一种加速器调度方法及相关装置

本申请要求于2022年08月15日提交中国专利局、申请号为202210977273.6、申请名称为“一种加速器调度方法及相关装置”的中国专利申请的优先权，其全部内容通过引用结合在本申请中。

技术领域

本申请涉及网络通信技术领域，尤其涉及一种加速器调度方法及相关装置。

背景技术

近年来，中央处理器（Central Processing Unit，CPU）通用计算能力的发展放缓，而机器学习和Web服务等新兴业务需要的运算能力却成指数级增长。随着硬件技术的成熟发展，CPU无论是在计算能力，还是资源成本上相对于一些专用加速硬件已经没有绝对优势，这也促使硬件加速技术得到各大公司的青睐。

目前，因特尔提出了快速辅助技术（QuickAssist Technology，QAT），即一种高性能数据安全和压缩的加速方案，该方案利用QAT芯片分担对称/非对称加密计算，无损压缩等大计算量的任务，来降低CPU使用率并提高整体系统性能。具体的，将一个加速器设备抽象成一个逻辑加速器，QAT支持一个逻辑加速器内的加速单元单实例、或者配置简单的环状轮询负载均衡。但由于当前QAT只支持对单加速器设备的操作，即应用进程初始化时就必须从多个加速器中选定此后一直使用的目标加速器，进而无论应用进程在任何处理核上运行，任务都会被调度至该目标加速器进行处理。在此过程中，会存在应用进程间争抢加速器，导致加速器负载不均衡，加速器性能降低等问题。

因此，如何提供一种加速器调度方法及相关装置，以提升系统的计算性能，是亟待解决的问题。

发明内容

本申请实施例所要解决的技术问题在于，提供一种加速器调度方法及相关装置，以提升系统的计算性能。

第一方面，本申请实施案例提供一种加速器调度方法，其特征在于，所述方法包括：接收第一请求，所述第一请求用于请求配置第一任务的目标加速器，所述第一任务为第一进程中当前运行的任务，所述第一进程运行在处理器中，所述处理器包括多个处理核，所述第一任务包括多个子任务，每个所述子任务与所述多个处理核中的一个对应；响应于所述第一请求，基于所述第一任务中的第一个子任务对应的处理核，从N个加速器中确定目标加速器，N为大于0的整数；将所述第一任务调度至所述目标加速器进行处理。

在本发明实施例中，可以为应用进程中当前运行的任务从多个加速器中选择一个当前最优的作为目标加速器，并可以将任务调度至目标加速器进行处理，从而提升系统计算性能。具体的，应用进程用于运行应用服务，应用服务中可以包括多个任务，每个任务中又可以包括多个子任务，对于应用进程当前运行的任务来说，根据任务对应的处理核，可以从多个加速器中选择一个与该处理核亲和度较高的作为目标加速器（亲和度越高的加速器能够更加快速的访问到该处理核对应的内存储器），从而可以将任务调度至目标加速器进行处理。而在现有技术中，应用服务初始化时就必须从多个加速器中选定此后一直使用的目标加速器，进而在应用进程上无论运行任何任务，任务都会被调度至该目标加速器进行处理。但在该过程中，不同任务可能在不同的处理核上运行，不同的处理核与该目标加速器的亲和度不同，当任务对应的处理核与该目标加速器的亲和度较低时，目标加速器无法快速地访问到该处理核对应的内存储器，导致计算效率下降等问题。综上，在本申请中，针对应用服务中的多个任务，在运行时都可以根据任务对应的处理核，为每个任务从多个加速器中选择一个最优的作为目标加速器，避免将任务调度至亲和度较低的加速器上处理，导致计算效率下降的问题，从而提升了系统计算性能。

在一种可能的实现方式中，所述基于所述第一任务中的第一个子任务对应的处理核，从N个加速器中确定目标加速器，包括：根据所述第一个子任务对应的处理核和所述N个加速器的实时状态信息，从所述N个加速器中确定所述目标加速器。

在本发明实施例中，可以基于第一任务中的第一个子任务对应的处理核以及多个加速器的实时状态信息（即包括了每个加速器当前状态的信息），从N个加速器中确定与该处理核亲和度较高且当前性能最佳

的为目标加速器，以确保目标加速器能够快速地处理第一任务并且也能够更加快速地访问到该处理核对应的内存储器，从而提升系统计算性能。

在一种可能的实现方式中，所述方法还包括：获取所述实时状态信息，所述实时状态信息中包括所述 N 个加速器中每个所述加速器的负载状态信息，内存访问速度信息，加速器能力值信息中的一种或多种；其中，所述内存访问速度信息用于指示对应加速器访问不同内存储器时分别对应的访问速度，所述加速器能力值信息用于指示对应加速器包括的加速单元个数。

在本发明实施例中，每个加速器设备上线时，都可以获取加速器位置信息并设置 NUMA 亲和度，以确定加速器的内存访问速度信息，该信息用于指示对应加速器访问不同内存储器时分别对应的访问速度；可以获取每个加速器中包括的加速单元个数，以确定加速器能力值信息，即加速单元个数越多，表示加速器计算效率越高；还可以动态获取加速器负载状况，并实时更新加速器负载值。通过本发明实施例，在为第一任务选择目标加速器前，能够及时获取到多个加速器的实时状态信息，以从多个加速器中选择出当前最优的作为目标加速器。进一步地，可以将任务调度至目标加速器进行处理，从而提升系统计算性能。

在一种可能的实现方式中，所述根据所述第一个子任务对应的处理核和所述 N 个加速器的实时状态信息，从所述 N 个加速器中确定所述目标加速器，包括：根据所述第一个子任务对应的处理核和所述 N 个加速器的所述实时状态信息，基于预设评分规则从所述 N 个加速器中确定评分最高的加速器为所述目标加速器。

在本发明实施例中，可以先获取多个加速器的实时状态信息，然后查看第一任务中的第一个子任务对应的处理核信息，并基于处理核信息确定处理核所在的 NUMA 节点，以确定 NUMA 亲和度。进一步地，根据每个加速器对应的 NUMA 亲和度、加速器能力值、加速器负载值，并基于预设评分规则对每个加速器进行综合评分，将综合评分最高的确定为最优加速器。进一步地，可以将任务调度至目标加速器进行处理，从而提升系统计算性能。

在一种可能的实现方式中，所述将所述第一任务调度至所述目标加速器进行处理，包括：判断所述目标加速器的所述负载状态信息对应的负载值是否大于预设值；若所述负载值小于或等于所述预设值，获取所述目标加速器的路径信息，并基于所述目标加速器的路径信息将所述第一任务调度至所述目标加速器进行处理。

在本发明实施例中，将任务调度至目标加速器前，可以先判断该目标加速器当前的负载值是否大于预设值，若目标加速器当前的负载值小于或等于预设值，表示目标加速器处于空闲状态或任务量较少的状态，目标加速器能够及时处理第一任务，进而可以获取目标加速器的路径信息，并基于路径信息将任务调度至目标加速器进行处理，来降低处理器使用率并提升系统计算性能。

在一种可能的实现方式中，所述方法还包括：若所述负载值大于所述预设值，则将所述第一任务调度至所述处理器进行处理。

在本发明实施例中，将任务调度至目标加速器前，可以先判断该目标加速器当前的负载值是否大于预设值，若目标加速器当前的负载值大于预设值，表示目标加速器处于繁忙状态，目标加速器无法及时处理第一任务，或者跨 NUMA、跨片代价过大，进而可以将第一任务还是交给处理器进行处理，以确保系统的计算性能。

在一种可能的实现方式中，所述方法还包括：将所述目标加速器的路径信息与所述第一个子任务对应的处理核，更新至所述第一任务对应的上下文信息。

在本发明实施例中，当确定了任务对应的目标加速器后，可以确定目标加速器的路径信息和处理核信息，并将目标加速器的路径信息和处理核信息更新至任务对应的上下文信息中，以便后续可以根据上下文信息中的目标加速器的路径信息，将任务中的子任务调度至目标加速器进行处理，以及根据上下文信息中的处理核信息访问对应的内存储器，从而提升系统计算性能。

在一种可能的实现方式中，所述方法还包括：在执行所述多个子任务中的第 i 个子任务时，从所述上下文信息中，获取所述目标加速器的路径信息，并将所述第 i 个子任务调度至所述目标加速器进行处理，i 为大于 1 的整数。

在本发明实施例中，执行任务中的子任务时，可以获取任务对应的上下文信息，并基于上下文信息中的目标加速器的路径信息将子任务调度至目标加速器进行处理，从而提升系统计算性能。

第二方面，本申请提供了一种加速器调度装置，其特征在于，所述加速器调度装置包括任务管理单元，所述任务管理单元，用于：接收第一请求，所述第一请求用于请求配置第一任务的目标加速器，所述第一

任务为第一进程中当前运行的任务，所述第一进程运行在处理器中，所述处理器包括多个处理核，所述第一任务包括多个子任务，每个所述子任务与所述多个处理核中的一个对应；响应于所述第一请求，基于所述第一任务中的第一个子任务对应的处理核，从 N 个加速器中确定目标加速器，N 为大于 0 的整数；将所述第一任务调度至所述目标加速器进行处理。

在一种可能的实现方式中，所述任务管理单元，具体用于：根据所述第一个子任务对应的处理核和所述 N 个加速器的实时状态信息，从所述 N 个加速器中确定所述目标加速器。

在一种可能的实现方式中，所述加速器调度装置还包括加速器管理单元，所述加速器管理单元，用于：向所述任务管理单元发送所述实时状态信息，所述实时状态信息中包括所述 N 个加速器中每个所述加速器的负载状态信息，内存访问速度信息，加速器能力值信息中的一种或多种；其中，所述内存访问速度信息用于指示对应加速器访问不同内存储器时分别对应的访问速度，所述加速器能力值信息用于指示对应加速器包括的加速单元个数。

在一种可能的实现方式中，所述任务管理单元，具体用于：根据所述第一个子任务对应的处理核和所述 N 个加速器的所述实时状态信息，基于预设评分规则从所述 N 个加速器中确定评分最高的加速器为所述目标加速器。

在一种可能的实现方式中，所述任务管理单元，具体用于：判断所述目标加速器的所述负载状态信息对应的负载值是否大于预设值；若所述负载值小于或等于所述预设值，获取所述目标加速器的路径信息，并基于所述目标加速器的路径信息将所述第一任务调度至所述目标加速器进行处理。

在一种可能的实现方式中，所述任务管理单元，还用于：若所述负载值大于所述预设值，则将所述第一任务调度至所述处理器进行处理。

在一种可能的实现方式中，所述任务管理单元，还用于：将所述目标加速器的路径信息与所述第一个子任务对应的处理核，更新至所述第一任务对应的上下文信息。

在一种可能的实现方式中，所述任务管理单元，还用于：在执行所述多个子任务中的第 i 个子任务时，从所述上下文信息中，获取所述目标加速器的路径信息，并将所述第 i 个子任务调度至所述目标加速器进行处理，i 为大于 1 的整数。

第三方面，本申请提供了一种计算机存储介质，其特征在于，所述计算机存储介质存储有计算机程序，该计算机程序被处理器执行时实现上述第一方面任意一项所述的方法。

第四方面，本申请实施例提供一种电子设备，该电子设备中包括处理器，处理器被配置为支持该电子设备实现第一方面提供的一种加速器调度方法中相应的功能。该电子设备还可以包括存储器，存储器用于与处理器耦合，其保存该电子设备必要的程序指令和数据。该电子设备还可以包括通信接口，用于该电子设备与其他设备或通信网络通信。

第五方面，本申请提供了一种芯片系统，该芯片系统包括处理器，用于支持电子设备实现上述第一方面中所涉及的功能，例如，生成或处理上述加速器调度方法中所涉及的信息。在一种可能的设计中，所述芯片系统还包括存储器，所述存储器，用于保存电子设备必要的程序指令和数据。该芯片系统，可以由芯片构成，也可以包含芯片和其他分立器件。

第六方面，本申请提供一种计算机程序，其特征在于，所述计算机程序包括指令，当所述计算机程序被计算机执行时，使得所述计算机执行上述第一方面中任意一项所述的方法。

附图说明

图 1 为本发明实施例提供的一种主机的系统架构示意图。

图 2 是本申请实施例中的一种加速器调度方法的流程示意图。

图 3 为本发明实施例提供的一种软件运行在主机上的示意图。

图 4 为本发明实施例提供的一种软件架构的示意图。

图 5 为本发明实施例提供的一种任务管理的示意图。

图 6 为本发明实施例提供的一种确定目标加速器的示意图。

图 7 为本发明实施例提供的一种加速器管理器的示意图。

图 8 为本发明实施例提供的一种任务管理器确定目标加速器示意图。

图 9 为本发明实施例提供的一种联合多种设备调度任务的示意图。

图 10 是本发明实施例提供的本申请提供了一种加速器调度装置的结构示意图。

具体实施方式

下面将结合本申请实施例中的附图，对本申请实施例进行描述。

本申请的说明书和权利要求书及所述附图中的术语“第一”、“第二”、“第三”和“第四”等是用于区别不同对象，而不是用于描述特定顺序。此外，术语“包括”和“具有”以及它们任何变形，意图在于覆盖不排他的包含。例如包含了一系列步骤或单元的过程、方法、系统、产品或设备没有限定于已列出的步骤或单元，而是可选地还包括没有列出的步骤或单元，或可选地还包括对于这些过程、方法、产品或设备固有的其它步骤或单元。

在本文中提及“实施例”意味着，结合实施例描述的特定特征、结构或特性可以包含在本申请的至少一个实施例中。在说明书中的各个位置出现该短语并不一定均是指相同的实施例，也不是与其它实施例互斥的独立的或备选的实施例。本领域技术人员显式地和隐式地理解的是，本文所描述的实施例可以与其它实施例相结合。

基于上述，本发明实施例提供一种主机。请参见图 1，图 1 为本发明实施例提供的一种主机的系统架构示意图，该主机可以包括但不限于处理器 101，内存储器 102，加速器 103，其中，

处理器 101，可以包括一个或多个处理单元（也可称处理核），例如：处理器 101 可以包括中央处理单元（Central Processing Unit，CPU）、应用处理单元（application processor，AP）、调制解调处理单元、图形处理单元（graphics processing unit，GPU）、图像信号处理单元（image signal processor，ISP）、视频编解码单元、数字信号处理单元（digital signal processor，DSP）、基带处理单元和神经网络处理单元（neural-network processing unit，NPU）等中的一个或多个。其中，不同的处理单元可以是独立的器件，也可以集成在一个或多个器件中。

在非一致性内存访问架构（Non Uniform Memory Access，NUMA）中，每个处理器 101 中可以包括多个 NUMA 节点，每个 NUMA 节点由处理器 101 中的计算单元（也可以是处理核）组成，不同 NUMA 节点之间可以通过总线进行连接。其中，离 NUMA 节点较近的内存储器 102，通常称为为本地内存；离 NUMA 节点较远的内存储器 102，通常称为远端内存。NUMA 节点可以直接的访问到本地内存，但无法直接访问远端内存。

处理器 101，可运行操作系统、文件系统（如闪存文件系统）、或应用进程等，以控制连接到处理器 101 的多个硬件或软件元件，并且可处理各种数据并执行操作。处理器 101 可将外存储器（如磁盘）中存储的指令或数据加载到内存储器 102 中，并把需要运算的指令或数据调到处理器 101 中进行运算，当运算完成后处理器 101 再将结果临时存储在内存储器 102 中，并将需要长期存储的指令或数据通过控制器存储至外存储器中。例如，CPU0 中的 NUMA 节点 0 可以将外存储器中存储的指令或数据加载到本地内存中，并把需要运算的指令或数据调到处理器 101 中进行运算，当运算完成后处理器 101 再将结果临时存储在本地内存中。在本发明实施例中，由于处理器 101 计算性能有限，处理器 101 也可以将大计算量的任务分配给加速器 103（可以由专用芯片组成，用于处理特定任务）进行处理，如利用 QAT 芯片分担对称/非对称加密计算，无损压缩等大计算量的任务，来降低处理器 101 使用率并提高系统性能。后续将详细描述，如何确定目标加速器，以及如何将任务调度至目标加速器进行处理，在此先不赘述。

内存储器 102，通常为掉电易失性存储器，断电时会丢失其上存储的内容，也可称为内存（Memory）或主存储器。本申请中的内存储器 102 包括可读可写的运行内存，其作用是用于暂时存放处理器 101 或加速器 103 中的运算数据，以及与外存储器或其他外部存储器交互数据，可作为操作系统或其他正在运行中的程序的临时数据的存储媒介。

在 NUMA 架构中，处理器 101 中的 NUMA 节点可以将所需的数据存储至本地内存中，以便于 NUMA 节点或者亲和度较高的加速器 103 能够快速获取到所需的数据。例如，假设加速器 103 为 QAT 芯片，且 CPU0 中的 NUMA 节点 0 与 QAT 芯片建立连接，NUMA 节点 0 可以将自身的计算任务调度至加速器 103 进行处理，在此过程中，可以先将执行任务所需的数据或指令加载到本地内存中，然后加速器 103 可以直接访问本地内存获取所需的数据或指令，并执行计算任务，从而降低处理器 101 使用率并提高系统性能。

内存储器 102 可以包括，动态随机存取存储器（DRAM）、静态随机存取存储器（SRAM）、同步动态随机存取存储器（SDRAM）等中的一种或多种。其中，DRAM 又包括双倍速率同步动态随机存储器（Double Data Rate Synchronous Dynamic Random Access Memory，DDR SDRAM）简称 DDR、二代双倍速率同步动态随机存储器（DDR2）、三代双倍速率同步动态随机存储器（DDR3）、四代低功耗双倍数据率同步动态随机存储器（Low Power Double Data Rate 4，LPDDR4）、和五代低功耗双倍数据率同步动态随机存储器（Low

Power Double Data Rate 5，LPDDR5）等。

可以理解的是，图 1 中的主机的系统架构只是本发明实施例提供的一些示例性的实施方式，本发明实施例中的主机的系统架构包括但不仅限于以上实现方式。

下面对本发明实施例所基于的具体方法架构进行描述。

参见图 2，图 2 是本申请实施例中的一种加速器调度方法的流程示意图，下面将结合附图 2 对本申请实施例中的加速器调度方法进行描述。需要说明的是，为了更详细的描述本申请实施例中的加速器调度方法，本申请在各个流程步骤中描述了相应的执行主体分别为上述图 1 中的主机，但不代表本申请实施例只能通过所描述的执行主体进行对应的方法流程。

步骤 S201：主机接收第一请求。

具体的，第一请求可以理解为针对第一任务发起的初始应用请求，所述第一请求用于请求配置第一任务的目标加速器，所述第一任务为第一进程中当前运行且还未与加速器进行关联的任务，所述第一进程运行在处理器中，所述处理器包括多个处理核，所述第一任务包括多个子任务，每个所述子任务与所述多个处理核中的一个对应。如图 3 所示，图 3 为本发明实施例提供的一种软件运行在主机上的示意图，图中第一进程可以用于运行一个应用服务，该应用服务中可以存在多个任务，如在 t0 时第一进程可以运行应用服务中的任务 1（如压缩任务），在 t1 时第一进程可以运行应用服务中的任务 2（如解压缩任务），可以将第一进程当前正在运行且还未与加速器进行关联的任务确定为第一任务。其中，第一任务中又可以包括多个子任务，每个子任务可以根据操作系统的调度在相同或不同的处理核上运行，如 t0 时任务 1 的第一个子任务在 CPU0 的 NUMA 节点 0 中的处理核 0 上运行，之后任务 1 的其他子任务可以一直在该处理核上运行，也可以在其他处理核上运行。在本申请中，对于第一任务来说，处理器 101 可以发起第一请求，用于请求配置第一任务的目标加速器，进而可以将第一任务从处理器 101 调度至目标加速器进行处理，提升了系统计算性能。

例如，如图 3 所示，当前在第一进程中运行任务 1（即将第一进程中当前运行的任务确定第一任务），其中，任务 1 中可以包括多个子任务，且第一个子任务在 CPU0 的 NUMA 节点 0 中的处理核 0 上运行。处理器 101（具体可以是 CPU0 的 NUMA 节点 0 中的处理核 0）可以发起第一请求，该第一请求用于请求配置任务 1 的目标加速器，进而可以将第一任务从处理器 101 调度至目标加速器进行处理，提升了系统计算性能。

接下来，将结合图 4 对主机上运行的软件架构进行描述，如图 4 所示，图 4 为本发明实施例提供的一种软件架构的示意图，图中应用系统负责执行应用服务（每个应用服务都可以运行在一个进程中，如第一进程），并且应用系统可以通过调用应用库子系统或驱动子系统实现加速器的调用。其中，应用库子系统可以包括但不限于开放式安全套接层协议（Open Secure Sockets Layer，OpenSSL）加速器引擎、加密/解密库、压缩/解压缩库等，可以向上层提供标准接口。此外，软件架构中还可以包括加速器用户态库，加速器用户态库是配合加速器内核态框架（User space Accelerator，UACCE）使用的，两者结合起来可以让注册到 UACCE 框架的加速器硬件以用户态软件的地址方式访问内存储器 102，使得处理器 101 和加速器两者之间可以建立基于先进先出（First Input First Output，FIFO）的通讯渠道。本申请中，在软件架构中增加了加速器调度引擎，加速器调度引擎可以用于接收加速器用户态库发起的第一请求，该第一请求用于请求配置当前第一进程中正在运行的任务的目标加速器，进而加速器调度引擎可以基于当前多个加速器的状态信息等，从多个加速器中选择一个作为目标加速器，并将任务调度至目标加速器进行处理，提升了系统计算性能。其中，加速器调度引擎中又可以包括任务管理器和加速器管理器，任务管理器可以用于响应第一请求并确定目标加速器等；加速器管理器可以用于获取多个加速器当前的状态信息等。

在一些实施例中，如图 5 所示，图 5 为本发明实施例提供的一种任务管理的示意图，加速器调度引擎中的任务管理器接收到每个请求时，都可以判断该请求是否为初始应用请求（即上述提及的第一请求），若是，则可以将请求对应的任务标记为“可配置”任务，然后可以按照接收顺序将任务添加到任务队列中，以便后续为“可配置”任务选择目标加速器；若否，表示之前已为任务配置了目标加速器，则可以将请求对应的任务标记为“已配置”任务，然后可以按照接收顺序将任务添加到任务队列中，以便后续基于任务对应的目标加速器信息将“已配置”任务调度至对应的加速器进行处理。

步骤 S202：主机响应于所述第一请求，基于所述第一任务中的第一个子任务对应的处理核，从 N 个加速器中确定目标加速器。N 为大于 0 的整数。

具体的，可以是主机上运行的软件架构中的加速器调度引擎（具体可以是加速器调度引擎中的任务管

理器）响应于第一请求；第一请求用于请求配置第一任务的目标加速器，第一任务中可以包括多个子任务，每个子任务可以对应一个处理核，且多个子任务中的第一个子任务的处理核是在任务执行前可以确定的，但其他子任务对应的处理核在任务执行前无法直接确定，而是在任务执行过程由操作系统根据多个处理核性能确定的。因此，可以基于第一子任务对应的处理核，从 N 个加速器中选择一个最优的作为第一任务的目标加速器。其中，N 个加速器可以为同类型的加速设备，也可以为多种不同类型的加速设备，可以根据实际情况进行选择。需要说明的是，N 个加速器中的每个加速器与处理器 101 中的部分处理核亲和度较高，即加速器在处理亲和度较高的处理核上运行的任务时，能够更加快速地访问对应的内存储器 102，从而能够提升系统计算效率。

例如，如图 6 所示，图 6 为本发明实施例提供的一种确定目标加速器的示意图，图中应用服务运行在第一进程中，且在 t0 时刻运行任务 1，在 t1 时刻运行任务 2。在 t0 时，加速器用户态库可以向任务管理器发送针对任务 1 的初始应用请求，任务管理器基于任务 1 中的第一个子任务对应的处理核（即 CPU0 的 NUMA 节点 0 中的处理核 0），可以将目标加速器确定为加速器 1。相对于加速器 2，加速器 1 与 CPU0 的 NUMA 节点 0 中的处理核 0 亲和度更高，即加速器 1 能够更加快速的访问到 NUMA 节点 0 对应的内存储器 102。在 t1 时，加速器用户态库可以向任务管理器发送针对任务 2 的初始应用请求，任务管理器基于任务 2 中的第一个子任务对应的处理核（即 CPU1 的 NUMA 节点 1 中的处理核 2），可以将目标加速器确定为加速器 2。相对于加速器 1，加速器 2 与 CPU1 的 NUMA 节点 1 中的处理核 2 亲和度更高，即加速器 2 能够更加快速的访问到 NUMA 节点 1 对应的内存储器 102。

而现有技术中，在应用服务初始化时，会为第一进程配置一个加速器，如加速器 1。之后在第一进程上运行每个关于该应用服务的加速任务都会被调度至加速器 1 进行处理，如在 t0 时刻，将任务 1 从 CPU0 的 NUMA 节点 0 中的处理核 0 调度至加速器 1 进行处理，在 t1 时刻，也会将任务 2 从 CPU1 的 NUMA 节点 1 中的处理核 2 调度至加速器 1 进行处理。但是，在此过程中，加速器 1 无法直接访问 CPU1 的 NUMA 节点 1 对应的内存储器 102，因此需要进行跨片访问，增加了访问时间，降低了加速器处理性能。

综上，由于在本申请中可以为应用服务中包括的多个任务中的每个任务都分配一个目标加速器，且在选择目标加速器时可以结合每个任务中的第一个子任务对应的处理核，因此不同任务在不同时间执行时，都能为其从多个加速器中选择一个最优的作为目标加速器。在运行整个应用服务过程中，有效降低应用接口复杂度，直接建连、使用，不需要了解系统拓扑信息和硬件情况，避免多加速器负载不均；动态调度，提升应用性能，避免在非绑核情况下的性能抖动，降低应用争抢加速器时性能影响。

在一种可能的实现方式中，所述主机基于所述第一任务中的第一个子任务对应的处理核，从 N 个加速器中确定目标加速器，包括：主机根据所述第一个子任务对应的处理核和所述 N 个加速器的实时状态信息，从所述 N 个加速器中确定所述目标加速器。

具体的，可以是主机上运行的软件架构中的加速器调度引擎（具体可以是加速器调度引擎中的任务管理器）为第一任务选择目标加速器。在确定目标加速器时，可以基于第一任务中的第一个子任务对应的处理核以及多个加速器的实时状态信息（即包括了每个加速器当前状态的信息），从 N 个加速器中确定与该处理核亲和度较高且当前性能最佳的为目标加速器，以确保目标加速器能够快速处理第一任务并且能够更加快速的访问到该处理核对应的内存储器 102，从而提升系统计算性能。需要说明的是，实时状态信息可以包括但不限于 N 个加速器中每个加速器的负载状态信息，内存访问速度信息，加速器能力值信息等。

在一种可能的实现方式中，所述方法还包括：主机获取所述实时状态信息，所述实时状态信息中包括所述 N 个加速器中每个所述加速器的负载状态信息，内存访问速度信息，加速器能力值信息中的一种或多种；其中，所述内存访问速度信息用于指示对应加速器访问不同内存储器时分别对应的访问速度，所述加速器能力值信息用于指示对应加速器包括的加速单元个数。

具体的，可以是主机上运行的软件架构中的加速器调度引擎（具体可以是加速器调度引擎中的加速器管理器）负责维护加速器设备实时状态，可以包括但不限于加速器设备在线/可用状态、负载状态信息、内存访问速度信息、加速器能力值信息等。如图 7 所示，图 7 为本发明实施例提供的一种加速器管理器的示意图，图中加速器管理器可以查看多个加速器中每个加速器设备的在线/可用状态，且可以动态维护加速器设备的在线、下线状态，支持动态上下线；设备上线时，加速器管理器可以获取设备加速器位置信息（即加速器拓扑状态）设置 NUMA 亲和度，以确定加速器的内存访问速度信息，该信息用于指示对应加速器访问不同内存储器时分别对应的访问速度；加速器管理器可以获取每个加速器中包括的加速单元个数，以确定加速器能力值信息，即加速单元个数越多，表示加速器计算效率越高；加速器管理器还可以动态获取加速器负载状况，并实时更新加速器负载值。

在一种可能的实现方式中，所述主机根据所述第一个子任务对应的处理核和所述 N 个加速器的实时状态信息，从所述 N 个加速器中确定所述目标加速器，包括：主机根据所述第一个子任务对应的处理核和所述 N 个加速器的所述实时状态信息，基于预设评分规则从所述 N 个加速器中确定评分最高的加速器为所述目标加速器。

具体的，可以是主机上运行的软件架构中的加速器调度引擎（具体可以是加速器调度引擎中的任务管理器）负责从多个加速器中选择目标加速器。首先，任务管理器可以先获取多个加速器的实时状态信息，然后查看第一任务中的第一个子任务对应的处理核信息，基于处理核信息确定处理核所在的NUMA 节点，以确定 NUMA 亲和度。进一步地，根据每个加速器对应的 NUMA 亲和度、加速器能力值、加速器负载值，基于预设评分规则对每个加速器进行综合评分，并将综合评分最高的确定为最优加速器。

例如，如图 8 所示，图 8 为本发明实施例提供的一种任务管理器确定目标加速器示意图，图中任务管理器，对于“可调度”任务，结合加速器管理器提供的加速器实时状态，可以确定目标加速器。进一步地，可以基于加速器用户态库将“可调度”任务调度至目标加速器进行处理，来降低处理器 101 使用率并提高系统性能。

步骤 S203：主机将所述第一任务调度至所述目标加速器进行处理。

具体的，当从多个加速器中选择出目标加速器后，可以将第一任务调度至目标加速器进行处理，来降低处理器 101 使用率并提升系统计算性能。

在一种可能的实现方式中，所述主机将所述第一任务调度至所述目标加速器进行处理，包括：主机判断所述目标加速器的所述负载状态信息对应的负载值是否大于预设值；若所述负载值小于或等于所述预设值，获取所述目标加速器的路径信息，并基于所述目标加速器的路径信息将所述第一任务调度至所述目标加速器进行处理。

具体的，可以是主机上运行的软件架构中的加速器调度引擎（具体可以是加速器调度引擎中的任务管理器）负责将任务调度至目标加速器进行处理。但在将任务调度至目标加速器前，可以先判断该目标加速器当前的负载值是否大于预设值，若目标加速器当前的负载值小于或等于预设值，表示目标加速器处于空闲状态或任务量较少的状态，目标加速器能够及时处理第一任务，进而任务管理器可以获取目标加速器的路径信息，并基于路径信息将任务调度至目标加速器进行处理，来降低处理器 101 使用率并提升系统计算性能。

可选的，加速器管理器可以针对每个加速器都生成一个加速器连接池，基于加速器连接池可以确定目标加速器的路径信息。加速器管理器可以动态维护加速器连接池，供任务调度器使用。其中，加速器设备上线时，可以建立连接池；加速器设备使用中，可以根据任务状况扩缩容连接池；加速器设备下线时，可以销毁连接池，释放资源。

在一种可能的实现方式中，所述方法还包括：若所述负载值大于所述预设值，则主机将所述第一任务调度至所述处理器进行处理。

具体的，在将任务调度至目标加速器前，可以先判断该目标加速器当前的负载值是否大于预设值，若目标加速器当前的负载值大于预设值，表示目标加速器处于繁忙状态，目标加速器无法及时处理第一任务，或者跨 NUMA、跨片代价过大，进而任务管理器可以将第一任务还是交给处理器进行处理，以确保系统的计算性能。

例如，如图 9 所示，图 9 为本发明实施例提供的一种联合多种设备调度任务的示意图，图中当任务管理器从任务队列中获取到“可配置”任务后，可以基于加速器状态信息从多个加速器中为“可配置”任务选择一个最优的作为目标加速器，如加速器 A。进一步地，在将任务调度至加速器 A 进行处理前，可以先判断该加速器 A 当前的负载值是否大于预设值，若加速器 A 当前的负载值小于或等于预设值，表示加速器 A 能够及时处理任务，进而任务管理器可以获取加速器 A 的路径信息，并基于路径信息将任务调度至加速器 A 进行处理；若加速器 A 当前的负载值大于预设值，表示加速器 A 处于繁忙状态，加速器 A 无法及时处理任务，或者跨 NUMA、跨片代价过大，进而任务管理器可以将任务还是交给处理器进行处理，以确保系统的计算性能。

在一种可能的实现方式中，所述方法还包括：主机将所述目标加速器的路径信息与所述第一个子任务对应的处理核，更新至所述第一任务对应的上下文信息。

具体的，当任务管理器确定了任务对应的目标加速器后，可以确定目标加速器的路径信息和处理核信息，并将目标加速器的路径信息和处理核信息更新至任务对应的上下文信息中，以便后续可以根据上下文信息中的目标加速器的路径信息，将任务中的子任务调度至目标加速器进行处理，以及根据上下文信息中

的处理核信息访问对应的内存储器 102，从而提升系统计算性能。

可选的，在为任务选择了目标加速器后，可以将任务标记为“已调度”任务（可以理解为任务中的所有子任务都可以标记为“已调度”任务），以便后续直接基于任务对应的上下文信息，将子任务调度至对应的加速器进行处理。

可选的，当任务管理器接收到初始应用请求后，任务管理器可以下发申请会话请求，任务管理器分配对应会话资源，用来存放任务对应的上下文信息（如加速器路径信息，CPU 核信息等）。

在一种可能的实现方式中，所述方法还包括：在执行所述多个子任务中的第 i 个子任务时，主机从所述上下文信息中，获取所述目标加速器的路径信息，并将所述第 i 个子任务调度至所述目标加速器进行处理，i 为大于 1 的整数。

具体的，任务管理器可以接收第二请求，第二请求为任务运行中加速器用户态库针对任务中的子任务（即“已调度”任务）下发的加速请求。任务管理器响应于第二请求，并获取任务对应的上下文信息，并基于上下文信息中的目标加速器的路径信息将任务调度至目标加速器进行处理，从而提升系统计算性能。

可选的，对于加速器用户态库提供的阶段化初始化请求（加密/解密初始化、压缩/解压缩初始化），即对于“已调度”任务来说，若任务管理器接收到阶段化初始化请求，则任务管理器可以将任务标记为“可调度”，并更新处理核信息，以及将加速器路径清空，以重新为任务选择加速器。

在本申请中，可以为应用服务中包括的多个任务中的每个任务都分配一个目标加速器，且在选择目标加速器时可以结合每个任务中的第一个子任务对应的处理核，因此不同任务在不同时间执行时，都能为其从多个加速器中选择一个最优的作为目标加速器。在运行整个应用服务过程中，有效降低应用接口复杂度，直接建连、使用，不需要了解系统拓扑信息和硬件情况，避免多加速器负载不均；动态调度，提升应用性能，避免在非绑核情况下的性能抖动，降低应用争抢加速器时性能影响；实时刷新加速器信息，支持动态上下线设备，应用无感知。

上述详细阐述了本发明实施例的方法，下面提供了本发明实施例的相关装置。

请参见图 10，图 10 是本发明实施例提供的本申请提供了一种加速器调度装置的结构示意图，该加速器调度装置 30，可以包括任务管理单元，加速器管理单元。其中，各个模块的详细描述如下。

所述任务管理单元 301，用于：接收第一请求，所述第一请求用于请求配置第一任务的目标加速器，所述第一任务为第一进程中当前运行的任务，所述第一进程运行在处理器中，所述处理器包括多个处理核，所述第一任务包括多个子任务，每个所述子任务与所述多个处理核中的一个对应；响应于所述第一请求，基于所述第一任务中的第一个子任务对应的处理核，从 N 个加速器中确定目标加速器，N 为大于 0 的整数；将所述第一任务调度至所述目标加速器进行处理。

在一种可能的实现方式中，所述任务管理单元 301，具体用于：根据所述第一个子任务对应的处理核和所述 N 个加速器的实时状态信息，从所述 N 个加速器中确定所述目标加速器；基于所述目标加速器的路径信息，将所述第一任务调度至所述目标加速器进行处理。

在一种可能的实现方式中，所述加速器调度装置还包括加速器管理单元 302，所述加速器管理单元，用于：向所述任务管理单元发送所述实时状态信息，所述实时状态信息中包括所述 N 个加速器中每个所述加速器的负载状态信息，内存访问速度信息，加速器能力值信息中的一种或多种；其中，所述内存访问速度信息用于指示对应加速器访问不同内存储器时分别对应的访问速度，所述加速器能力值信息用于指示对应加速器包括的加速单元个数。

在一种可能的实现方式中，所述任务管理单元 301，具体用于：根据所述第一个子任务对应的处理核和所述 N 个加速器的所述实时状态信息，基于预设评分规则从所述 N 个加速器中确定评分最高的加速器为所述目标加速器。

在一种可能的实现方式中，所述任务管理单元 301，具体用于：判断所述目标加速器的所述负载状态信息对应的负载值是否大于预设值；若所述负载值小于或等于所述预设值，获取所述目标加速器的路径信息，并基于所述目标加速器的路径信息将所述第一任务调度至所述目标加速器进行处理。

在一种可能的实现方式中，所述任务管理单元 301，还用于：若所述负载值大于所述预设值，则将所述第一任务调度至所述处理器进行处理。

在一种可能的实现方式中，所述任务管理单元 301，还用于：将所述目标加速器的路径信息与所述第一个子任务对应的处理核，更新至所述第一任务对应的上下文信息。

在一种可能的实现方式中，所述任务管理单元 ^01，还用于：在执行所述多个子任务中的第 i 个子任

务时，从所述上下文信息中，获取所述目标加速器的路径信息，并将所述第 i 个子任务调度至所述目标加速器进行处理，i 为大于 1 的整数。

需要说明的是，本发明实施例中所描述的加速器调度装置 30 中各功能单元的功能可参见上述图 2 中所述的方法实施例中加速器调度装置所执行的步骤 S201-步骤 S203 的相关描述，此处不再赘述。

本申请提供了一种计算机存储介质，其特征在于，所述计算机存储介质存储有计算机程序，该计算机程序被处理器执行时实现上述任意一种加速器调度方法。

本申请实施例提供一种电子设备，该电子设备中包括处理器，处理器被配置为支持该电子设备实现上述任意一种加速器调度方法中相应的功能。该电子设备还可以包括存储器，存储器用于与处理器耦合，其保存该电子设备必要的程序指令和数据。该电子设备还可以包括通信接口，用于该电子设备与其他设备或通信网络通信。

本申请提供了一种芯片系统，该芯片系统包括处理器，用于支持电子设备实现上述所涉及的功能，例如，生成或处理上述一种加速器调度方法中所涉及的信息。在一种可能的设计中，所述芯片系统还包括存储器，所述存储器，用于保存电子设备必要的程序指令和数据。该芯片系统，可以由芯片构成，也可以包含芯片和其他分立器件。

本申请提供一种计算机程序，其特征在于，所述计算机程序包括指令，当所述计算机程序被计算机执行时，使得所述计算机执行上述一种加速器调度方法。

在上述实施例中，对各个实施例的描述都各有侧重，某个实施例中没有详述的部分，可以参见其他实施例的相关描述。

需要说明的是，对于前述的各方法实施例，为了简单描述，故将其都表述为一系列的动作组合，但是本领域技术人员应该知悉，本申请并不受所描述的动作顺序的限制，因为依据本申请，某些步骤可能可以采用其他顺序或者同时进行。其次，本领域技术人员也应该知悉，说明书中所描述的实施例均属于优选实施例，所涉及的动作和模块并不一定是本申请所必须的。

在本申请所提供的几个实施例中，应该理解到，所揭露的装置，可通过其它的方式实现。例如，以上所描述的装置实施例仅仅是示意性的，例如上述单元的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式，例如多个单元或组件可以结合或者可以集成到另一个系统，或一些特征可以忽略，或不执行。另一点，所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些接口，装置或单元的间接耦合或通信连接，可以是电性或其它的形式。

上述作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

另外，在本申请各实施例中的各功能单元可以集成在一个处理单元中，也可以是各个单元单独物理存在，也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现，也可以采用软件功能单元的形式实现。

上述集成的单元如果以软件功能单元的形式实现并作为独立的产品销售或使用时，可以存储在一个计算机可读取存储介质中。基于这样的理解，本申请的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质中，包括若干指令用以使得一台计算机设备（可以为个人计算机、服务器或者网络设备等，具体可以是计算机设备中的处理器）执行本申请各个实施例上述方法的全部或部分步骤。其中，而前述的存储介质可包括：U 盘、移动硬盘、磁碟、光盘、只读存储器（Read-Only Memory，缩写：ROM）或者随机存取存储器（Random Access Memory，缩写：RAM）等各种可以存储程序代码的介质。

以上所述，以上实施例仅用以说明本申请的技术方案，而非对其限制；尽管参照前述实施例对本申请进行了详细的说明，本领域的普通技术人员应当理解：其依然可以对前述各实施例所记载的技术方案进行修改，或者对其中部分技术特征进行等同替换；而这些修改或者替换，并不使相应技术方案的本质脱离本申请各实施例技术方案的精神和范围。

权 利 要 求 书

1.一种加速器调度方法，其特征在于，所述方法包括：

接收第一请求，所述第一请求用于请求配置第一任务的目标加速器，所述第一任务为第一进程中当前运行的任务，所述第一进程运行在处理器中，所述处理器包括多个处理核，所述第一任务包括多个子任务，每个所述子任务与所述多个处理核中的一个对应；

响应于所述第一请求，基于所述第一任务中的第一个子任务对应的处理核，从N个加速器中确定目标加速器，N为大于0的整数；

将所述第一任务调度至所述目标加速器进行处理。

2.如权利要求1所述的方法，其特征在于，所述基于所述第一任务中的第一个子任务对应的处理核，从N个加速器中确定目标加速器，包括：

根据所述第一个子任务对应的处理核和所述N个加速器的实时状态信息，从所述N个加速器中确定所述目标加速器。

3.如权利要求2所述的方法，其特征在于，所述方法还包括：

获取所述实时状态信息，所述实时状态信息中包括所述N个加速器中每个所述加速器的负载状态信息，内存访问速度信息，加速器能力值信息中的一种或多种；

其中，所述内存访问速度信息用于指示对应加速器访问不同内存储器时分别对应的访问速度，所述加速器能力值信息用于指示对应加速器包括的加速单元个数。

4.如权利要求2或3所述的方法，其特征在于，所述根据所述第一个子任务对应的处理核和所述N个加速器的实时状态信息，从所述N个加速器中确定所述目标加速器，包括：

根据所述第一个子任务对应的处理核和所述N个加速器的所述实时状态信息，基于预设评分规则从所述N个加速器中确定评分最高的加速器为所述目标加速器。

5.如权利要求2-4任意一项所述的方法，其特征在于，所述将所述第一任务调度至所述目标加速器进行处理，包括：

判断所述目标加速器的所述负载状态信息对应的负载值是否大于预设值；

若所述负载值小于或等于所述预设值，获取所述目标加速器的路径信息，并基于所述目标加速器的路径信息将所述第一任务调度至所述目标加速器进行处理。

6.如权利要求5所述的方法，其特征在于，所述方法还包括：

若所述负载值大于所述预设值，则将所述第一任务调度至所述处理器进行处理。

7.如权利要求5或6所述的方法，其特征在于，所述方法还包括：

将所述目标加速器的路径信息与所述第一个子任务对应的处理核，更新至所述第一任务对应的上下文信息。

8.如权利要求7所述的方法，其特征在于，所述方法还包括：

在执行所述多个子任务中的第i个子任务时，从所述上下文信息中，获取所述目标加速器的路径信息，并将所述第i个子任务调度至所述目标加速器进行处理，i为大于1的整数。

9.一种加速器调度装置，其特征在于，所述加速器调度装置包括任务管理单元，所述任务管理单元，用于：

接收第一请求，所述第一请求用于请求配置第一任务的目标加速器，所述第一任务为第一进程中当前运行的任务，所述第一进程运行在处理器中，所述处理器包括多个处理核，所述第一任务包括多个子任务，每个所述子任务与所述多个处理核中的一个对应；

响应于所述第一请求，基于所述第一任务中的第一个子任务对应的处理核，从 N 个加速器中确定目标加速器，N 为大于 0 的整数；

将所述第一任务调度至所述目标加速器进行处理。

10.如权利要求 9 所述的装置，其特征在于，所述任务管理单元，具体用于：

根据所述第一个子任务对应的处理核和所述 N 个加速器的实时状态信息，从所述 N 个加速器中确定所述目标加速器。

11.如权利要求 10 所述的装置，其特征在于，所述加速器调度装置还包括加速器管理单元，所述加速器管理单元，用于：

向所述任务管理单元发送所述实时状态信息，所述实时状态信息中包括所述 N 个加速器中每个所述加速器的负载状态信息，内存访问速度信息，加速器能力值信息中的一种或多种；

其中，所述内存访问速度信息用于指示对应加速器访问不同内存储器时分别对应的访问速度，所述加速器能力值信息用于指示对应加速器包括的加速单元个数。

12.如权利要求 10 或 11 所述的装置，其特征在于，所述任务管理单元，具体用于：

根据所述第一个子任务对应的处理核和所述 N 个加速器的所述实时状态信息，基于预设评分规则从所述 N 个加速器中确定评分最高的加速器为所述目标加速器。

13.如权利要求 10-12 任意一项所述的装置，其特征在于，所述任务管理单元，具体用于：

判断所述目标加速器的所述负载状态信息对应的负载值是否大于预设值；

若所述负载值小于或等于所述预设值，获取所述目标加速器的路径信息，并基于所述目标加速器的路径信息将所述第一任务调度至所述目标加速器进行处理。

14.如权利要求 13 所述的装置，其特征在于，所述任务管理单元，还用于：

若所述负载值大于所述预设值，则将所述第一任务调度至所述处理器进行处理。

15.如权利要求 13 或 14 所述的装置，其特征在于，所述任务管理单元，还用于：

将所述目标加速器的路径信息与所述第一个子任务对应的处理核，更新至所述第一任务对应的上下文信息。

16.如权利要求 15 所述的装置，其特征在于，所述任务管理单元，还用于：

在执行所述多个子任务中的第 i 个子任务时，从所述上下文信息中，获取所述目标加速器的路径信息，并将所述第 i 个子任务调度至所述目标加速器进行处理，i 为大于 1 的整数。

17.一种计算机存储介质，其特征在于，所述计算机存储介质存储有计算机程序，该计算机程序被处理器执行时实现上述权利要求 1-8 中任意一项所述的方法。