任务相关联的工作负载的处理,并且在时间406开始与第三CPU任务相关联的工作负载的处理。在时间407,当定时窗口返回到定时窗口2时,第三CPU任务工作负载的处理变得被预占。此时,集群2恢复与第二CPU任务相关联的工作负载的处理。

[0066] 在时间408,接收新数据帧(帧2)。在时间409,CPU 104执行第四CPU任务。在时间410,GPU驱动器108调度与第四CPU任务相关联的轻工作负载,并且确定对于第四CPU任务不需要附加处理。因此,GPU驱动器108在定时窗口1的其余部分期间调度低优先级工作负载。一旦定时窗口2开始,低优先级工作负载就被预占,并且CPU 104执行第五CPU任务。然后,在时间411,GPU驱动器108将与第五CPU任务相关联的工作负载调度到集群1和2两者。在时间412,GPU 106完成与第五CPU任务相关联的工作负载的执行并且针对定时窗口2的其余部分恢复低优先级工作负载的处理。

[0067] 图4B描绘了单独或与图4A组合使用的另选紧密耦接协处理器方案。接下来参考图400B,在时间413,CPU 104执行第一CPU任务。在时间414,CPU 104注册与第一CPU任务相关联的对应工作负载以确定附加处理是否为必须的。在一个示例中,CPU 104可在后续定时窗口期间调度的列表中注册待分配到GPU 106的工作负载。如图400B所示,CPU 104在定时窗口1的其余部分期间不执行附加任务。在时间415,定时窗口改变为定时窗口2。在此时间点,GPU驱动器108将来自先前定时窗口的注册工作负载进行排队以执行。然后,GPU驱动器108将与第一CPU任务相关联的工作负载调度到集群1以用于处理。GPU驱动器108确定第一CPU任务需要处理并且在时间416开始在集群1上处理与第一CPU任务相关联的工作负载。同时,CPU 104在时间415执行第二CPU任务并且记录针对下一个定时窗口的与第二CPU任务相关联的工作负载。在时间418,集群1完成与第一CPU任务相关联的工作负载的执行。由于在定时窗口1期间没有附加工作负载在排队,因此集群1和2开始针对定时窗口2的其余部分处理低优先级工作负载。

[0068] 在时间419,定时窗口改变为定时窗口1,并且从先前定时窗口排队的工作负载现在可由GPU 106的集群1和/或2执行。然而,在一些示例中,排队的工作负载在当前定时窗口内延迟指定时间。例如,排队的工作负载任选地包括完成工作负载所需的估计时间。如果估计时间小于当前定时窗口的持续时间,则排队的工作负载可被延迟,直到当前定时窗口中剩余的时间等于所估计的排队的工作负载的完成时间。这在图400B中示出,因为与第二CPU任务相关联的工作负载不由GPU 106在定时窗口1的开始时行,而是在已经过去一些时间

之后在时间421开始。相反,集群1和2两者在时间419期间处理低优先级工作负载,直到定时窗口1中的剩余时间等于有关与第二CPU任务相关联的工作负载的估计完成时间。

[0069] 同样,在时间419,CPU 104开始处理第三CPU任务。在时间420,CPU 104注册与第三CPU任务相关联的工作负载,GPU 106在时间422开始通过集群1执行该工作负载持续后续定时窗口的持续时间。

[0070] 随后接收新数据帧(帧2)。从时间423开始,CPU 104开始在定时窗口1处理第四CPU任务。由于没有工作负载由CPU 104在先前定时窗口注册,因此GPU 106在时间423开始的定时窗口1的持续时间内开始处理低优先级工作负载。定时窗口2在时间424开始。此时,GPU 106的集群1执行对于集群1的与第四CPU任务相关联的轻工作负载,同时CPU 104开始处理第五CPU任务。在时间425,集群1完成处理与第四CPU任务相关联的轻工作负载并且在定时窗口2的持续时间内恢复低优先级工作负载(以及集群2)的处理。在时间426,CPU 104执行第六CPU任务,同时集群1和2开始处理来自先前定时窗口的与第五CPU任务相关联的工作负载。一旦完成,集群1和2在定时窗口1的持续时间内恢复在时间427开始的低优先级工作负载的处理。

[0071] 在一些示例中,给定任务的优先级顺序基于其中最初调度该给定任务的定时窗口。例如,对于要在GPU上执行的给定三个工作负载(W1,W2,W3)的集合,W1可在定时窗口1中具有最高优先级并且因此在定时窗口1期间将不会由W2或W3抢占。一旦定时窗口2开始,优先级就可改变以使得W2具有最高优先级,从而使得W2能够立即调度以用于执行并且如果W1在定时窗口1期间未完成执行则抢占W1。类似地,一旦定时窗口切换到定时窗口3,W3则具有最高优先级并且可立即被调度以用于执行并可抢占W2。因此,当定时窗口在定时窗口1、2和3之间改变时,分配到GPU的工作负载之间的优先级顺序也可改变。

[0072] 图5A至图5B描绘了在CPU与GPU之间同步工作的图。具体地,图5A描绘了阻塞CPU-GPU同步,其中CPU基于由GPU同时执行的处理来处理任务,并且图5B描绘了将与单个CPU任务相关联的多个工作负载发动到GPU的多个集群。在图5A和图5B两者中,水平轴由时间表示。

[0073] 参考图5A,系统500A包括任何数量的CPU核心(CPU 1、CPU 2和多达N个量的CPU-N,其中N是整数)。另外,系统500A包括任何数量的GPU集群(集群1、集群2和多达N个量的集群N,其中N是整数)。每个CPU核心被配置为执行如先前所描述的一个或多个应用程序的任务,并且每个GPU集群被配置为执行与由CPU核心执行的任务相关联的工作负载,如先前所描述。在图5A所示的给定时间(单个定时窗口内或跨越数据帧的多个定时间隔的时间段)开始时,CPU核心CPU 1和CPU 2两者分别执行不同任务502和504。任务502和任务504两者需要由GPU集群进行的处理。因此,与在CPU 2上执行的任务504相关联的一个或多个工作负载508被调度到集群2,并且与在CPU 1上执行的任务502相关联的一个或多个工作负载505被调度到集群1。

[0074] 在各种示例中,一些CPU核心可基于对应GPU集群当前是否正在执行工作负载来执行任务。例如,考虑CPU 1。如图5A所示,CPU 1最初开始执行任务502持续一定时间段。当CPU 1执行任务502时,集群1开始执行与任务502相关联的一个或多个工作负载505。虽然集群1执行与第一CPU任务相关联的工作负载,但CPU 1被配置为延迟另一个任务(任务510)的处理,该任务可以是与任务502相同的任务,直到集群1完成执行与先前任务相关联的工作负

载。在此示例中,CPU 1和集群1以“阻塞”同步配置进行配置,因为CPU 1阻塞新任务的执行持续一定时间段,直到足够处理资源在GPU上可用,即GPU上的对应集群。

[0075] 在附加或另选的示例中,一些CPU-GPU同步配置不被“阻塞”。在这些示例中,CPU被配置为独立于对应GPU集群当前是否在执行与另一个任务相关联的工作负载而执行任务。如图5A所示,CPU 2最初开始执行任务504。一旦CPU 2完成执行504,与任务504相关联的一个或多个工作负载被调度到集群2以用于执行。当集群2执行工作负载508时,CPU 2然后执行另一个任务506。与CPU 1不同,CPU 2被配置为立即开始执行任务506,同时集群2执行与先前任务504相关联的工作负载508。非阻塞执行可通过以下来实现:(1)帧缓冲,其中CPU使多个帧排队以供GPU处理;(2)并行执行,其中CPU按顺序发动GPU上的多个工作负载(通过将工作负载请求切片到多个缓冲区的相同序列的部分,或在多个不相关序列中);和(3)批处理,其中CPU合并跨一个请求并行执行的多个类似请求。

[0076] 在一些示例中,CPU-GPU同步是部分“阻塞”的,使得CPU自由地与对应GPU同时执行任务,直到GPU变得过度积压了来自CPU的工作负载。在该情况下,CPU可等待直到GPU完成特定量的工作负载以“赶上”CPU。例如,CPU 2可等待特定时间段直到集群2在执行任务512之前完成工作负载508的执行。

[0077] 现在参考图5B,在一些示例中,与一个或多个CPU任务相关联的工作负载被调度到多个GPU集群。如图500B所示,与由CPU 1执行的任务相关联的工作负载被分配到集群1。具体地,当CPU 1按顺序执行任务514、任务518、任务526和任务534时,与这些任务相关联的一个或多个工作负载520、528被调度到集群1以用于执行。另选地说明了,CPU 1和集群1共享1:1同步配置,其中与由CPU 1执行的任务相关联的工作负载被调度到仅集群1以用于执行。相反,当CPU 2执行任务516、任务530和任务532中的任一者时,与这些任务中的任一者相关联的一个或多个工作负载可被调度到集群2和/或集群N。因此,CPU 1共享1:N同步配置,其中与由CPU 1执行的任务相关联的工作负载被调度到多个GPU集群以用于执行。如图500B所示,当CPU 2执行任务516时,与任务516相关联的工作负载522被调度到集群N,并且与任务516相关联的工作负载524另外被调度到集群2。如图500B所示的GPU集群还可实现如系统500A中所述的阻塞同步技术。

[0078] 协处理器调度策略(以及本文进一步描述的协处理器分配策略)的示例任选地包括预占策略,该预占策略管理被配置用于由协处理器GPU执行的工作负载的预占。预占调度的示例在图6A至图6B中示出(尽管也在其他附图中描绘),其中图6A描绘了非预占策略并且图6B描绘了CPU与GPU之间的预占策略的示例。参考图6A,集群1被配置为执行与由CPU 1和CPU 2两者执行的任务相关联的工作负载。CPU 1在CPU 2执行高优先级任务604的同时最初执行低优先级任务602。由于两个任务都需要通过GPU的处理,因此集群1被配置为执行与低优先级任务602(例如,最佳努力任务)相关联的工作负载606,并且还执行与高优先级任务604(例如,安全关键任务)相关联的工作负载608。如图6A所示,CPU 1在CPU2完成高优先级任务604之前完成低优先级任务602。因此,一旦CPU 1完成执行低优先级任务602,集群1开始执行工作负载606,而CPU 2继续执行高优先级任务604。

[0079] 在CPU 2完成高优先级任务604之后,集群1可执行与高优先级任务604相关联的工作负载608。然而,在CPU 2完成高优先级任务604时,集群1已经执行与CPU 1上执行的低优先级任务602相关联的工作负载606。在如图6A所示的非预占示例中,与低优先级任务602相

关联的工作负载606不由包括较高优先级工作负载的工作负载发动请求预占。因此,一旦集群1完成执行工作负载606,集群1在后续定时周期期间执行与高优先级任务604相关联的工作负载608。另选地说明了,由GPU集群执行的低优先级工作负载不能由具有被配置用于由相同GPU集群执行的较高优先级工作负载的后续工作负载发动请求预占。在这些示例中,后续工作负载发动请求替代地在先到先服务的基础上执行。

[0080] 相反,图6B示出了预占调度的示例,其中较低优先级的工作负载由较高优先级工作负载预占。如先前所描述的,任务604在优先级方面高于任务602;然而,集群1最初执行与低优先级任务602相关联的工作负载606,因为在CPU 2完成执行高优先级任务604之前,CPU 1完成执行低优先级任务602。在CPU 2完成执行高优先级任务604之后,集群1可执行与高优先级任务604相关联的工作负载608。工作负载608在优先级方面高于当前由集群1执行的工作负载606,因此集群1被配置为停止较低优先级工作负载606的执行并且开始高优先级工作负载608的执行。也就是说,高优先级工作负载608预占低优先级工作负载606。在一些示例中,集群1被配置为在完成高优先级工作负载608时,并且任选地当不存在具有比低优先级工作负载606更高的优先级的未决工作负载发动请求时,恢复低优先级工作负载606的执行。当执行较低优先级工作负载的进展可被存储和访问以用于在后续时间段进行处理时,这些示例是有用的。在其他示例中,预占较低优先级工作负载重置较低优先级工作负载上的进展并且由此要求GPU从开始重启较低优先级工作负载。与存储作业进展的示例相比,这些示例需要更少的处理带宽。

[0081] 对于实现协处理器(诸如GPU)上的工作负载预占的示例,协处理器例如从驱动器接收指定何时将发生较低优先级工作负载的预占的请求。可通过硬件和/或软件来实现预占策略。在一个硬件示例中,预占在命令边界处发生,使得一旦完成命令,或在最早可预占命令(即,当GPU可实现下一个命令时),较低优先级工作负载(或包括较低优先级工作负载的集合的背景)被预占。在另一个硬件示例中,预占在线程边界处发生,其中较低优先级背景停止发出附加的较低优先级工作负载并且在当前执行的所有工作负载完成时变得被预占。在又一个硬件示例中,工作负载执行通过将工作负载状态保存到存储器中来预占,一旦恢复执行就可复原该工作负载状态。在另一个硬件示例中,在执行线程期间发生预占,其中GPU可立即停止较低优先级线程的执行并且将线程的先前执行的部分存储到存储器中以用于稍后执行。

[0082] 协处理器还可通过软件实现预占。在一个软件示例中,如先前在硬件具体实施中描述的,预占在线程边界处发生。在另一个软件示例中,预占在接收到请求时立即发生,并且相同背景内的任何当前或先前执行的工作负载必须在稍后时间段重启,类似于如图6B中所提及的重置较低优先级工作负载。在另一个软件示例中,在执行工作负载期间,预占在限定的检查点处发生。检查点可在工作负载边界处或在执行工作负载期间的任何点处设置并且被保存以用于未来执行。一旦协处理器可继续执行工作负载,它就恢复执行保存检查点处的工作负载。另外地或另选地,在协处理器执行工作负载发动请求之前,工作负载被切片成多个子部分(例如,子内核),并且预占可在任何切片子内核边界处发生。

[0083] 图7描绘了示出用于将用于执行的工作负载调度到协处理器的示例性方法的流程图。方法700可以经由相对于图1至图6描述的技术来实施,但也可以经由其它技术来实施。为了便于解释,流程图的框以大致顺序的方式布置;然而,应当理解,这种布置结构仅仅是

示例性的,并且应当认识到,与本文所述的这些方法(和该图所示的框)相关联的处理可按不同次序发生(例如,其中与这些框相关联的处理中的至少一些处理以并行方式和/或以事件驱动方式执行)。

[0084] 方法700包括框702,即接收来自在处理器上执行的一个或多个任务的工作负载发动请求,诸如由在协处理器或其他处理单元上实现的驱动器。工作负载发动请求包括与由处理器执行的任务相关联的工作负载的列表,并且可包括其他参数,诸如列表中的工作负载的优先级和执行协处理器上的相应工作负载所需的处理资源。在框704处,方法700根据工作负载发动请求并且基于协处理器调度策略生成至少一个发动请求来行进。然后,驱动器或其他处理单元可基于发动请求和协处理器调度策略来调度工作负载以用于在协处理器上执行(框705)。

[0085] 取决于示例,方法700基于协处理器调度策略的条款来行进。任选地,方法700行进到框706并且调度工作负载以用于独立于处理器或其他外部事件的时间段(例如,定时窗口和/或数据帧边界)执行。在此松散耦接配置中,协处理器可与处理器的定时异步地调度工作负载。此类松散耦接配置任选地基于由协处理器接收的工作负载之间的优先级顺序来实现工作负载调度。例如,即使协处理器可与处理器定时窗口异步地调度工作负载,协处理器调度策略可包括抢占策略,该抢占策略通过较高优先级工作负载抢占当前在协处理器上执行或排队的较低优先级工作负载。

[0086] 另外地或另选地,方法700任选地行进到框708并且基于处理器的定时窗口调度工作负载以用于执行。在一个具体实施中,方法700在处理器的相同定时窗口期间调度工作负载以用于在协处理器上执行。在另一个具体实施中,方法700在相同定时窗口中调度工作负载以用于在协处理器上执行,其中协处理器106与处理器104在相同定时窗口中同步,但在协处理器上执行与和其他优先级规则或处理可用性一致的不同队列和/或任务相关联的工作负载的方面保持自由度。也就是说,协处理器调度策略任选地包括抢占策略,该抢占策略应用于紧密耦接配置并且基于工作负载的优先级顺序调度工作负载以用于执行。当工作负载发动请求包括与当前在协处理器上执行的工作负载相比具有更高优先级的工作负载时,协处理器调度策略配置协处理器以抢占较低优先级工作负载并且将较高优先级工作负载同步到处理器的后续定时窗口或甚至在协处理器与处理器之间的另一个常见定时窗口。

[0087] 协处理器分配策略

[0088] 如先前相对于图1A所述,在一些示例中,协处理器106(特别是驱动器108)被配置为实现协处理器分配策略,该协处理器分配策略根据至少一个工作负载发动请求将与任务105相关联的工作负载分配到集群114、116以用于执行。协处理器分配策略限定工作负载在协处理器106的一个或多个指定集群上执行的位置。图8至图11示出了在协处理器(诸如协处理器106)上分配工作负载的各种示例。本文进一步描述的分配技术可结合先前描述的调度策略或作为独立示例实现。例如,分配策略可实现在调度工作负载以用于执行的背景中描述的抢占技术。

[0089] 图8A至图8B描绘了协处理器的多个集群之间的数据耦接的图。在图8A所示的示例中,每个集群(集群1、集群2、集群3)被配置为执行与由处理器执行的任务相关联的工作负载。在一些示例中,诸如图8A所示,数据在帧边界处耦接或者另选地说明了,数据以提供给处理器的输入数据的帧速率耦接在协处理器与处理器之间。此数据耦接在图8A中通过从以

数字802指示的框架1边界延伸的虚线示出。在框架1边界802处,每个集群开始处理与相同处理器任务相关联的工作负载,并且即使通过以数字804指示的定时窗口边界也继续这样做。数据未耦接在定时窗口边界处,这使得集群能够独立于定时窗口1边界804并且在定时窗口2边界806处继续处理工作负载。一旦新帧2边界808到达,当数据由处理器(图8A中未示出)接收时,来自新数据帧的数据再次耦接在集群之间。

[0090] 图8B描绘了与数据帧边界相反的定时窗口边界处的数据耦接的图。在帧1边界802处,没有数据被分配到集群。相反,与处理器任务相关联的工作负载在定时窗口2边界804处被分配到所有三个集群。在定时窗口2边界804和定时窗口1边界806之间(在定时窗口2期间),集群可完成处理与一个处理器任务相关联的工作负载,并且可开始处理与另一个处理器任务相关联的工作负载或者可开始处理低优先级工作负载,直到集群之间的数据再次耦接在定时窗口边界806。如图8A至图8B中描绘的数据耦接不一定是排他性的并且可进行组合以具有耦接在定时窗口边界和帧边界两者的数据。

[0091] 图9描绘了应用于协处理器(诸如GPU)的多个集群的协处理器分配策略的图。可基于图8A至图8B的数据耦接技术来修改图9的背景中描述的分配策略。尽管在图9中示出了四个分配策略,但其他分配策略和它们的组合是可能的。例如,在第一时间间隔(例如,在一个数据帧内),根据交错策略将工作负载分配到集群,然而在不同时间间隔(例如,后续数据帧),根据策略-分布式策略来分配工作负载。另外,分配策略可应用于GPU的每个集群或可应用于集群的子集,其中GPU的集群的一个子集可遵循与集群的另一个子集不同的分配策略。

[0092] 在一些示例中,使用排他性策略将GPU作业分配到集群,其中与不同CPU任务相关联的工作负载在一个或多个定时间隔内被专门分配到不同集群。参考图9,与第一CPU任务相关联的作业(工作A)被分配到集群1,与第二CPU任务相关联的作业(工作B)被分配到集群2,并且与第三CPU任务相关联的作业(工作C)被分配到集群3,其中第一、第二和第三CPU任务彼此不同。排他性策略可以不同方式实现。例如,在排他性访问策略中,GPU的所有集群(以及因此所有计算单元)专用于处理相同CPU任务的作业。在排他性切片策略中,GPU被切片成多个GPU分区,每个分区包括一个或多个集群或集群的部分。在此示例中,来自相同CPU任务的工作负载仅被分配到单个GPU分区或集群组。在多个CPU任务的情况下,来自每个CPU任务的工作负载被分别分配到GPU的不同分区。对于具有足够隔离以包括多个隔离集群或分区的集群,排他性策略(或本文进一步描述的分配策略中的任一者)使得能够以分区级别或集群级别分配工作负载。此分配策略可用于实现如上所述的空间隔离。

[0093] 与排他性分配策略相比,交错分配策略将与相同CPU任务相关联的工作负载同时分配到GPU的多个集群。如图9所示,与第一CPU任务相关联的作业(工作A)被分配到集群1-3以用于在给定时间段(例如,定时窗口、数据帧或它们的一部分)内处理,之后是与第二CPU任务相关联的作业(工作B)在下一个数据耦接边界被分配到那些相同集群。可针对在指定给定时间段内需要分配的任何附加CPU任务重复此过程。另外,可从较早数据耦接边界恢复未完成的工作负载,并且在下一个数据耦接边界将该工作负载同时分配到所有三个集群。如图9所示,首先将工作A分配到集群1-3,然后是工作B,然后以工作C结束,并且针对后续定时间隔进行重复。此分配策略可用于实现如上所述的时间隔离。

[0094] 排他性分配策略和交错分配策略两者对应于静态分配策略,该静态分配策略独立

于工作负载优先级或计算资源将工作负载分配到集群/分区。相反,策略-分布式分配策略例示了考虑工作负载优先级和集群/分区的计算资源的动态分配策略。与处理器任务相关联的工作负载(其在优先级方面高于与另一个处理器任务相关联的另一个工作负载)通常将在较低优先级工作负载之前被分配并且与低优先级工作负载相比通常将被分配到更多可用集群。工作负载被分配到的集群或分区的量取决于处理工作负载所必需的资源量和/或协处理器中当前可用的计算资源的量。

[0095] 在图9所描绘的示例中,工作A需要最大量的计算资源来处理,而工作B需要最小量的计算资源来处理。为了适应更大数量的必要资源,工作A在给定时间间隔内占用集群1的计算资源的总体,并且在给定时间间隔的一部分内占用集群2和3的计算资源。相比之下,工作B仅在给定时间间隔的一部分内占用集群2的计算资源。策略-分布式策略可用于评估后续定时窗口的GPU工作负载的新集合以动态地调整工作负载的分配,使得一旦计算资源可用,较高优先级或较大工作负载就可开始。在策略-分布式分配策略(诸如图9所示的分配策略)中,在给定定时窗口期间执行工作负载之后变得可用的任何处理资源然后被分派到定时窗口中的最高优先级工作负载(即使在较低优先级工作负载之后调度最高优先级工作负载)。如图9所示,最高优先级工作负载是工作A,并且一旦工作B和C完成执行,被分派集群2和3的处理资源。

[0096] 工作负载有时可能需要超出协处理器中的当前可用计算资源的处理。因此,在一些示例中,分配策略(包括先前描述的分配策略中的任一者)包括取决于协处理器的硬件和系统参数来管理超过当前可用计算资源的排队工作负载的分配的策略。在一个示例中,超过当前可用计算资源的工作负载简单地保持排队,直到满足工作负载的处理要求的更多计算资源变得可用,由此使有限数量的可用计算资源空闲直到后续时间段。在另一个示例中,可用计算资源被分配到当前在协处理器上执行的最高优先级工作负载;也就是说,当前执行的最高优先级工作负载接收比最初请求的处理资源更多的处理资源(例如,集群、分区或计算单元)。在另一个示例中,即使当前可用计算资源不足,重工作负载也开始执行。在又一个示例中,具有满足可用计算资源的足够处理需求的最高优先级工作负载被分配可用计算资源。

[0097] 图9还描绘了共享分配策略的示例。共享分配策略将与不同处理器任务相关联的工作负载分配到协处理器的相同集群以在时间间隔内同时处理工作负载。例如,集群1接收其在定时窗口内同时处理的工作A、工作B和工作C的部分。集群2和3也在相同定时窗口内接收工作A、工作B和工作C的部分。在集群之间共享的部分可以或可不进行等分,并且在一些示例中取决于集群的处理能力和工作负载的处理要求。

[0098] 协处理器分配策略可包括本文所述的策略的组合。例如,协处理器分配策略可包括混合排他性共享策略,其中一个或多个集群被专门分配工作负载(即,一个集群接收与一个队列相关联的工作负载并且另一个集群接收与另一个队列相关联的工作负载),而另一个集群实现包括与不同任务相关联的工作负载的共享策略。

[0099] 图10A至图10C描绘了示出被配置为将工作负载分配到GPU 1000的多个集群的示例性系统的框图。GPU 1000包括被配置用于执行与需要在GPU 1000上处理的CPU任务相关联的工作负载的执行的硬件1004,并且在一些示例中,包括不同集群之间的硬件或软件隔离以减小或消除处理故障,从而免于影响并行处理。参考图10A,共享背景1012包括队列

1014,其包含对应于第一CPU任务的一个或多个工作负载(工作)1015。基于协处理器调度策略和/或协处理器分配策略,将工作负载1015分配到集群1006或计算单元1007中的一者或多者。背景1012任选地包括与不同CPU任务相关联的一个或多个附加工作负载1016。

[0100] 工作负载1015和任选的工作负载1016被发送到命令流化器1010以用于分配到集群1006或集群1008。例如,如果队列1014仅包括工作负载1015,则工作负载1015被分配到包括多个计算单元1007的集群1006中的至少一者。然而,当队列1014包含工作负载1016时,命令流化器1010被配置为将工作负载1016分配到集群1008的至少一个计算单元1009。在其他示例中,工作负载的分配由软件方法管理。如图10A所示,GPU 1000包括对应于单个队列1014的单个命令流化器1010。

[0101] 在一些示例中,GPU 1000包括将工作负载分配到GPU上的不同计算资源的多个队列和命令流化器。例如,图10B描绘了包括第一队列1014和第二队列1013的背景1012。第一队列1014不同于第二队列1013以提供足够空间或时间隔离并且包括对应于第一CPU任务的一个或多个工作负载1015,而第二队列1013包括对应于第二CPU任务的一个或多个工作负载1016。GPU 1000还包括第一命令流化器1010和第二命令流化器1011。在此示例中,第一命令流化器1010被配置为从第一队列1014接收工作负载1015并且将请求分配到集群1006(或集群1006的计算单元1007)中的一者或多者以用于处理请求中的工作负载。同时,第二命令流化器1011被配置为从第二队列1013接收工作负载1016并且将工作负载分配到包括多个计算单元1009的集群1008。尽管图10B示出了耦接到两个集群1006的第一命令流化器1010和耦接到单个集群1008的第二命令流化器1011,但第一命令流化器1010和第二命令流化器1011可被配置为通过软件、硬件或它们的组合将工作负载分配到由GPU 1000支持的任何数量的集群。

[0102] 在另一个示例中,图10C描绘了各自包括相应工作负载1015和1016的多个私人背景1012A和1012B。每个背景是不同的并且通过硬件或软件约束与其他背景隔离以提供足够的空间或时间隔离。另外,每个背景可被配置为将工作负载提供给相应命令流化器以用于分配到GPU 1000的集群。如图10C所示,背景1012A将工作负载1015提供给命令流化器1010,并且背景1012B将工作负载1016提供给命令流化器1011。然后,命令流化器1010将工作负载分配到一个或多个集群1006,并且命令流化器1011将工作负载1016分配到集群1008。在给定背景内,工作负载可并行或不按顺序执行。

[0103] 图11描绘了示出用于将内核分配到协处理器的处理资源的示例性方法的流程图。方法1100可以经由相对于图1至图10描述的技术来实施,但也可以经由其他技术来实现。为了便于解释,流程图的框以大致顺序的方式布置;然而,应当理解,这种布置结构仅仅是示例性的,并且应当认识到,与本文所述的这些方法(和该图所示的框)相关联的处理可按不同次序发生(例如,其中与这些框相关联的处理中的至少一些处理以并行方式和/或以事件驱动方式执行)。

[0104] 方法1100包括从在处理器上执行的一个或多个任务接收一个或多个工作负载发动请求,如框1102所示。然后,方法1100通过基于协处理器分配策略生成包括一个或多个工作负载的至少一个发动请求来行进到框1104。然后,方法1100通过基于协处理器分配策略将发动请求中识别的工作负载分配到协处理器上的处理资源来行进到框1105。例如,方法1100任选地行进到框1106以根据排他性策略将发动请求的每个工作负载分配到计算单元

的专用集群。

[0105] 另外地或另选地,方法1100行进到框1108,并且根据交错策略跨多个不同集群分配发动请求的每个工作负载。在本策略的一个示例中,发动请求中的第一工作负载(例如,具有最高优先级的工作负载)在第一时间间隔期间首先被分配到所有集群,然后是第二工作负载在第二时间间隔期间被分配到所有集群,以此类推,使得每个工作负载顺序地分配到集群中的每一者。

[0106] 另外地或另选地,方法1100行进到框1110,并且根据策略-分布式策略基于计算参数和/或工作负载的优先级将发动请求的每个工作负载分配到至少一个集群。例如,在集群上的执行的持续时间内,每个工作负载被单独地分配到至少一个集群。与处理器任务相关联的工作负载(其在优先级方面高于另一个工作负载)通常将在较低优先级工作负载之前被分配并且与低优先级工作负载相比通常将被分配到更多可用集群。工作负载被分配到的集群或分区的量取决于处理工作负载所必需的资源量和/或协处理器中当前可用的计算资源的量。

[0107] 在一些示例中,协处理器分配策略包括取决于协处理器的硬件和系统参数来管理超过当前可用计算资源的排队工作负载的分配的策略。在一个示例中,超过当前可用计算资源的工作负载简单地保持排队,直到满足工作负载的处理要求的更多计算资源变得可用,由此使有限数量的可用计算资源空闲直到后续时间段。在另一个示例中,可用计算资源被分配到当前在协处理器上执行的最高优先级工作负载;也就是说,当前执行的最高优先级工作负载接收比最初请求的处理资源更多的处理资源(例如,集群、分区或计算单元)。在另一个示例中,即使当前可用计算资源不足,重工作负载也开始执行。并且在又一个示例中,具有满足可用计算资源的足够处理需求的最高优先级工作负载被分配可用计算资源。

[0108] 另外地或另选地,方法1100行进到框1112并且在相同定时间间隔期间在多个集群之间分配发动请求的多个工作负载,使得根据共享分配策略在多个集群之间共享工作负载的部分。在一个示例中,在相同定时间间隔期间跨所有集群共享发动请求中的每个工作负载,使得每个集群同时处理每个工作负载。其他协处理器分配策略是可能的。

[0109] 图12描绘了示出用于在执行协处理器上的工作负载时(诸如在当前执行的工作负载用完协处理器上的执行预算时)管理处理资源的示例性方法的流程图。方法1200可以经由相对于图1至图11描述的技术来实施,但也可以经由其他技术来实现。例如,方法1200可在工作负载被调度和/或分配到协处理器以用于执行时按顺序实现,但也可在调度事件期间实现以维持处理器与协处理器之间的适当同步。为了便于解释,流程图的框以大致顺序的方式布置;然而,应当理解,这种布置结构仅仅是示例性的,并且应当认识到,与本文所述的这些方法(和该图所示的框)相关联的处理可按不同次序发生(例如,其中与这些框相关联的处理中的至少一些处理以并行方式和/或以事件驱动方式执行)。

[0110] 方法1200包括框1202并且接收例如来自自由驱动器接收的工作负载发动请求的关于工作负载预算约束的信息。在当前执行的工作负载用完预算时,方法1200行进到框1203并且确定在协处理器上处理工作负载之后是否存在剩余的任何附加处理预算。如果存在剩余的附加处理预算,则方法1200行进到框1204并且从已完成工作负载获取对应任务预算并另外地接收已完成工作负载的优先级。从那里,可使用附加预算和优先级信息以在后续定时间间隔期间处理排队的工作负载。

[0111] 如果没有预算可用,则方法1200行进到框1206以预占和/或停止当前执行的工作负载。任选地,方法1200然后可行行进到框1208并且将工作负载重新调度到协处理器以用于执行。当根据如先前描述的协处理器调度策略调度工作负载时,可实现此示例。另外地或另选地,方法1200行进到框1210(直接从框1208或从框1206)以重新分配工作负载优先级并且任选地基于已更新工作负载优先级来重新调度工作负载以用于执行。

[0112] 图13描绘了示出用于在背景中优先化工作负载的示例性方法的流程图。方法1300可以经由相对于图1至图12描述的技术来实施,但也可以经由其他技术来实现。为了便于解释,流程图的框以大致顺序的方式布置;然而,应当理解,这种布置结构仅仅是示例性的,并且应当认识到,与本文所述的这些方法(和该图所示的框)相关联的处理可按不同次序发生(例如,其中与这些框相关联的处理中的至少一些处理以并行方式和/或以事件驱动方式执行)。

[0113] 方法1300包括框1302并且将来自一个或多个工作负载发动请求的工作负载分选到一个或多个背景中。在一些示例中,协处理器包括多个队列,每个队列独立地被配置有与和另一个队列相关联的工作负载隔离的不同工作负载。在这些示例中,工作负载被分选到多个背景中的每一者中。另选地,对于具有仅一个背景的协处理器,所有工作负载被分选到单个背景中。

[0114] 然后,方法1300通过基于背景中的每个工作负载的优先级来分选给定背景内的工作负载来行进到框1304。对于由协处理器支持的每个背景,并行地重复或进行此步骤。在一些示例中,背景的数量取决于协处理器上的队列的数量。例如,协处理器可具有分别对应于背景中的一者的两个队列。对于实现多个背景的协处理器,方法1300任选地行进到框1306以基于与背景相关联的队列的优先级来对背景进行分选。将在排队背景列表中首先调度和分配具有最高优先级的队列。对于单队列协处理器,不需要框1306,因为协处理器计算资源将接收包含被调度以用于执行的工作负载的列表的单个队列。一旦基于队列的优先级选择背景,计算资源就开始基于背景内的工作负载的优先级在选定背景中执行相应队列。每个队列和/或背景的优先级的分选是针对时间点确定的,并且可在附加工作负载请求变得可用时进行进一步更新或调整。

[0115] 图14A至图14C描绘了示出用于调度和分配工作负载的示例性方法的流程图。方法1400A-1400C可以经由相对于图1至图13描述的技术来实施,但也可以经由其他技术来实现。具体地,方法1400A-1400C可如下文进一步描述的那样按顺序执行。此外,方法1400A-1400C中的一些步骤可以是任选的,这取决于处理器与协处理器之间的架构和耦接。为了便于解释,流程图的框以大致顺序的方式布置;然而,应当理解,这种布置结构仅仅是示例性的,并且应当认识到,与本文所述的这些方法(和该图所示的框)相关联的处理可按不同次序发生(例如,其中与这些框相关联的处理中的至少一些处理以并行方式和/或以事件驱动方式执行)。

[0116] 从框1402开始,方法1400A选择多个背景中的最高优先级背景,该多个背景各自包括要调度和分配到协处理器的处理资源的多个工作负载。然后,方法1400A行进到框1403,并且针对给定背景中的工作负载确定是否存在保持在背景中的较高优先级工作负载。如果在背景中不存在较高优先级工作负载,则方法1400A任选地行进到框1404以分派一个或多个集群以便执行如由协处理器分配策略限定的工作,该协处理器分配策略的示例如先前上

文所述。另外地或另选地，方法1400A在框1408处终止。

[0117] 对于在背景中仍然存在的较高优先级工作负载，方法1400A替代地行进到框1406并且准备背景中的最高优先级工作负载以用于执行。然后，方法1400A可进一步行进到指示符框A(框1410)以进一步继续进入方法1400B。

[0118] 从指示符框A(框1410)，方法1400B行进到框1411并且确定在协处理器上是否存在足够空间以执行在框1406中准备的较高优先级工作负载。如果是，则方法1400B行进到框1418并且在协处理器上发动较高优先级工作负载。在背景支持多个队列的示例中，工作负载可在队列的相关联工作负载在协处理器上执行之前分布在队列之间。

[0119] 如果协处理器上可用的空间不足，则方法1400B替代地通过确定是否存在利用协处理器上的额外集群的当前执行或调度的任何工作负载来任选地行进到框1412，其可基于与任务、分区或定时窗口相关联的调度参数(包括预算、优先级规则、所请求的集群数量等参数)来确定。如果执行或调度的工作负载都未利用额外集群，则方法1400B行进到框1416并且基于预占策略来预占较低优先级工作负载，直到存在足够集群以供较高优先级工作负载执行。从那里，方法1400B可行进返回到框1411以确定GPU上是否存在足够空间。否则，如果存在利用协处理器上的额外集群的此类工作负载，则方法1400B替代地通过使用额外集群来预占工作负载而任选地行进到框1413。然后，方法1400B任选地在框1414处再次确定在预占额外集群之后在协处理器上是否存在足够空间来发动较高优先级工作负载。如果否，则方法1400B行进到框1416并且基于预占策略来预占较低优先级工作负载，直到存在足够集群以供较高优先级工作负载执行。然而，如果足够空间在框1414处可用，则方法1400B行进到框1418并且在协处理器上发动较高优先级工作负载。然后，方法1400B可行进到指示符框B(框1420)并且进一步继续进入方法1400C。

[0120] 从指示符框B开始(框1420)，方法1400C行进到框1421并且确定在协处理器上是否存在任何空闲或可用的集群。如果不存在空闲集群(所有集群当前正在处理工作负载)，则方法1400C在框1428处结束。如果在协处理器上存在空闲集群，则方法1400C然后任选地行进到框1422以确定是否存在可用于下一个最高优先级工作负载的足够空间。如果在空闲集群上存在可用的足够空间以处理下一个工作负载，则方法1400C行进到框1426并且准备最高优先级工作以用于在空闲集群中的至少一者上执行。然而，如果在框1422处存在空闲集群但不存在足够空间来执行下一个最高优先级工作负载，则方法1400C任选地行进到框1424以基于协处理器分配策略来分派空闲集群。例如，并非执行下一个最高优先级工作负载，可通过策略-分布式协处理器策略或本文所述的其他协处理器分配策略中的任一者来分派空闲集群以在其他集群上处理当前执行的工作负载。然后，方法1400C可在框1428处结束。

[0121] 本文所述的方法和技术可在数字电子电路中或用可编程处理器(例如，专用处理器或通用处理器诸如计算机)固件、软件或每一者的各种组合来实现。体现这些技术的装置可包括适当的输入和输出设备、可编程处理器以及有形地体现用于由可编程处理器执行的程序指令的存储介质。可通过以下方式执行体现这些技术的过程：可编程处理器执行指令程序以通过对输入数据进行操作以及生成输出来执行期望的功能。这些技术可有利地在可编程系统上可执行的一个或多个程序中实现，该可编程系统包括至少一个可编程处理器，该至少一个可编程处理器经耦合以从数据存储系统、至少一个输入设备和至少一个输

出设备接收数据和指令,以及将数据和指令传输到数据存储系统、至少一个输入设备和至少一个输出设备。一般来讲,处理器将从只读存储器和/或随机存取存储器接收指令和数据。适合于有形地体现计算机程序指令和数据的存储设备包括所有形式的非易失性存储器,包括(以举例的方式)半导体存储器设备,诸如可擦除可编程只读存储器(EPROM)、电可擦除可编程只读存储器(EEPROM)和闪存存储器设备;磁盘,诸如内部硬盘和可移动磁盘;磁光盘;和数字视频盘(DVD)。前述任一者可由专门设计的专用集成电路(ASIC)补充或并入其中。

[0122] 示例性实施方案

[0123] 实施例1包括一种处理系统,包括:处理器;协处理器,该协处理器被配置为实现处理引擎;处理引擎调度器,该处理引擎调度器被配置为调度工作负载以用于在该协处理器上执行;其中该处理引擎调度器被配置为从在该处理器上执行的一个或多个任务接收一个或多个工作负载发动请求,并且作为响应,基于协处理器调度策略来生成用于提交到该协处理器的至少一个发动请求;其中基于该协处理器调度策略,该处理引擎调度器选择激活哪些协处理器集群以基于该至少一个发动请求执行由队列识别的工作负载;并且其中该协处理器调度策略限定以下中的至少一者:紧密耦接协处理器调度,其中调度由该至少一个发动请求识别的工作负载以在其中该一个或多个任务在该处理器上执行的定时窗口的时间内立即在该协处理器上执行;或紧密耦接协处理器调度,其中调度由该至少一个发动请求识别的工作负载以基于优先级顺序并且通过以下中的任一者在该协处理器上执行:相对于该处理器和该协处理器两者共同的外部事件,或在其中该一个或多个任务在该处理器上执行的该定时窗口的该时间之后的后续定时窗口期间。

[0124] 实施例2包括根据实施例1所述的处理系统,其中该协处理器调度策略限定松散耦接协处理器调度,其中调度由该至少一个发动请求识别的工作负载以独立于在该处理器上执行的该一个或多个任务的定时窗口并且基于该工作负载的优先级顺序来执行。

[0125] 实施例3包括根据实施例1至2中任一项所述的处理系统,其中该处理器包括中央处理单元(CPU),该CPU包括至少一个处理核心,并且该协处理器包括图形处理单元(GPU)、处理加速器、现场可编程门阵列(FPGA)或专用集成电路(ASIC)。

[0126] 实施例4包括根据实施例1至3中任一项所述的处理系统,其中该处理器包括多个处理器核心,并且其中该处理引擎调度器被配置为生成至少一个发动请求,该至少一个发动请求将与该多个处理核心中的一个处理器核心相关联的工作负载调度到该协处理器的多个集群以用于执行。

[0127] 实施例5包括根据实施例1至4中任一项所述的处理系统,其中调度由该至少一个发动请求识别的该工作负载以在其中该一个或多个任务在该处理器上执行的该定时窗口的该时间之后的后续定时窗口期间和/或在该处理器的后续数据帧边界期间执行。

[0128] 实施例6包括根据实施例1至5中任一项所述的处理系统,其中该协处理器调度策略包括限定耦接协处理器调度的预占策略,其中被调度用于在该协处理器上执行的或当前在该协处理器上执行的一个或多个工作负载被配置为由排队以基于该优先级顺序进行执行的一个或多个工作负载预占。

[0129] 实施例7包括根据实施例6所述的处理系统,其中当前正在该协处理器上执行的该一个或多个工作负载被配置为由排队以进行执行的一个或多个较高优先级工作负载预占,

并且其中该协处理器被配置为：存储当前正在该协处理器上执行的该一个或多个工作负载；并且重新调度所存储的一个或多个工作负载以用于在已执行该较高优先级工作负载之后的后续定时窗口期间执行。

[0130] 实施例8包括根据实施例6至7中任一项所述的处理系统，其中该预占策略限定以下中的至少一者：耦接协处理器调度，其中当前在该协处理器上执行的一个或多个工作负载被配置为完成并且排队以用于执行的后续工作负载由较高优先级工作负载预占；耦接协处理器调度，其中当前在该协处理器上执行的一个或多个工作负载被配置为由较高优先级工作负载预占；耦接协处理器调度，其中当前在该协处理器上执行的一个或多个工作负载被配置为由较高优先级工作负载预占，其中该一个或多个工作负载包括识别已经执行的相应工作负载的一部分的指示符，并且其中该一个或多个工作负载被配置为从该指示符开始进行存储和重新执行；或耦接协处理器调度，其中被调度用于执行的该一个或多个工作负载被分区成多个子部分，并且该多个子部分中的每个子部分被配置为由较高优先级工作负载预占。

[0131] 实施例9包括根据实施例1至8中任一项所述的处理系统，其中该处理引擎包括计算引擎、渲染引擎或人工智能 (AI) 推断引擎，并且其中该处理引擎调度器包括计算引擎调度器、渲染引擎调度器或推断引擎调度器。

[0132] 实施例10包括协处理器，该协处理器被配置为耦接到处理器且被配置为实现处理引擎，该协处理器包括：至少一个集群，该至少一个集群被配置为执行工作负载；处理引擎调度器，该处理引擎调度器被配置为调度工作负载以用于在该协处理器上执行；其中该处理引擎调度器被配置为从在该处理器上执行的一个或多个任务接收一个或多个工作负载发动请求，并且作为响应，基于协处理器调度策略来生成用于提交的至少一个发动请求；其中基于该协处理器调度策略，该处理引擎调度器选择激活该至少一个集群中的哪个集群以执行由包括该至少一个发动请求的队列识别的工作负载；并且其中该协处理器调度策略限定以下中的至少一者：紧密耦接协处理器调度，其中调度由该至少一个发动请求识别的工作负载以在其中该一个或多个任务在该处理器上执行的定时窗口的时间内立即在该协处理器上执行；或紧密耦接协处理器调度，其中调度由该至少一个发动请求识别的工作负载以基于优先级顺序并且通过以下中的任一者在该协处理器上执行：相对于该处理器和该协处理器两者共同的外部事件，或在其中该一个或多个任务在该处理器上执行的该定时窗口的该时间之后的后续定时窗口期间。

[0133] 实施例11包括根据实施例10所述的协处理器，其中该处理引擎包括计算引擎、渲染引擎或人工智能 (AI) 推断引擎，并且其中该处理引擎调度器包括计算引擎调度器、渲染引擎调度器或推断引擎调度器。

[0134] 实施例12包括根据实施例10至11中任一项所述的协处理器，其中该协处理器调度策略限定松散耦接协处理器调度，其中调度由该至少一个发动请求识别的工作负载以独立于在该处理器上执行的该一个或多个任务的定时窗口并且基于该工作负载的优先级顺序来执行。

[0135] 实施例13包括根据实施例10至12中任一项所述的协处理器，其中该处理引擎调度器被配置为生成至少一个发动请求，该至少一个发动请求将与该处理器的多个处理核心中的一个处理核心相关联的工作负载调度到该协处理器的多个集群以用于执行。

[0136] 实施例14包括根据实施例10至13中任一项所述的协处理器,其中调度由该至少一个发动请求识别的该工作负载以在其中该一个或多个任务在该处理器上执行的该定时窗口的该时间之后的后续定时窗口期间和/或在该处理器的后续数据帧边界期间执行。

[0137] 实施例15包括根据实施例10至14中任一项所述的协处理器,其中该协处理器调度策略包括限定耦接协处理器调度的预占策略,其中被调度用于在该协处理器上执行的或当前在该协处理器上执行的一个或多个工作负载被配置为由排队以基于该优先级顺序进行执行的一个或多个工作负载预占。

[0138] 实施例16包括根据实施例15所述的协处理器,其中当前在该协处理器上执行的该一个或多个工作负载被配置为由排队以进行执行的一个或多个较高优先级工作负载预占,并且其中该协处理器被配置为:存储当前在该协处理器上执行的一个或多个工作负载;并且重新调度所存储的一个或多个工作负载以用于在已执行该较高优先级工作负载之后的后续定时窗口期间执行。

[0139] 实施例17包括根据实施例15至16中任一项所述的协处理器,其中该预占策略限定以下中的至少一者:耦接协处理器调度,其中当前在该协处理器上执行的一个或多个工作负载被配置为完成并且排队以用于执行的后续工作负载由较高优先级工作负载预占;耦接协处理器调度,其中当前在该协处理器上执行的一个或多个工作负载被配置为由较高优先级工作负载预占;耦接协处理器调度,其中当前在该协处理器上执行的一个或多个工作负载被配置为由较高优先级工作负载预占,其中该一个或多个工作负载包括识别已经执行的相应工作负载的一部分的指示符,并且其中该一个或多个工作负载被配置为从该指示符开始进行存储和重新执行;或耦接协处理器调度,其中被调度用于执行的该一个或多个工作负载被分区成多个子部分,并且该多个子部分中的每个子部分被配置为由较高优先级工作负载预占。

[0140] 实施例18包括一种方法,包括:从在处理器上执行的一个或多个任务接收一个或多个工作负载发动请求,其中该一个或多个工作负载发动请求包括被配置用于在协处理器上执行的一个或多个工作负载;基于协处理器调度策略,响应于该一个或多个工作负载发动请求而生成至少一个发动请求;基于该协处理器调度策略通过以下中的至少一者来调度该至少一个发动请求中识别的一个或多个工作负载以用于在该协处理器上执行:紧密耦接协处理器调度,其中调度由该至少一个发动请求识别的工作负载以在其中该一个或多个任务在该处理器上执行的定时窗口的时间内立即在该协处理器上执行;或紧密耦接协处理器调度,其中调度由该至少一个发动请求识别的工作负载以基于优先级顺序并且通过以下中的任一者在该协处理器上执行:相对于该处理器和该协处理器两者共同的外部事件,或在其中该一个或多个任务在该处理器上执行的该定时窗口的该时间之后的后续定时窗口期间。

[0141] 实施例19包括根据实施例18所述的方法,包括由排队以基于该优先级顺序进行执行的一个或多个工作负载预占被调度用于执行或当前在该协处理器上执行的至少一个工作负载。

[0142] 实施例20包括根据实施例19所述的方法,其中预占被调度以用于执行或当前在该协处理器上执行的至少一个工作负载包括:耦接协处理器调度,其中当前在该协处理器上执行的一个或多个工作负载被配置为完成并且排队以用于执行的后续工作负载由较高优

先级工作负载预占;耦接协处理器调度,其中当前在该协处理器上执行的一个或多个工作负载被配置为由较高优先级工作负载预占;耦接协处理器调度,其中当前在该协处理器上执行的一个或多个工作负载被配置为由较高优先级工作负载预占,其中该一个或多个工作负载包括识别已经执行的相应工作负载的一部分的指示符,并且其中该一个或多个工作负载被配置为从该指示符开始进行存储和重新执行;或耦接协处理器调度,其中被调度用于执行的该一个或多个工作负载被分区成多个子部分,并且该多个子部分中的每个子部分被配置为由较高优先级工作负载预占。

[0143] 实施例21包括一种处理系统,包括:处理器;协处理器,该协处理器被配置为实现处理引擎;处理引擎调度器,该处理引擎调度器被配置为调度工作负载以用于在该协处理器上执行;其中该处理引擎调度器被配置为从在该处理器上正执行或已执行的一个或多个任务接收一个或多个工作负载发动请求,并且作为响应,生成用于提交到该协处理器的至少一个发动请求;其中该协处理器包括多个计算单元和与该多个计算单元中的一个或多个计算单元相关联的至少一个命令流化器;其中基于协处理器分配策略,该处理引擎调度器被配置为经由该至少一个命令流化器针对给定执行分区分配该协处理器的计算单元的集群,以根据工作负载优先级执行由该一个或多个工作负载发动请求识别的一个或多个工作负载;其中该协处理器分配策略至少限定:排他性分配策略,其中每个工作负载由计算单元的专用集群执行;交错分配策略,其中每个工作负载跨该计算单元集群的所有计算单元专门执行;策略-分布式分配策略,其中每个工作负载在该给定执行分区期间被单独地分配到该计算单元的集群中的至少一个集群和执行持续时间;或共享分配策略,其中每个工作负载不专门由该计算单元的集群执行,该计算单元的集群各自同时执行多个工作负载。

[0144] 实施例22包括根据实施例21所述的处理系统,其中该处理器包括中央处理单元(CPU),该CPU包括至少一个处理核心,并且该协处理器包括图形处理单元(GPU)、处理加速器、现场可编程门阵列(FPGA)或专用集成电路(ASIC)。

[0145] 实施例23包括根据实施例21至22中任一项所述的处理系统,其中该至少一个命令流化器被配置为从包括与在该处理器上正执行或已执行的多个任务相关联的工作负载的共享背景接收工作负载,并且其中该至少一个命令流化器被配置为将与该多个任务中的第一任务相关联的工作负载分配到第一集群集合,并且将与该多个任务中的第二任务相关联的工作负载分配到不同于该第一集群集合的第二集群集合。

[0146] 实施例24包括根据实施例21至23中任一项所述的处理系统,其中该至少一个命令流化器包括被配置为从共享背景接收工作负载的多个命令流化器,其中该共享背景包括多个队列,其中该多个命令流化器中的第一命令流化器被配置为:从该多个队列中的第一队列接收与在该处理器上正执行或已执行的第一任务相关联的第一工作负载,并且将该第一工作负载分配到计算单元的集群的第一集合;其中该多个命令流化器中的第二命令流化器被配置为:从该多个队列中的不同于该第一队列的第二队列接收与在该处理器上正执行或已执行的不同于该第一任务的第二任务相关联的第二工作负载;并且将该第二工作负载分配到与计算单元的集群的该第一集合不同的计算单元的集群的第二集合。

[0147] 实施例25包括根据实施例21至24中任一项所述的处理系统,其中该至少一个命令流化器包括多个命令流化器,其中该多个命令流化器中的第一命令流化器被配置为:从第一背景的第一队列接收与在该处理器上正执行或已执行的第一任务相关联的第一工作负

载,并且将该第一工作负载分配到计算单元的集群的第一集合;其中该多个命令流化器中的第二命令流化器被配置为:从不同于该第一背景的第二背景的第二队列接收与在该处理器上正执行或已执行的不同于该第一任务的第二任务相关联的第二工作负载;并且将该第二工作负载分配到与计算单元的集群的该第一集合不同的计算单元的集群的第二集合。

[0148] 实施例26包括根据实施例21至25中任一项所述的处理系统,其中该协处理器分配策略包括预占策略,该预占策略限定已分配到该计算单元的集群中的一个或多个集群或正在分配到该计算单元的集群中的该一个或多个集群的一个或多个工作负载被配置为由基于该工作负载优先级排队以便被分配用于在该协处理器上执行的一个或多个工作负载预占。

[0149] 实施例27包括根据实施例21至26中任一项所述的处理系统,其中:该处理引擎包括计算引擎并且该处理引擎调度器包括计算引擎调度器;该处理引擎包括渲染引擎并且该处理引擎调度器包括渲染引擎调度器;或该处理包括人工智能(AI)推断引擎并且该处理引擎调度器包括推断引擎调度器。

[0150] 实施例28包括根据实施例21至27中任一项所述的处理系统,其中该处理引擎调度器被配置为基于完成该工作负载所需的处理量来分配该计算单元的集群中的一个或多个集群以执行该工作负载。

[0151] 实施例29包括根据实施例28所述的处理系统,其中该处理引擎调度器被配置为分配该计算单元的集群中的一个或多个附加集群以执行该工作负载以便补偿完成该工作负载所需的该处理量超过该协处理器上的当前可用处理资源的情况。

[0152] 实施例30包括协处理器,该协处理器被配置为耦接到处理器且被配置为实现处理引擎,该协处理器包括:多个计算单元,该多个计算单元各自被配置为执行工作负载;处理引擎调度器,该处理引擎调度器被配置为分配工作负载以用于在该协处理器上执行;其中该处理引擎调度器被配置为从在该处理器上正执行或已执行的一个或多个任务接收一个或多个工作负载发动请求,并且作为响应,生成用于提交到该协处理器的至少一个发动请求;其中该协处理器包括与该多个计算单元中的一个或多个计算单元相关联的至少一个命令流化器;其中基于协处理器分配策略,该处理引擎调度器被配置为经由该至少一个命令流化器针对给定执行分区分配该协处理器的计算单元的集群,以根据工作负载优先级执行由该一个或多个工作负载发动请求识别的一个或多个工作负载;其中该协处理器分配策略至少限定:排他性策略,其中每个工作负载由该计算单元的集群中的专用集群执行;交错策略,其中每个工作负载跨该计算单元的集群中的至少一个集群的所有计算单元专门执行;策略-分布式策略,其中每个工作负载在该给定执行分区期间被单独地分配到该计算单元的集群中的至少一个集群和执行持续时间;共享策略,其中每个工作负载不专门由该计算单元的集群执行,该计算单元的集群各自同时执行多个工作负载。

[0153] 实施例31包括根据实施例30所述的协处理器,其中该协处理器分配策略限定该排他性策略、交错策略、策略-分布式策略或共享策略中的两者或更多者,并且其中该处理引擎调度器被配置为在与该处理器相关联的后续定时边界处将该协处理器分配策略从一个策略调整到第二策略。

[0154] 实施例32包括根据实施例30至31中任一项所述的协处理器,其中该处理引擎调度器被配置为确定分派到已完成工作负载的未使用处理资源,并且基于该未使用处理资源和

分派到后续排队工作负载的处理资源来分配该后续排队工作负载以用于在该协处理器上执行。

[0155] 实施例33包括根据实施例30至32中任一项所述的协处理器,其中该排他性策略限定排他性访问策略和排他性切片策略中的至少一者,其中该排他性访问策略限定分配策略,其中每个工作负载被分配到该计算单元的集群的所有集群,其中该排他性切片策略限定分配策略:其中与在该处理器上正执行或已执行的第一任务相关联的工作负载被分配到第一个集群并且其中与在该处理器上正执行或已执行的第二任务相关联的工作负载被分配到第二多个集群;和/或其中与在该处理器上正执行或已执行的第一任务相关联的工作负载被分配到集群的第一部分,并且其中与在该处理器上正执行或已执行的第二任务相关联的工作负载被分配到该集群的第二部分。

[0156] 实施例34包括根据实施例30至33中任一项所述的协处理器,其中该协处理器分配策略包括预占策略,该预占策略限定已分配到该计算单元的集群中的一个或多个集群或正在分配到该计算单元的集群中的该一个或多个集群的一个或多个工作负载被配置为由基于该工作负载优先级排队,以便被分配用于执行的一个或多个工作负载预占。

[0157] 实施例35包括根据实施例30至34中任一项所述的协处理器,其中:该处理引擎包括计算引擎并且该处理引擎调度器包括计算引擎调度器;该处理引擎包括渲染引擎并且该处理引擎调度器包括渲染引擎调度器;或该处理包括人工智能(AI)推断引擎并且该处理引擎调度器包括推断引擎调度器。

[0158] 实施例36包括根据实施例30至35中任一项所述的协处理器,其中该处理引擎调度器被配置为基于完成该工作负载所需的处理量来分配该计算单元的集群中的一个或多个集群以执行该工作负载。

[0159] 实施例37包括根据实施例36所述的协处理器,其中该处理引擎调度器被配置为分配该计算单元的集群中的一个或多个附加集群以执行该工作负载以便补偿完成该工作负载所需的该处理量超过该协处理器上的当前可用处理资源的情况。

[0160] 实施例38包括一种方法,包括:从在处理器上正执行或已执行的一个或多个任务接收一个或多个工作负载发动请求,其中该一个或多个工作负载发动请求包括被配置用于在协处理器上执行的一个或多个工作负载;响应于该一个或多个工作负载发动请求而生成至少一个发动请求;分配该协处理器的计算单元的集群以基于协处理器分配策略根据工作负载优先级执行该一个或多个工作负载发动请求中识别的一个或多个工作负载,其中该协处理器分配策略至少限定以下:排他性策略,其中每个工作负载由该计算单元的集群中的专用集群执行;交错策略,其中每个工作负载跨该计算单元的集群中的至少一个集群的所有计算单元专门执行;策略-分布式策略,其中每个工作负载在给定执行分区期间被单独地分配到该计算单元的集群中的至少一个集群和执行持续时间;共享策略,其中每个工作负载不专门由该计算单元的集群执行,该计算单元的集群各自同时执行多个工作负载。

[0161] 实施例39包括根据实施例38所述的方法,包括由排队以基于该工作负载优先级进行执行的一个或多个工作负载预占被调度用于执行或当前在该协处理器上执行的至少一个工作负载。

[0162] 实施例40包括根据实施例38至39中任一项所述的方法,其中分配该协处理器的计算单元的集群以执行在该一个或多个工作负载发动请求中识别的一个或多个工作负载包

括分配该计算单元的集群中的一个或多个附加集群以执行该一个或多个工作负载以便补偿完成该一个或多个工作负载所需的处理量超过该协处理器上的当前可用处理资源的情况。

[0163] 尽管本文已说明和描述了特定实施方案,但本领域的普通技术人员将认识到,经计算以实现相同目的的任何布置可替代所展示的特定实施方案。因此,显而易见的是,本发明仅受权利要求书以及其等同物所限制。

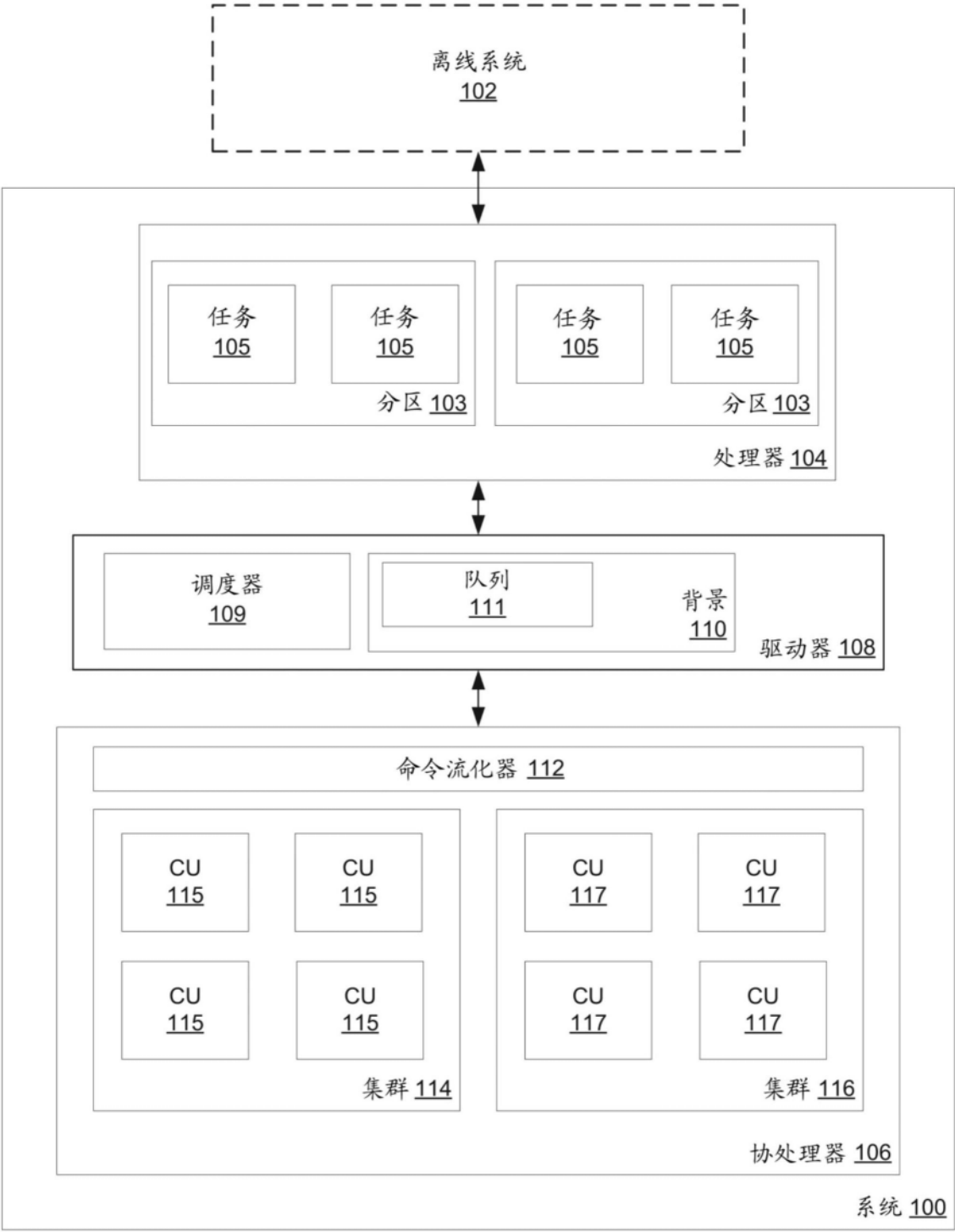


图1A

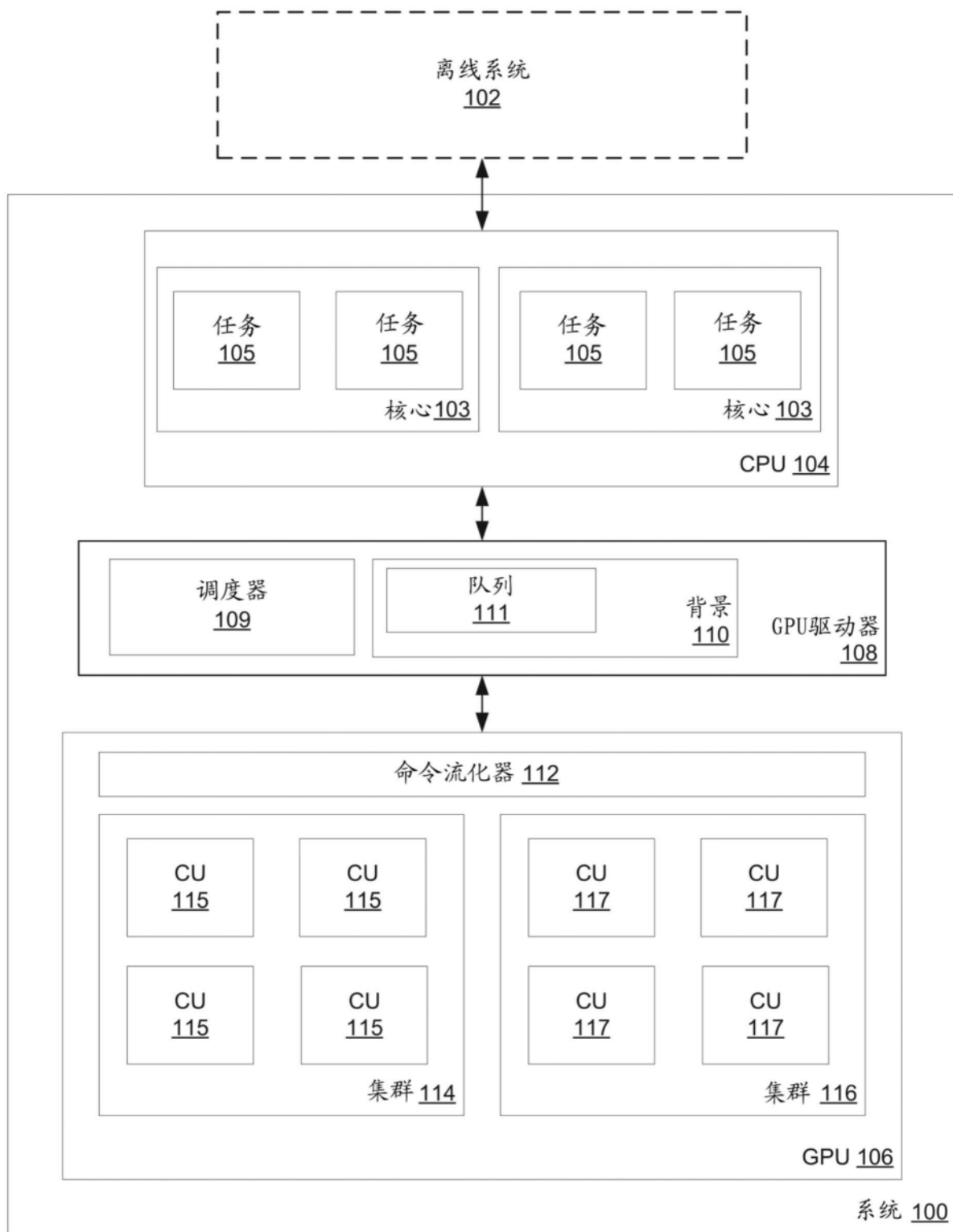


图1B

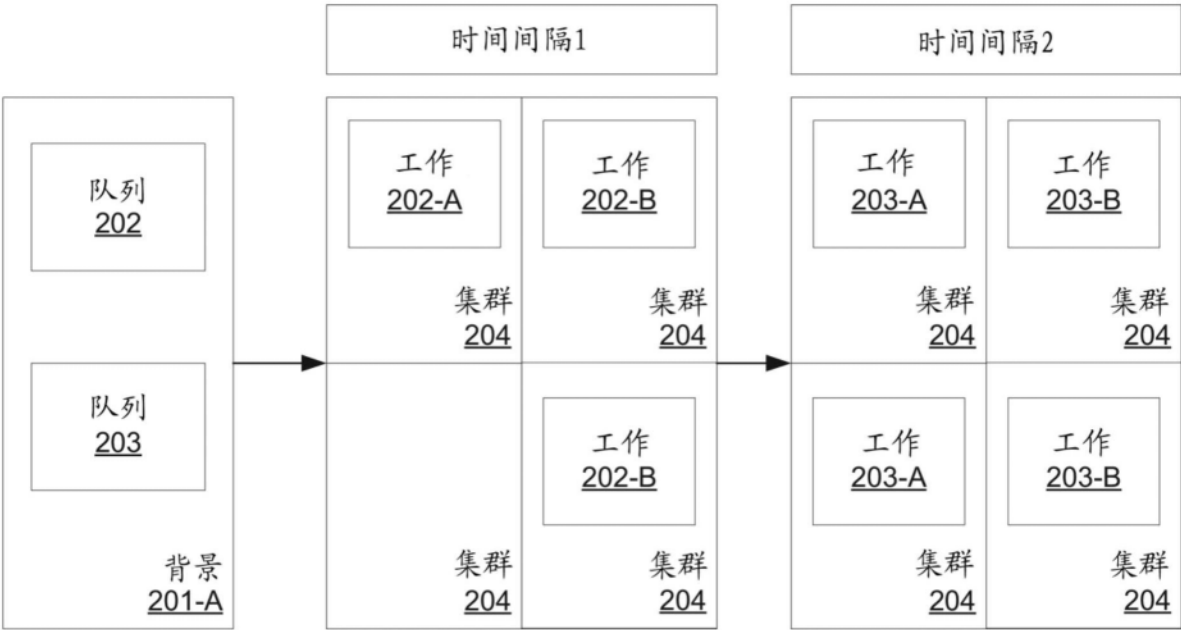


图2A

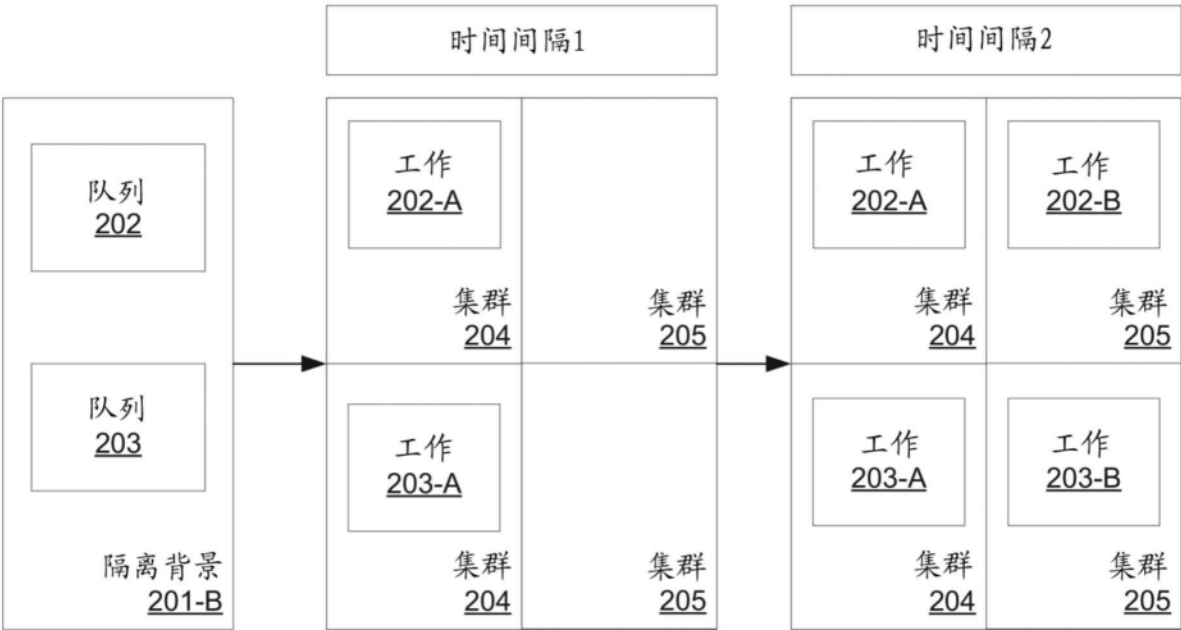


图2B

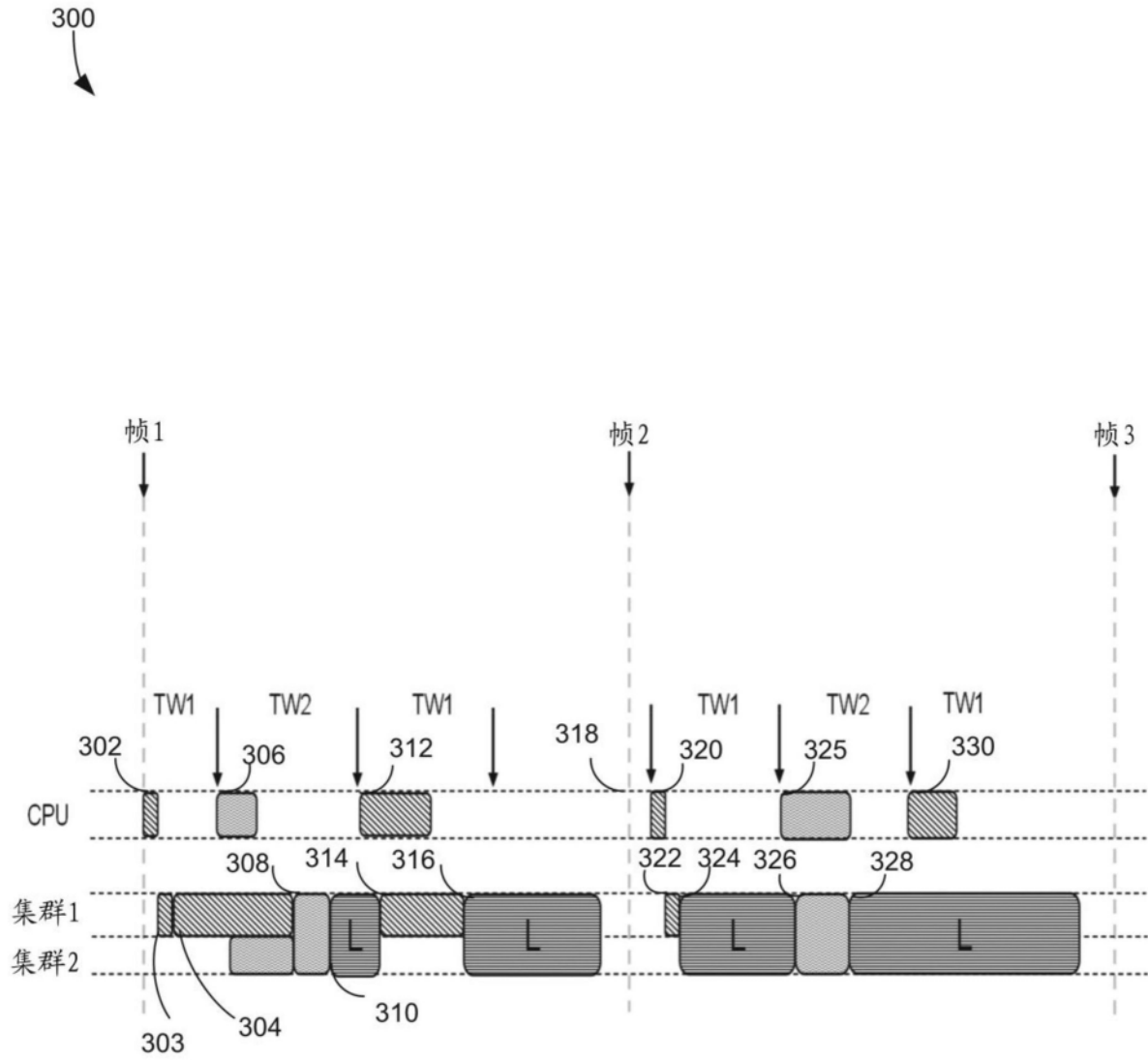


图3

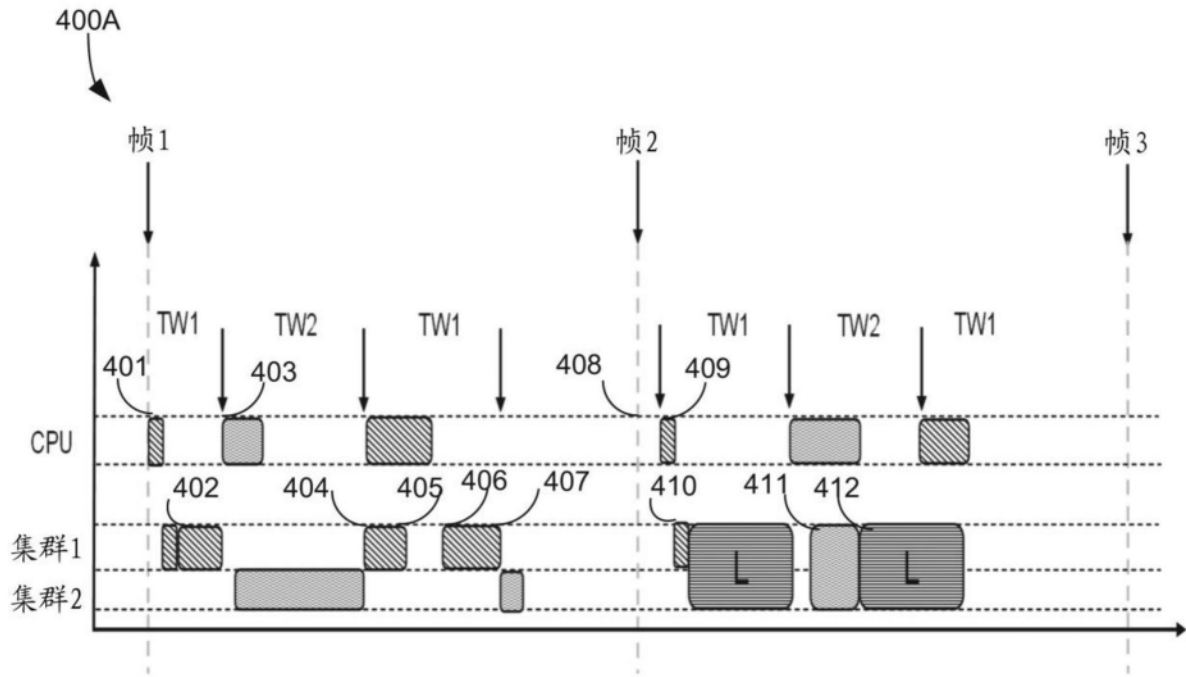


图4A

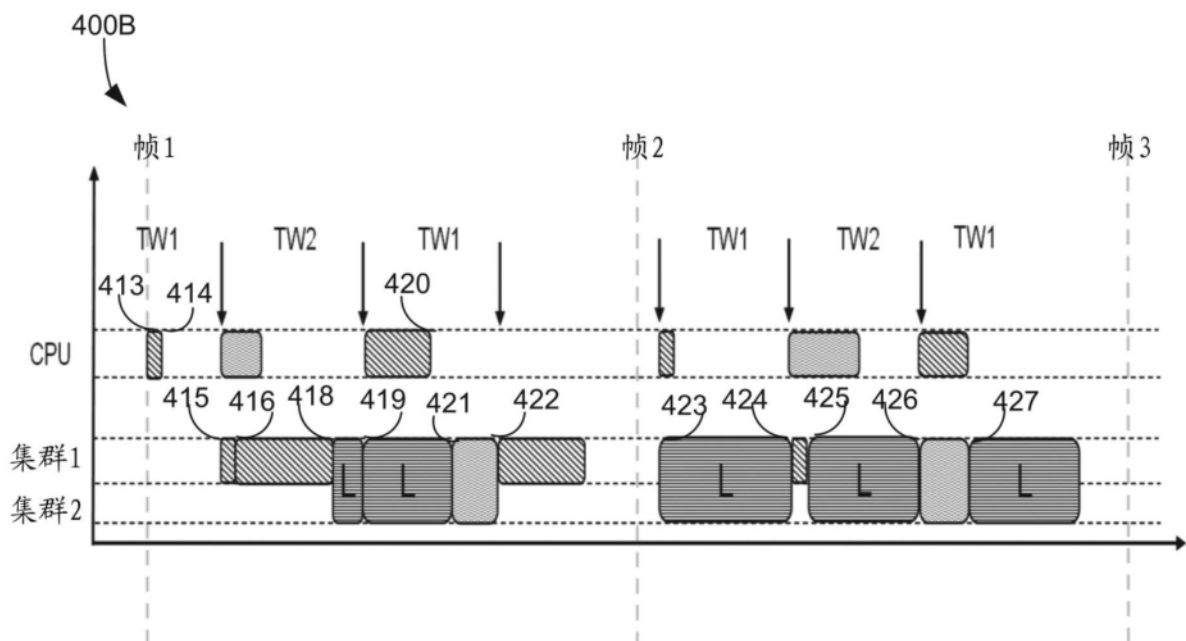


图4B

500A

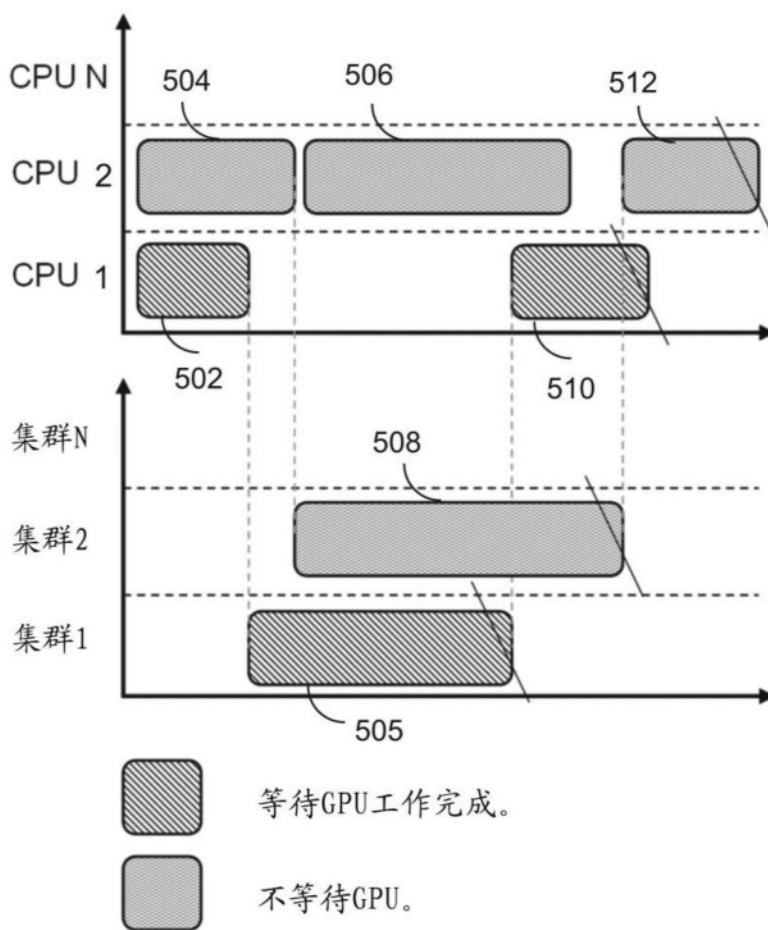


图5A

500B

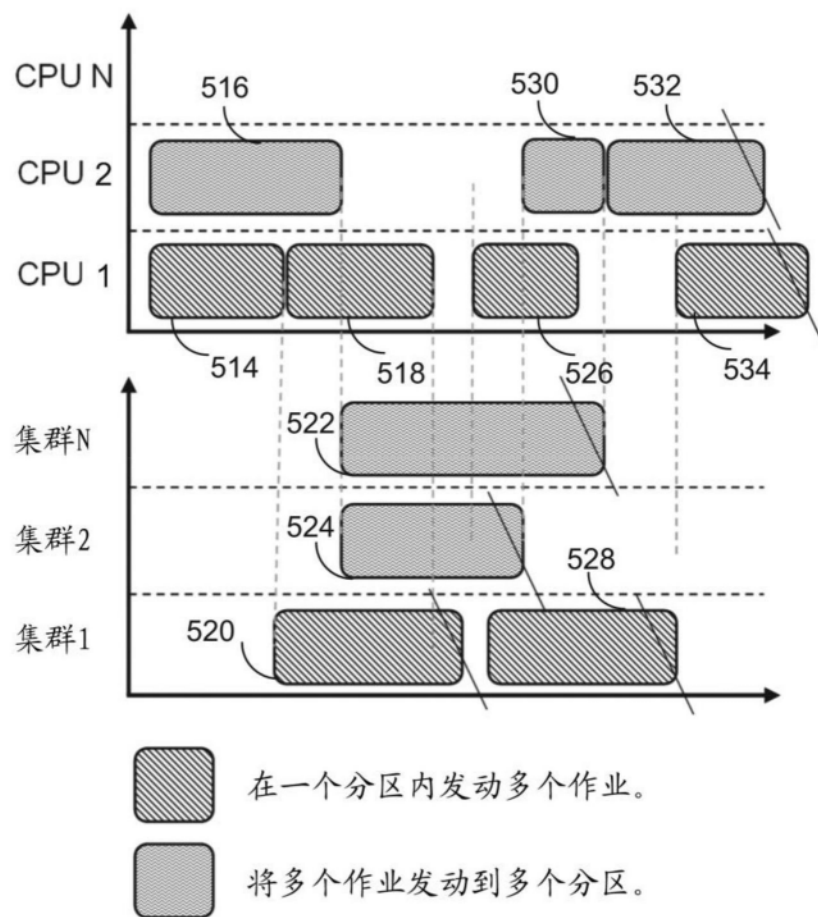


图5B

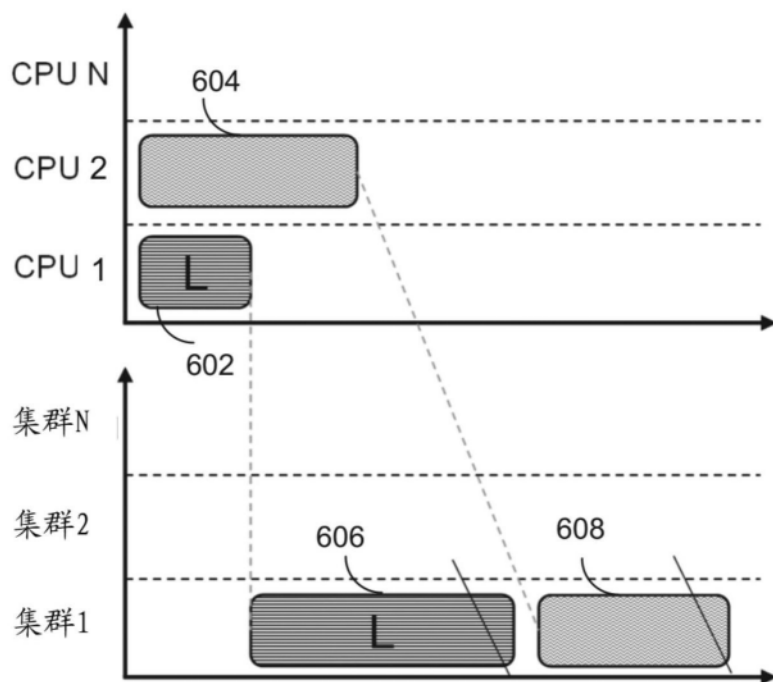


图6A

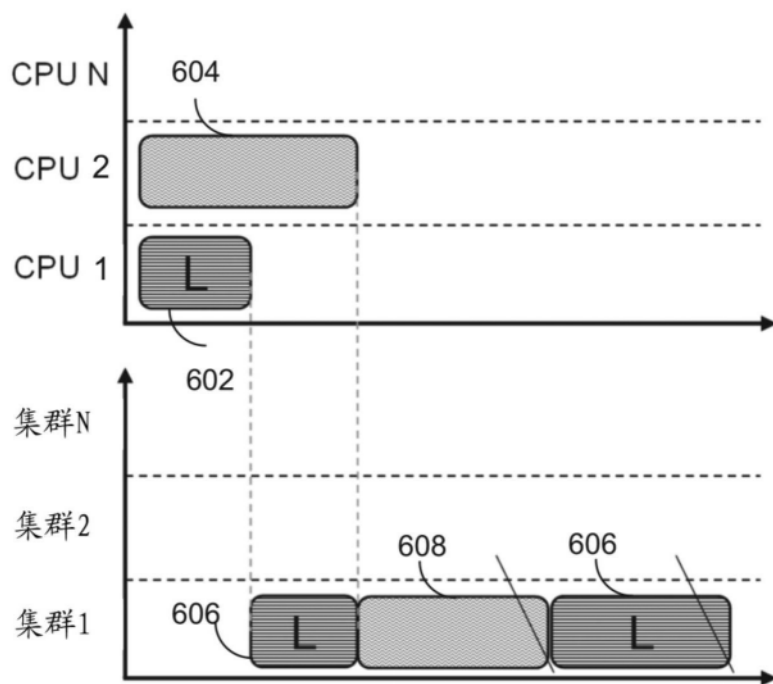


图6B

700

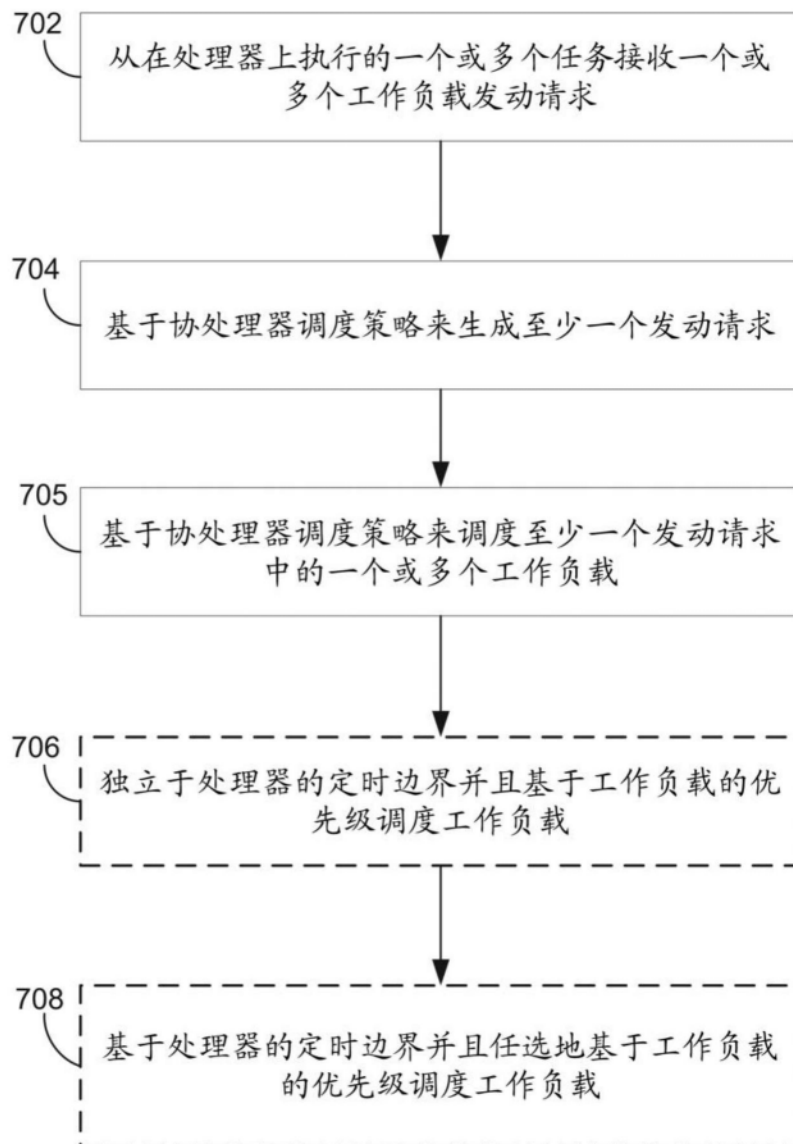


图7

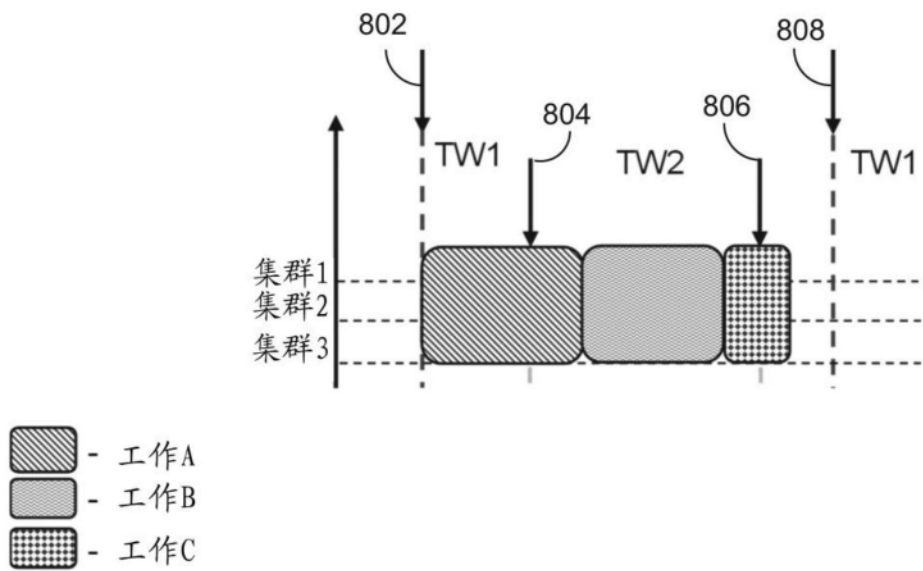


图8A

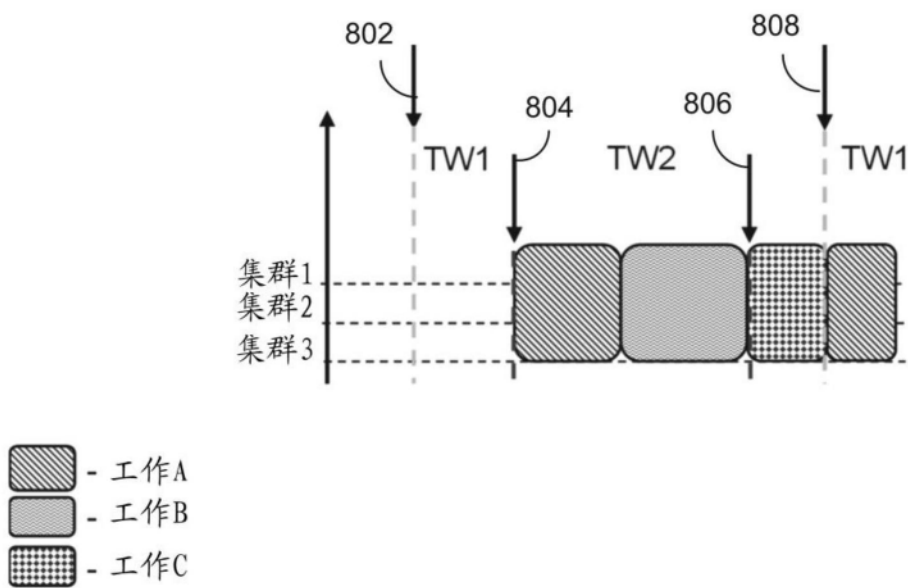


图8B

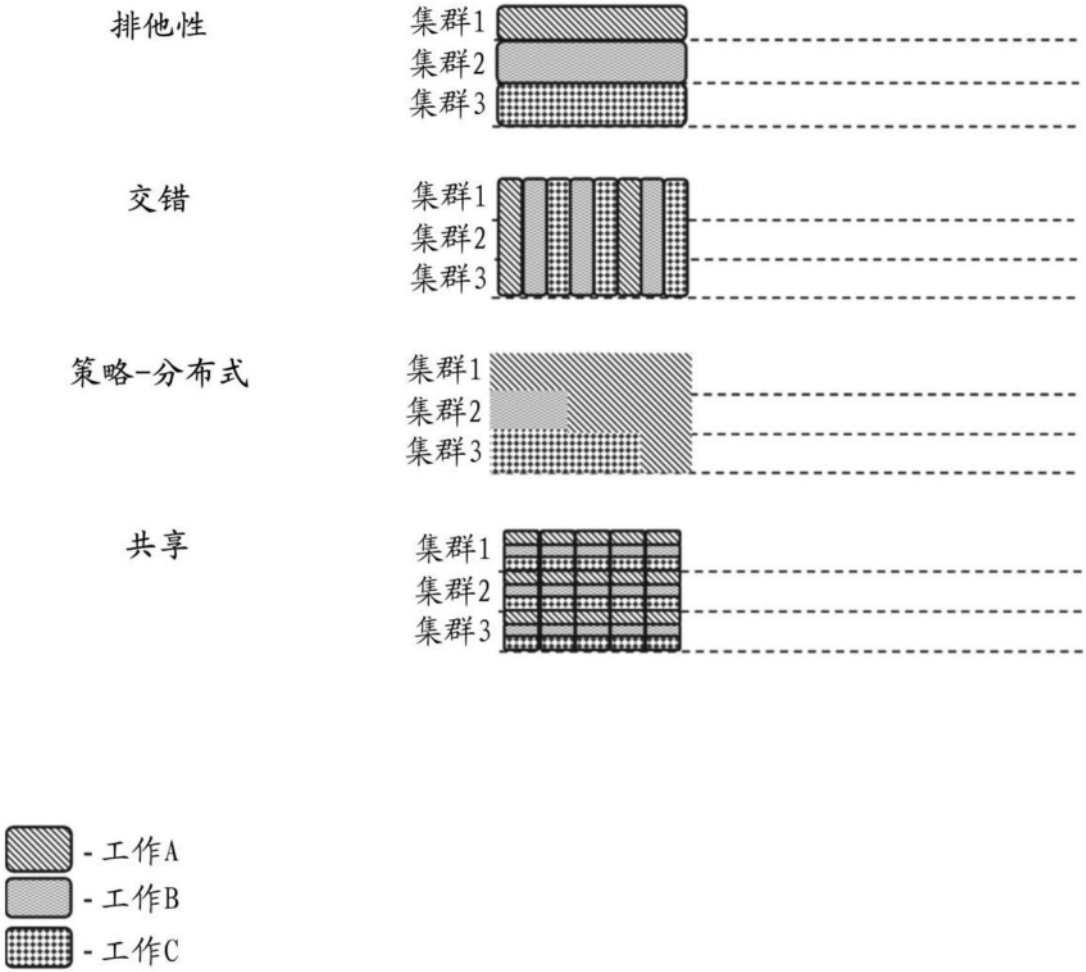


图9

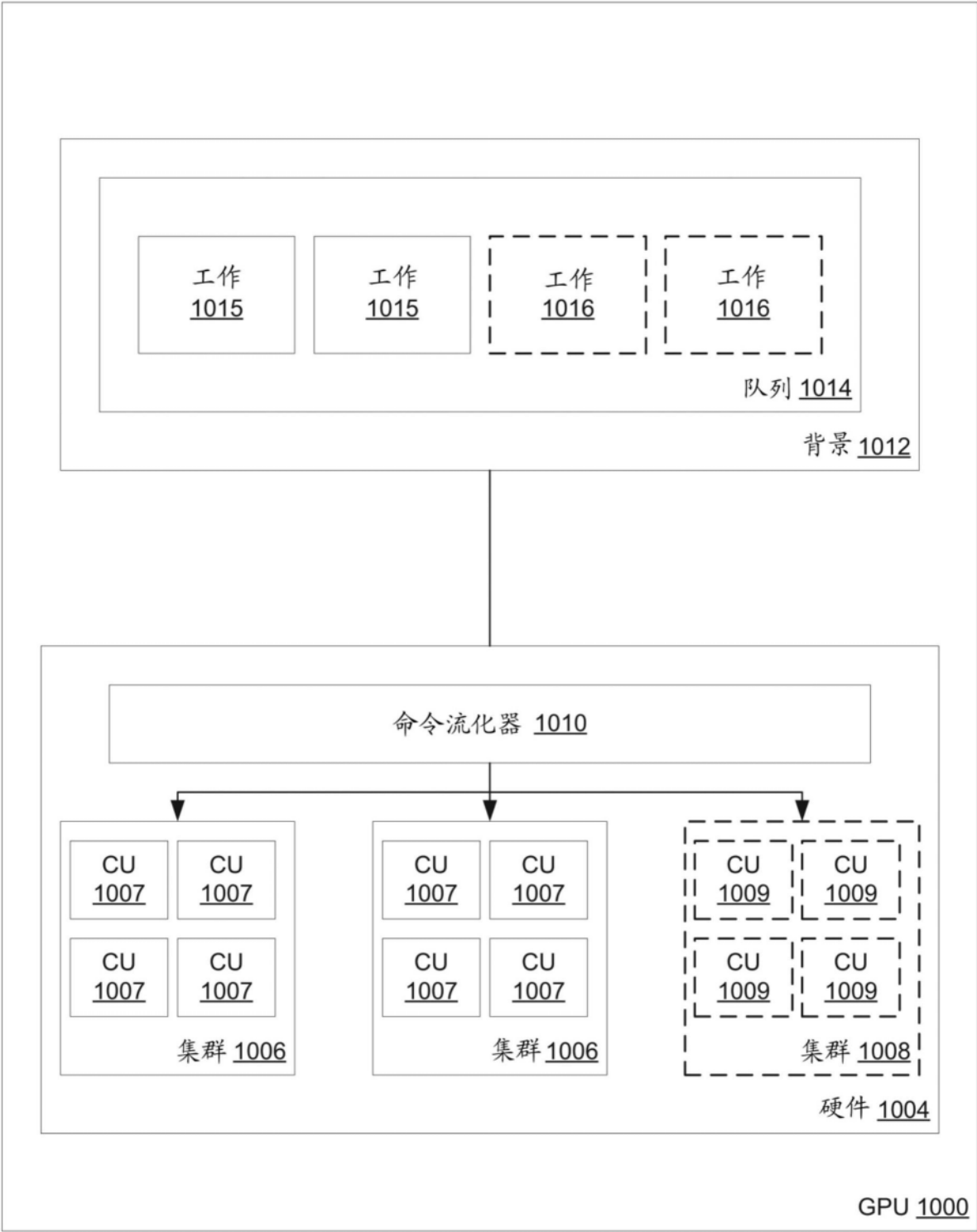


图10A

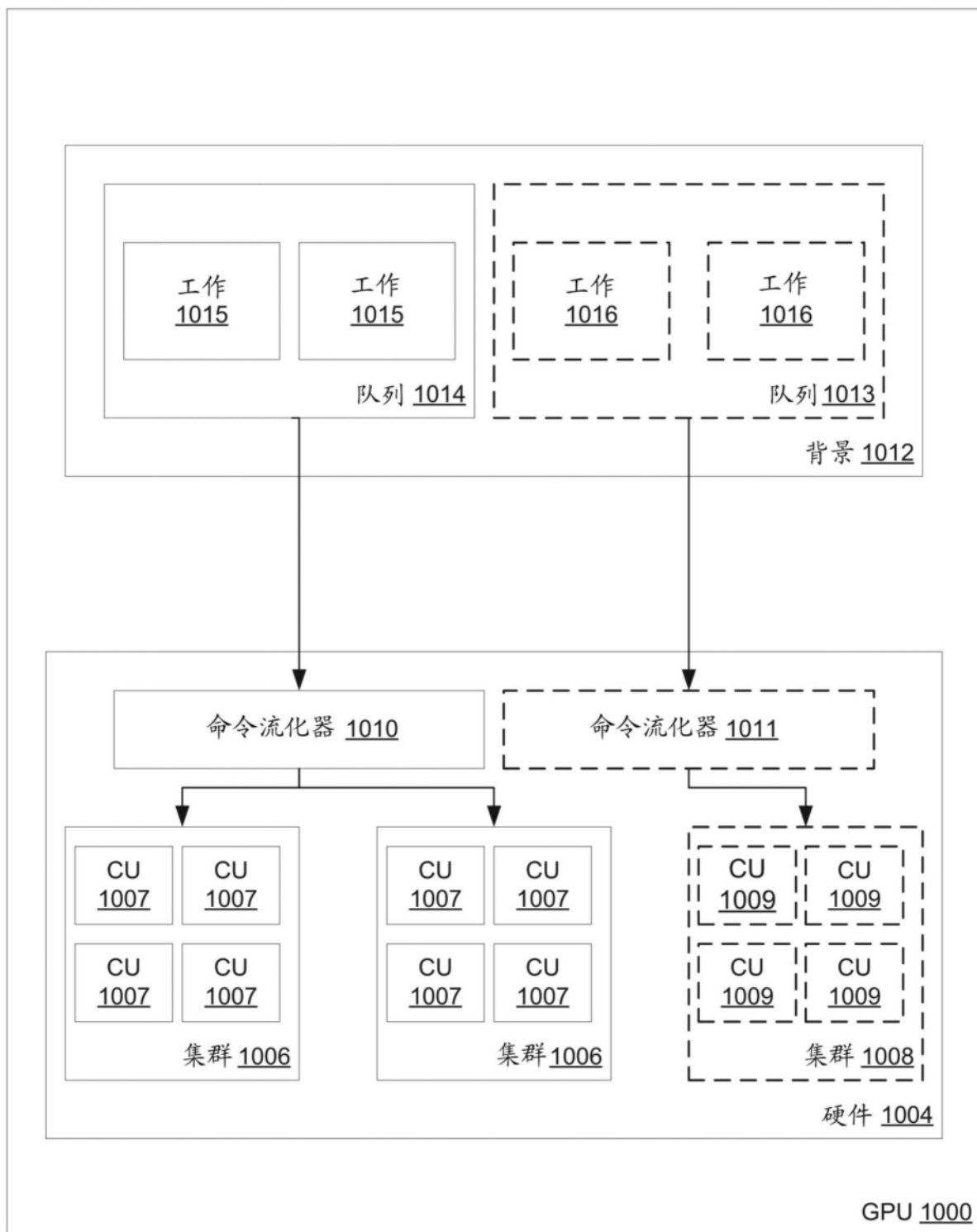


图10B

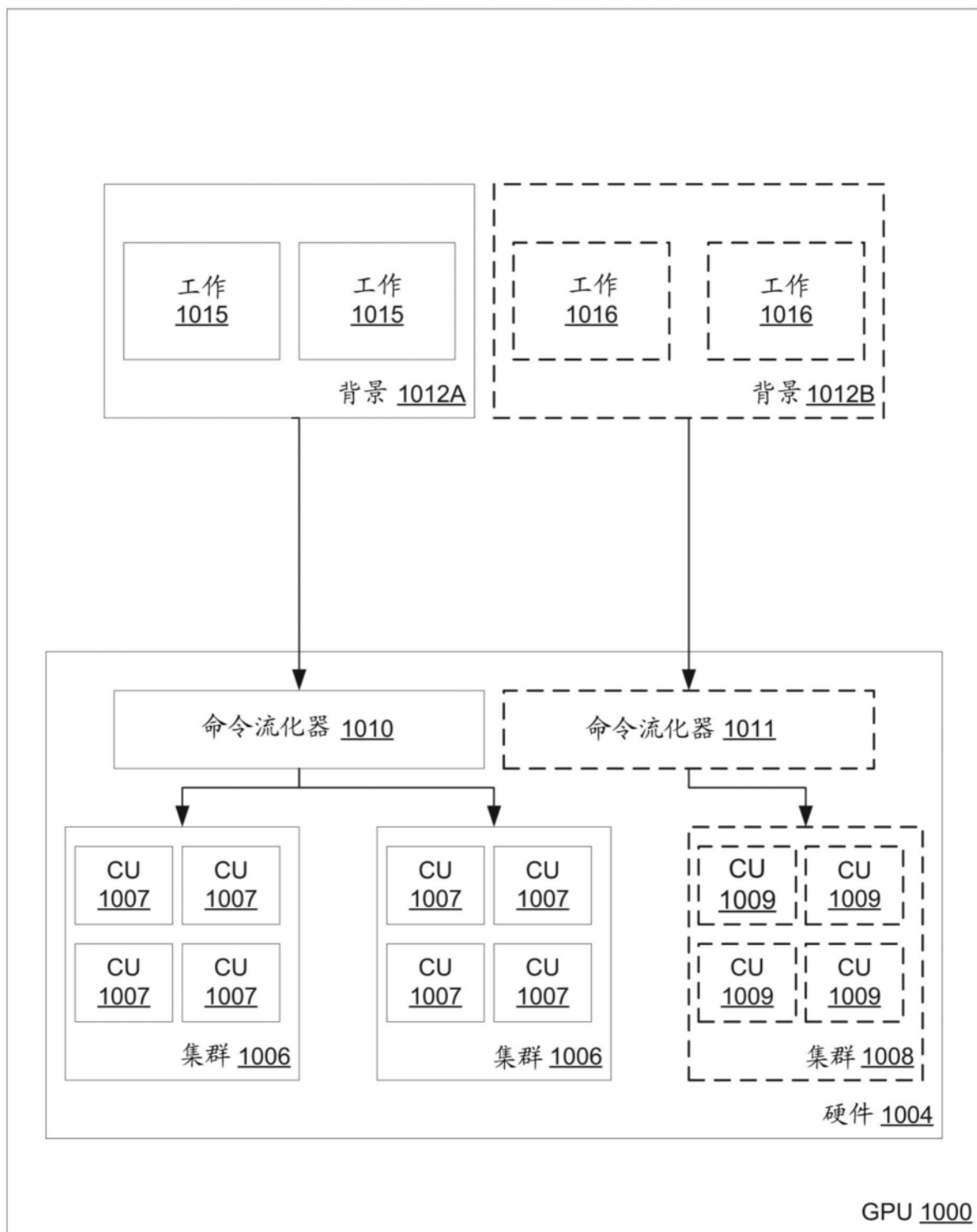


图10C

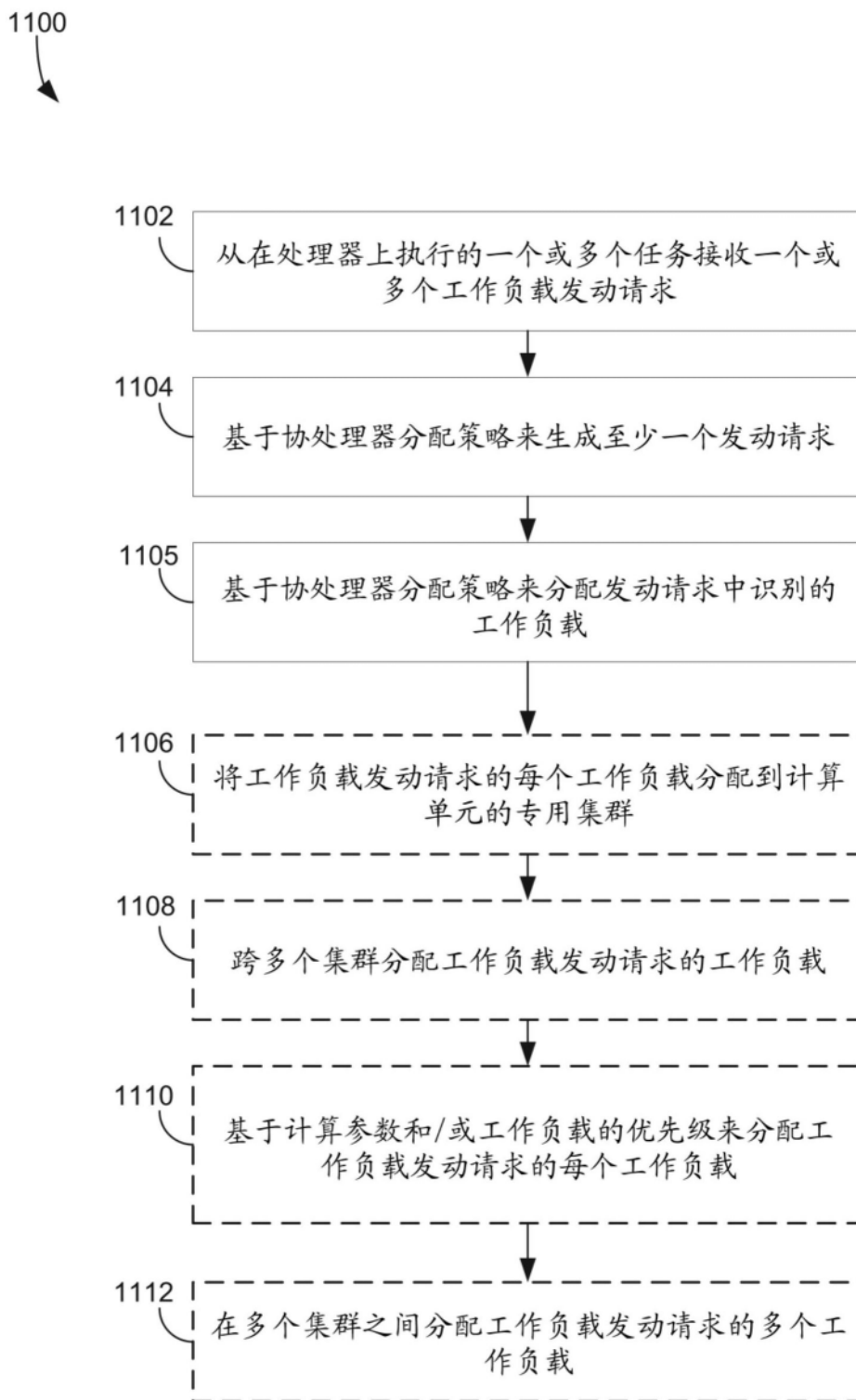


图11

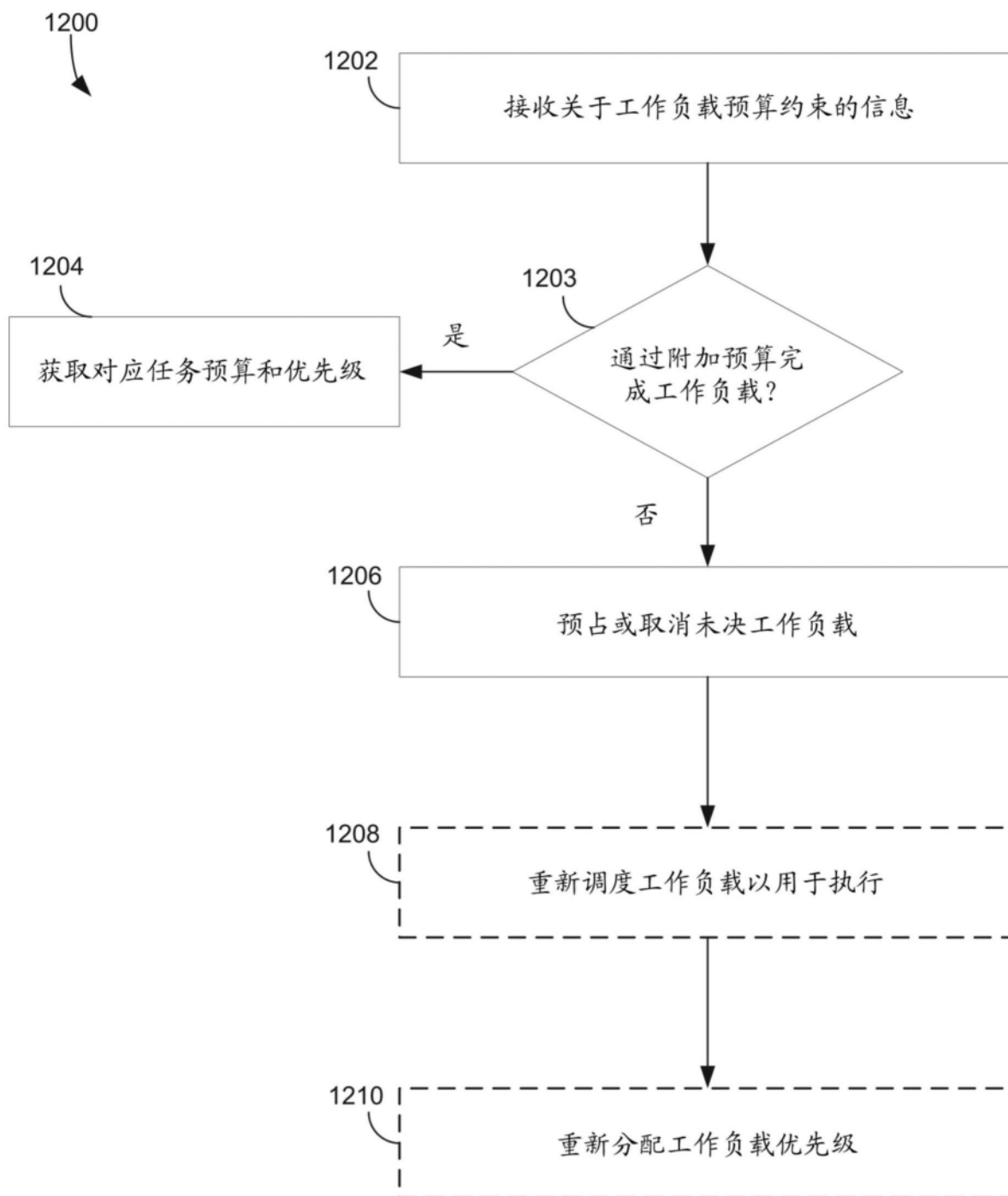


图12

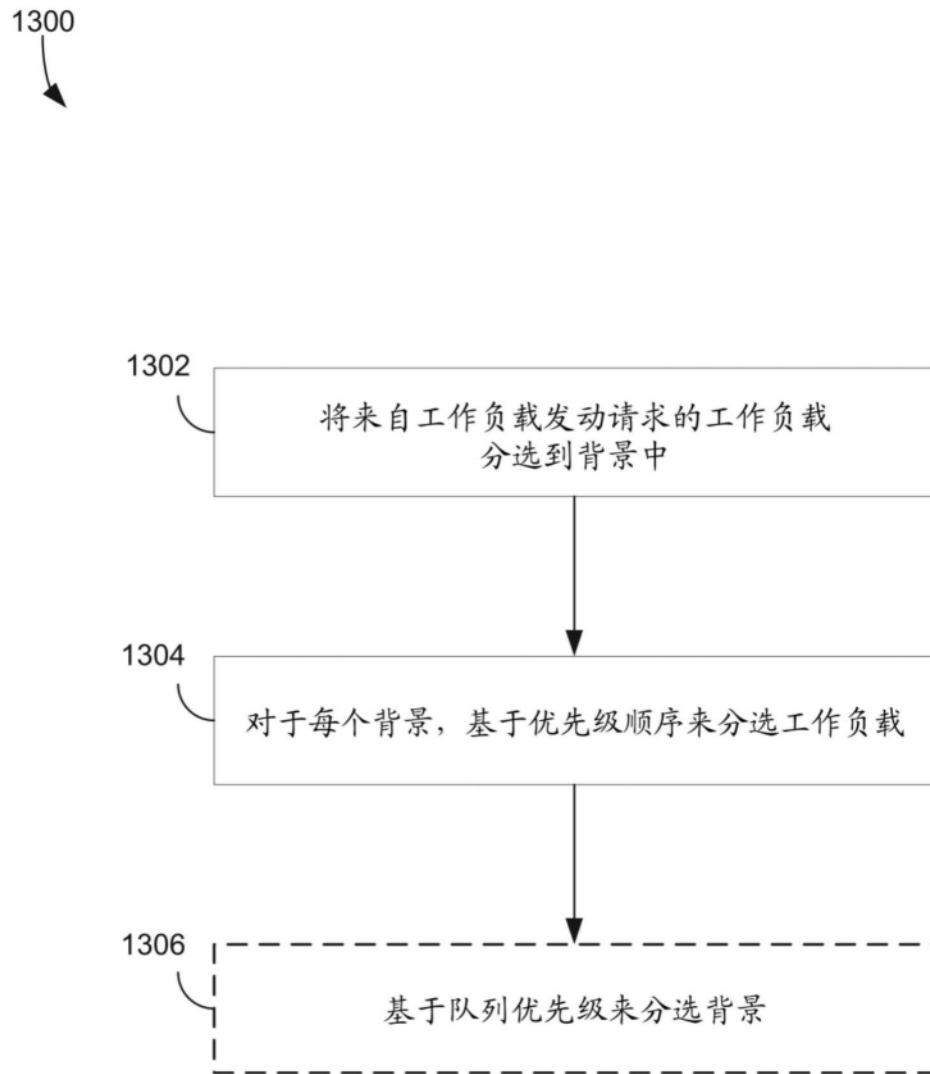


图13

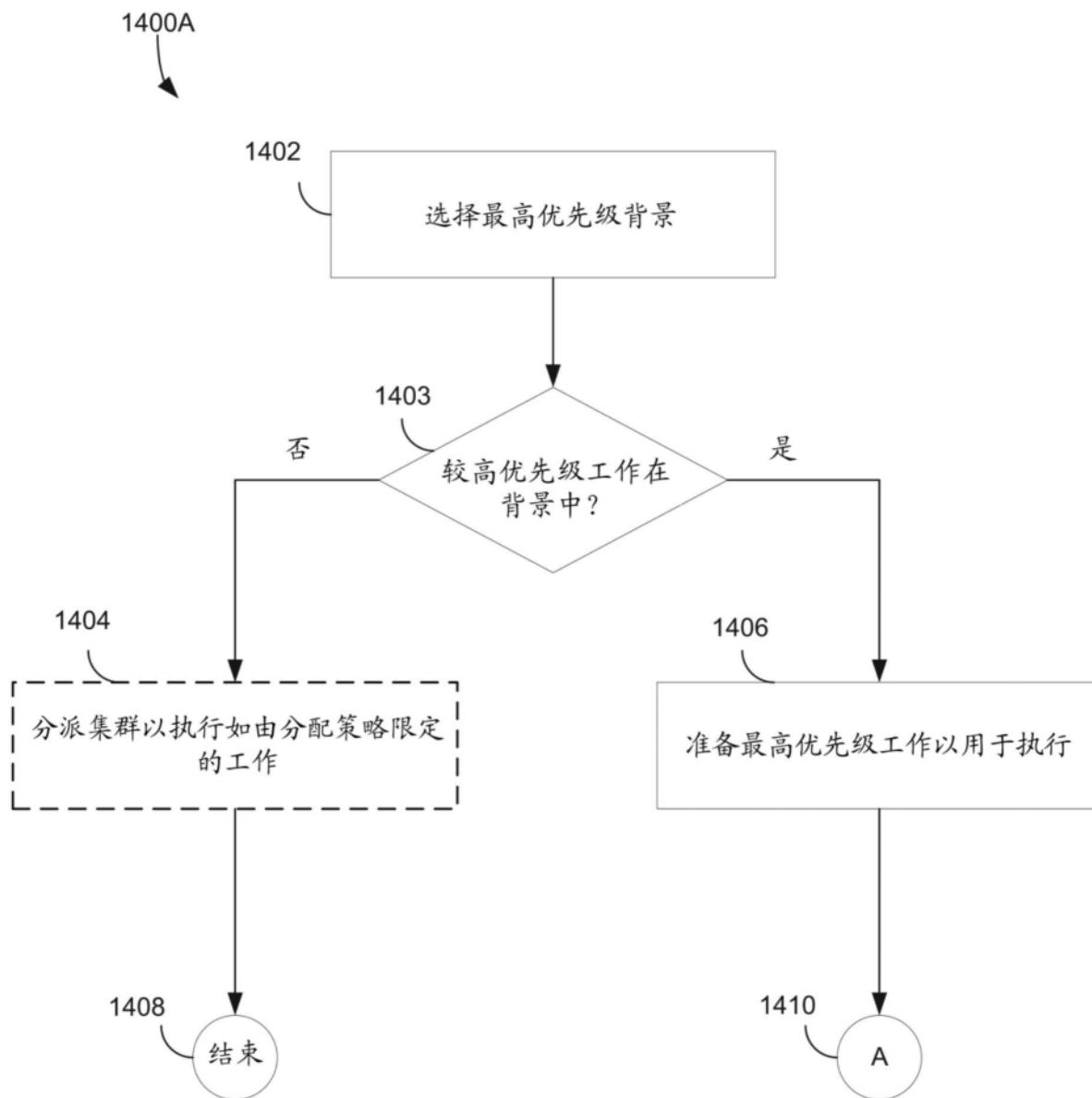


图14A

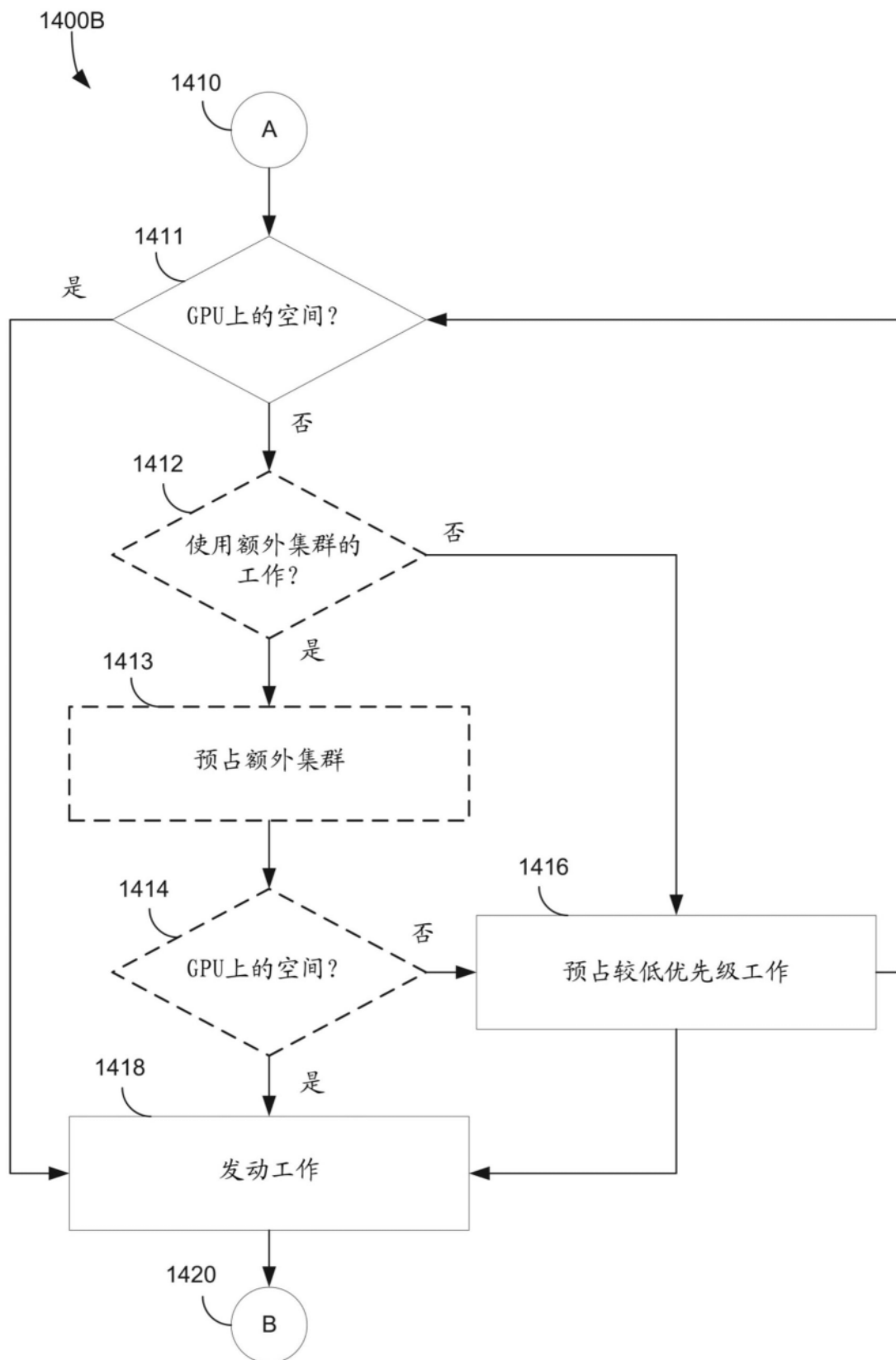


图14B

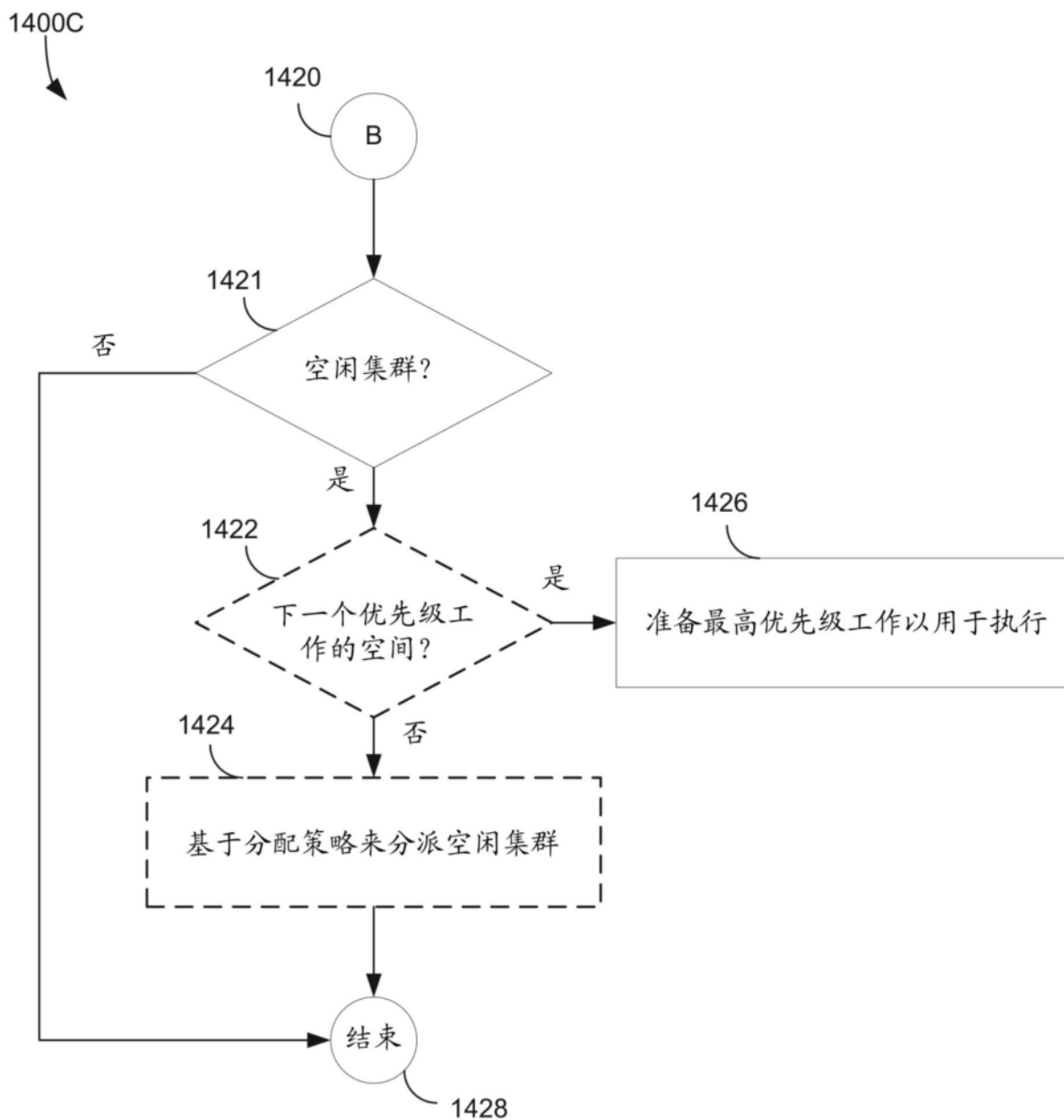


图14C