

一种调度方法、装置、系统和计算设备

本申请要求于 2021 年 5 月 21 日提交中国专利局、申请号为 202110558600.X、发明名称为“一种数据处理的方法、装置和计算机设备”的中国专利申请的优先权，以及于 2021 年 7 月 08 日提交的申请号为 202110770860.3、发明名称为“一种调度方法、装置、系统和计算设备”的中国专利申请的优先权，前述两件专利申请的全部内容通过引用结合在本申请中。

技术领域

本申请涉及计算机领域，尤其涉及一种调度方法、装置、系统和计算设备。

背景技术

传统服务器的处理器为对称多处理体系结构（symmetric multi-processing, SMP）或者非一致性内存架构(non-uniform memory architecture, NUMA)架构，即服务器上的多个处理器的指令集架构和微架构完全相同，操作系统通过对特定任务进行性能采样，判断任务所需要的合适算力，然后将任务调度到合适的处理器核上运行。在通用计算处理器内集成浮点加速器并通过特定的加速指令驱动是多样性计算时代通用处理器体系结构发展的趋势，这种类型的处理器跟一般的通用处理器相比，由于支持高性能浮点加速，微架构和指令集架构都略优不同。当通用处理器和带浮点加速的处理器在一台服务器内通过互联总线组成一个同构多指令集体系结构的计算系统时，传统的调度器将带浮点加速指令的任务调度到通用处理器核上时会产生无法识别的指令异常，导致任务非正常退出。现有的处理方法为通过编程框架，用户调用编程框架提供的接口，将应用发送至用户指定的处理器执行，需要用户对处理器支持的指令集类型有一定程度的了解，并且编程复杂。因此如何提供一种支持在同构多指令集体系结构下实现高性能的任务调度方法成为亟待解决的问题。

发明内容

本申请提供了一种调度方法、装置、系统和计算设备，能支持在同构多指令集体系结构下实现高性能计算的数据处理。

第一方面，提供一种调度方法，该方法由计算设备执行，该计算设备包括多个处理器，多个处理器中至少有一个处理器支持的指令集集合中至少有一个指令集的类型与其他处理器支持的指令集集合中的指令集的类型不同。该调度方法包括：首先获取应用的指令集类型，并从多个处理器中选择目标处理器，其中，应用的指令集的类型是目标处理器支持的指令集集合中的多个指令集的类型子集。然后将应用分配至目标处理器执行。通过上述方法，新的调度方法识别应用的指令集类型和处理器支持的指令集集合后再调度，让操作系统具备了调度不同指令集集合的处理器混搭系统的能力。

在一种可能的实现方式中，从多个处理器中选择目标处理器的方法，具体可以根据应用的指令集的类型从多个处理器中选择目标处理器，其中应用的指令集的类型是目标处理器支持的指令集集合中的多个指令集的类型子集。通过上述方法，可以保证将应用调度给支持应用的指令集的类型处理器，从而避免应用执行异常。

在另一种可能的实现方式中，目标处理器属于目标调度域，其中，目标调度域支持的指令集集合与目标处理器支持的指令集集合相同，在选择目标处理器之前，从多个调度域中选择目标调度域，其中，应用的指令集的类型是目标调度域支持的指令集集合中的多个指令集

的类型的子集。通过上述方法，可以利用调度域选择目标处理器，提高调度的效率。

在另一种可能的实现方式中，目标调度域仅包括目标处理器，或者包括目标处理器以及其他处理器，其中，其他处理器支持的指令集集合与目标处理器支持的指令集集合相同。通过上述方法可以将支持的指令集集合一样的处理器作为一个调度域统一管理，提高调度的效率。

在另一种可能的实现方式中，应用包括至少一个任务，调度方法还包括将任务的指令集的类型写入任务的任務控制块，其中，任务的指令集的类型与应用的指令集的类型相同。则获取应用的指令集的类型具体包括从任务控制块中获取任务的指令集的类型。通过上述方法，可以将应用分成多个任务，将每一个任务分配到不同的处理器上执行，提高应用的执行效率。

在另一种可能的实现方式中，将应用分配至所述目标处理器执行，具体可以将任务分配至目标处理器执行。

在另一种可能的实现方式中，目标处理器包括至少一个处理单元，则将任务分配至目标处理器执行，具体包括：当目标处理器支持多线程并行处理时，将任务分配至目标处理器的一个处理单元执行。通过上述方法，可以使用处理器的多线程并行处理技术，在一个处理器上同时执行多个任务，提高应用的执行效率。

在另一种可能的实现方式中，目标处理器包括至少一个处理核，则将任务分配至目标处理器执行，具体包括：当目标处理器不支持多线程并行处理时，将任务分配至目标处理器的一个处理核执行。通过上述方法，可以使用处理器的多核技术，在一个处理器上同时执行多个任务，提高应用的执行效率。

在另一种可能的实现方式中，当处理器执行所述任务失败时，将目标调度域的标识存入任务的任務控制块的第一存储单元中，其中，第一存储单元中存储的标识用于指示目标调度域所包括的处理器不能被分配用于执行所述应用。通过上述方法，可以避免同一个应用的任务再次被调度到不支持执行的处理器上，增加应用成功执行的概率。

在另一种可能的实现方式中，当处理器执行所述任务成功时，将目标调度域的标识存入任务的任務控制块的第二存储单元中，其中，第二存储单元中存储的标识用于指示目标调度域所包括的处理器能被分配用于执行所述应用。通过上述方法，可以减少调度器判断调度域支持任务执行的时间，提高应用执行的效率。

在另一种可能的实现方式中，获取待执行应用的指令集类型，具体包括：当应用可编译时，通过编译选项获取应用的指令集类型；或者，当应用不可编译时，通过异常指令获取应用的指令集类型。可以通过两种方式获取应用的指令集的类型，本申请提出的调度方法不仅可以调度待执行的新应用还可以调度已经存在可执行文件的旧应用。

在另一种可能的实现方式中，通过编译选项获取所述应用的指令集类型，具体包括：通过编译选项直接获取用户指定的执行应用的指令集的类型；或者先通过编译选项获取用户指定的执行应用的处理器类型，再根据处理器类型与指令集类型的关系确定应用的指令集的类型。

在另一种可能的实现方式中，通过编译选项获取应用的指令集类型后，还可以将应用编译为二进制文件，其中，二进制文件包括应用的指令集的类型。

在另一种可能的实现方式中，通过异常指令获取应用的指令集类型后，还可以解析异常指令的二进制代码，得到异常指令所属的指令集类型，再将应用的指令集类型设置为异常指令集的类型。

在另一种可能的实现方式中，任务为设备执行应用生成的进程。

在另一种可能的实现方式中，任务为设备执行应用生成的线程。

第二方面，本申请提供一种计算设备，包括多个处理器，其中，多个处理器中至少一个处理器支持的指令集集合中至少有一个指令集的类型与其他处理器支持的指令集集合中的指令集的类型不同。多个处理器中的主处理器用于实现上述第一方面或第一方面中任意一种可能的实现方式中的调度方法的操作步骤的功能。

第三方面，提供一种调度装置，所述调度装置包括用于执行第一方面或第一方面任一种可能实现方式中的调度方法的各个模块。所述调度装置部署于计算设备，其中，计算设备包括多个处理器，多个处理器中至少一个处理器支持的指令集集合中至少有一个指令集的类型与其他处理器支持的指令集集合中的指令集的类型不同。

第四方面，本申请提供一种调度系统，包括用于执行第一方面或第一方面任一种可能实现方式中的调度方法的调度装置和多个处理器，其中，多个处理器中至少一个处理器支持的指令集集合中至少有一个指令集的类型与其他处理器支持的指令集集合中的指令集的类型不同。其中，调度装置可以与多个处理器位于不同的计算设备；多个处理器中至少一个处理器也可以与其他的处理器位于不同的计算设备上。

第五方面，提供一种计算机可读存储介质，所述计算机可读存储介质中存储有指令，当其在计算机上运行时，使得计算机执行上述第一方面或第一方面任一种可能实现方式中所述方法的操作步骤。

第六方面，提供了一种包含指令的计算机程序产品，当其在计算机上运行时，使得计算机执行上述第一方面或第一方面任意一种可能的实现方式中所述方法的操作步骤。

本申请在上述各方面提供的实现方式的基础上，还可以进行进一步组合以提供更多实现方式。

附图说明

图1是本申请实施例提供的一种设备100的结构示意图；

图2是本申请提供的一种操作系统13的示意图；

图3是本申请提供的一种调度域的示意图；

图4是本申请提供的一种两级的调度域在内存中存储的结构体的示意图；

图5是本申请实施例提供的一种调度方法的流程示意图；

图6是本申请实施例提供的一种任务控制块的示意图；

图7是本申请实施例提供的另一种调度方法的流程示意图；

图8是本申请提供的一种设置“黑名单”的方法；

图9是本申请提供的一种设置“白名单”的方法；

图10是本申请提供的另一种任务控制块的示意图；

图11是本申请实施例提供的一种调度装置1100的示意图。

具体实施方式

下面结合附图对本申请实施例中的技术方案进行描述。图1是本申请实施例提供的一种设备100的结构示意图，设备100可以是任意具有至少两个处理器的计算设备（例如，服务器）。具有多处理器的设备可以采用对称多处理体系结构（symmetric multi-processing, SMP）或者非统一内存访问体系结构（None Uniform Memory Access, NUMA）。为了便于描述，本申请以下实施例以设备100采用NUMA结构为例进行描述，需要说明的是，本申请提供的一种任务调度的方法也适用于采用SMP结构的设备，以及具有其他多处理器结构的设备。

设备 100 包括处理节点 14 和处理节点 15, 其中, 多个处理节点和内存 16 通过总线连接, 总线可以是数据总线, 还可以是电源总线、控制总线和状态信号总线等, 该总线也可以为其他类型实现计算设备内器件间连接的总线, 例如快速通道互联 (intel quickpath interconnect, QPI)。需要说明的是, 设备 100 中处理节点的数量并不构成对本申请的限定, 图 1 仅以设备 100 包括两个处理节点为例进行说明。

设备 100 中每个处理节点都具有如图 1 所示的单个处理节点的结构, 下面以一个处理节点 (处理节点 14) 为例说明每个处理节点内部的组件以及连接方式。处理节点 14 包括处理器 140、内存 144 以及内存管理器 143。其中, 处理器 140、内存 144 和内存管理器 143 之间通过片内总线互联。

处理器 140 是物理的中央处理单元 (central processing unit, CPU) 的芯片。一个处理节点上的处理器可以通过片内总线快速访问该节点的本地内存, 也可以通过高速互连网络访问其他节点的内存。例如, 处理节点 14 的处理器 141 可以访问处理节点 14 的内存 144, 也可以访问处理节点 15 的内存 154。相比于处理器访问本处理节点的内存, 处理器访问其他处理节点的内存将花费更多时间。因此对于 NUMA 系统, 与处理器在同一个处理节点内的内存的理想状况下应包含与该处理器最相关联的信息。在一些实施例中, 处理器 140 又可以包括多个核 (core), 例如处理器核 141 和处理器核 142, 这些核都被集成在处理器 140 的芯片上, 每个处理器核可以有独立的 L1 缓存, 同一个处理器中的不同的处理器核可以共享 L2 缓存。在一些实施例中, 每个处理器核中又可以通过同时多线程 (simultaneous multithreading) 等技术使得一个处理器核可以并行执行多个线程或进程。处理每一个线程或进程的硬件单元, 从逻辑上看就是一个 CPU, 又可以称为处理单元, 例如处理单元 1411 和处理单元 1412。同一个处理器的处理单元之间共享所有处理器的资源, 包括 L1 缓存以及逻辑运算单元。

内存 144 或内存 154 是指与处理单元直接交换数据的内部存储器, 它可以随时读写数据, 而且速度很快, 作为操作系统或其他正在运行中的程序的临时数据存储器。内存 144 或内存 154 包括至少两种存储器, 例如既可以是随机存取存储器, 也可以是只读存储器 (Read Only Memory, ROM)。举例来说, 随机存取存储器是动态随机存取存储器 (Dynamic Random Access Memory, DRAM), 或者存储级存储器 (Storage Class Memory, SCM)。DRAM 是一种半导体存储器, 与大部分随机存取存储器 (Random Access Memory, RAM) 一样, 属于一种易失性存储器 (volatile memory) 设备。SCM 是一种同时结合传统储存装置与存储器特性的复合型储存技术, 存储级存储器能够提供比硬盘更快速的读写速度, 但存取速度上比 DRAM 慢, 在成本上也比 DRAM 更为便宜。然而, DRAM 和 SCM 在本实施例中只是示例性的说明, 内存还可以包括其他随机存取存储器, 例如静态随机存取存储器 (Static Random Access Memory, SRAM) 等。而对于只读存储器, 举例来说, 可以是可编程只读存储器 (Programmable Read Only Memory, PROM)、可抹除可编程只读存储器 (Erasable Programmable Read Only Memory, EPROM) 等。另外, 内存 144 或内存 154 还可以是双列直插式存储器模块或双线存储器模块 (Dual In-line Memory Module, 简称 DIMM), 即由动态随机存取存储器 (DRAM) 组成的模块, 还可以是固态硬盘 (Solid State Disk, SSD)。实际应用中, NUMA 系统中可配置多个内存, 以及不同类型的内存。本实施例不对内存的数量和类型进行限定。此外, 可对内存进行配置使其具有保电功能。保电功能是指系统发生掉电又重新上电时, 内存中存储的数据也不会丢失。具有保电功能的内存被称为非易失性存储器。

内存管理器用于管理与规划从内存到处理单元之间的数据传输, 可以是一个单独的芯片, 或集成在处理器的芯片中。

设备 100 支持不同指令集版本的处理器混搭组成一个同构多指令集体系结构，可以包括以下两种场景。

场景一：通用处理器和通用指令异构处理器混搭。

通用指令异构处理器是指在通用处理器内部集成包括浮点加速器在内的加速器，并增加特定的加速指令用来驱动加速器，使得加速器能够重用处理器上的各种资源，支持不同精度的浮点数计算的需要。示例性地，图 1 中处理器 140 可以为通用处理器，处理器 150 可以为集成浮点加速器的通用处理器。

场景二：不同时间生产的处理器的混搭。

随着芯片制造工艺的进步，不同时间生产的处理器的硬件电路的复杂程度不一样，因此，支持的指令集版本也略有区别。示例性地，图 1 中处理器 140 可以是通用处理器第 9 版本，处理器 150 可以是通用处理器第 8.6 版本。

在设备 100 中，操作系统 13 用于运行应用，并将应用生成的任务转化为指令分配给能够执行成功执行任务的处理器执行。在一种实施方式中，操作系统 13 可以在设备 100 的主处理器上运行，其余处理器收到运行应用请求后，将该请求发送给主处理器，由操作系统 13 将应用生成的任务转化为指令分配给能够执行成功执行任务的处理器执行。其中，主处理器可以由用户在设备 100 的多个处理器中预先指定，本申请对指定的方式不做限定，例如，主处理器可以为处理器 140 或者处理器 150。在另一种实施方式中，操作系统 13 还可以在设备 100 的多个处理器上运行，例如，操作系统 13 同时运行在处理器 140 和处理器 150 上。当一个处理器收到运行应用请求后，由该处理器上的操作系统 13 将应用生成的任务转化为指令分配给能够执行成功执行任务的处理器执行。此时，可以将执行操作系统 13 的处理器称为主处理器。

图 2 是本申请提供的一种操作系统 13 的示意图，以操作系统存储在内存 144 中为例，如图 2 所示，操作系统 13 包括编译器 131、装载器 132 和调度器 133。其中，编译器 131 用于将应用的代码编译为处理器可以执行的二进制文件。当二进制文件执行时，装载器 132 创建对应的进程或者线程用于加载该二进制文件的指令。一方面，每一个二进制文件加载一次可以创建一个新的进程，当一个进程被创建时，装载器 132 还会为该进程分配一段内存空间用于保存与进程有关的上下文，包括：进程标识符、进程的调度信息、进程的控制信息以及下一条待执行的指令的地址等，这段内存空间又可以称为进程控制块。另一方面，装载器 132 还可以为一个进程创建多个线程，用于执行进程中的多个子任务，多个线程可以共享进程的软件和硬件资源。当一个线程被创建时，装载器 132 会在进程控制块所占的空间中为该线程分配一段内存空间用于保存与线程有关的上下文，包括：线程标识符、线程的调度信息、线程的控制信息以及下一条待执行的指令的地址等，分配给线程的内存空间又可以称为线程控制块。最后，调度器 133 将应用的进程或线程调度至不同的处理单元执行。为了便与描述，以下实施例中将进程或者线程统称为任务，并且将进程控制块或者线程控制块统称为任务控制块。

操作系统 13 还可以包括解析器 134，可以用于解析指令的二进制代码得到该指令的指令集类型。

调度器 133 对任务进行调度时，可以采用多级别的调度域的管理方法，每个调度域包括一个或多个硬件。例如，参见图 3，硬件可以为一个处理器，则可以将包括一个或多个处理器的调度域称为物理域 (PHY_domain)；硬件还可以为一个处理器核，则可以将包括一个或多个处理器核的调度域称为核心域 (MC_domain)；硬件还可以为一个处理器单元，则可以将

包括一个或多个处理单元的调度域称为超线程域 (SMT_domain)。调度域表示调度器 133 执行调度策略的范围, 调度器 133 可以在每个调度域中执行不同的调度策略。每个调度域中的硬件又可以分为一个或多个调度组 (CPU group), 每个调度组可以看作是一个独立的执行单元, 调度器在调度域的不同调度组之间执行调度策略。例如, 参见图 3, 当调度域为物理域时, 可以将每一个处理器作为一个调度组; 当调度域为核心域时, 可以将每一个处理器核作为一个调度组; 当调度域为超线程域时, 可以将每一个处理单元作为一个组。当操作系统 13 启动时, 调度器 133 可以建立不同级别的调度域, 并且在每层调度域上建立相应的调度组。

示例性地, 以图 1 中设备 100 的处理器结构为例, 图 3 是本申请提供一种调度域的示意图, 如图 3 所示, 调度域可以分为超线程域、核心域以及物理域。具体地, 超线程域是最底层的调度域, 每个处理器核可以看作是一个超线程域, 例如处理器核 141 为超线程域 1。每个超线程域关联两个处理单元, 例如超线程域 1 关联处理单元 1411 和处理单元 1412。每个处理单元作为超线程域中的一个调度组, 例如, 调度组 0 为处理单元 1411。更高一级的调度域为核心域, 每个处理器可以看作是一个核心域, 例如处理器 140 为核心域 1。每个核心域关联两个处理器核, 例如, 核心域 1 关联处理器核 141 和处理器核 142。每个处理器核作为核心域中的一个调度组, 例如, 调度组 00 为处理器核 141。更高一级的物理域由系统中多个处理器组成, 每个物理域关联多个处理器, 例如物理域 1 关联处理器 140 和处理器 1250。在物理域中, 每一个处理器作为一个调度组, 例如, 调度组 000 为处理器 140。

可选地, 当多核处理器不支持多线程并行处理时, 调度域可以分为物理域以及核心域两个级别。

可选地, 当处理器只存在一个核且支持多线程并行处理时, 调度域可以分为物理域以及超线程域两个级别。

可选地, 当处理器只存在一个核且不支持多线程并行处理时, 调度域即为物理域。

调度器 133 可以将调度域的拓扑关系图以结构体的形式保存在内存中, 图 4 是本申请提供的一种两级的调度域在内存中存储的结构体的示意图, 如图 4 所示, 调度域呈现树状的拓扑结构, 不同级别的调度域之间以指针连接在一起, 低级别的调度域可以看作是高级别的调度域中的一个调度组。例如调度域 2 和调度域 3 分别是调度域 1 的调度组 1 和调度组 2。调度器可以首先在调度域 1 中的调度组 1 和调度组 2 之间执行调度域 1 的调度策略。选中一个调度组, 例如调度组 1 后, 调度器将任务分配给调度域 2, 接着在调度域 2 的调度组 3-5 之间执行调度域 2 的调度策略。选中一个调度组, 例如调度组 3 后, 调度器将任务分配给调度组 3。调度域的结构体中包括调度域的编号 (也可以称作调度域标识, 当结构体仅是一个调度组时, 为 NULL)、调度组的编号 (也可以称作调度组标识, 当该结构体不是其他调度域的调度组时, 为 NULL)、调度策略、调度域包括的硬件的编号、调度组的指针 (当结构体仅是一个调度组时, 为 NULL)。调度策略可以由用户设定, 用户可以为不同的级别的调度域设置不同的调度策略, 设置的方法本申请并不做限定。

当一个设备中多个调度组成同构多指令集体系结构的计算系统时, 通过本申请提供的任务调度的方法, 调度器 133 可以识别应用的任务中指令集的不同, 将任务调度到可以执行该指令集的处理单元中, 避免处理单元产生无法识别的指令异常而导致任务非正常退出。

下面将结合图 5 至图 10 对本申请实施例提供的数据处理的方法进行说明, 图 5 是本申请实施例提供的一种调度方法的流程示意图, 该方法可以使用图 2 所示的操作系统 13 执行, 具体地, 该方法可以使用运行操作系统 13 的主处理器执行, 该主处理器可以为用户指定的处理器也可以为运行操作系统 13 的处理器, 例如, 图 1 中的处理器 140 或者处理器 150。如图 5

所示，具体方法包括：

S501、编译器 131 获取应用的指令集的类型。

应用在被执行之前，需要先将代码编译为二进制可执行文件，再由调度器调度给处理器执行。编译器 131 在编译应用的代码时，可以采用以下两种方式通过编译选项获取应用的指令集的类型。

方法一：用户在编译选项中指定执行应用的处理器型号。

用户可以通过编译选项指定执行应用的处理器型号，例如：

`-march=cpu-type;`

其中，等号右边表示执行应用的处理器型号，则应用可以编译为在该类型的处理器上执行的二进制文件。

编译器 131 获取到处理器型号之后可以根据处理器型号确定该处理器支持的指令集集合，指令集集合中包括了处理器支持的所有指令集的类型。处理器型号与该处理器支持的指令集集合之间的关系可以通过表格的形式预置存储在设备中，示例性地，表 1 是本申请实施例提供的一种处理器型号与指令集集合关系的列表，表中记录了每一个处理器的型号（也可以称作标识）以及该型号的处理器支持的指令集集合。

表 1、处理器型号与指令集集合

型号	指令集集合
型号一	MMX, SSE, SSE2, SSE3, SSSE3
型号二	MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, POPCNT
型号三	MOVBE, MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, POPCNT, AVX, AVX2, AES, PCLMUL, FSGSBASE, RDRND, FMA, BMI, BMI2, F16C, RDSEED, ADCX, PREFETCHW
型号四	MOVBE, MMX, SSE, SSE2, SSE3, SSSE3, SSE4.1, SSE4.2, POPCNT, AVX, AVX2, AES, PCLMUL, FSGSBASE, RDRND, FMA, BMI, BMI2, F16C, RDSEED, ADCX, PREFETCHW, CLFLUSHOPT, XSAVEC, XSAVES

方式二：用户在编译选项中直接指定执行应用的指令集的类型。

用户可以通过编译选项“`-m-xxx`”直接指定执行应用的指令集的类型，表示应用可以编译为在该指令集类型的处理器上执行的二进制文件，例如：

`-mmmx -msse;`

表示应用编译的指令集的类型为 MMX 和 SSE。

S502、编译器 131 将指令集的类型写入应用的可执行二进制文件中。

编译过程中，编译器 131 将指令集的类型写入编译生成的可执行二进制文件中。示例性地，编译器 131 可以将指令集的类型写入可执行二进制文件的.comment 字段。

S503、运行可执行二进制文件，装载器 132 将指令集的类型放入任务控制块中。

当二进制文件执行时，装载器 132 根据二进制文件的指令，创建对应的任务，并在内存中创建对应的任务控制块用于保存与进程有关的上下文。装载器 132 将指令集的类型也存储在二进制文件对应的所有任务的任务控制块中。图 6 是本申请提供的一种任务控制块的示意图。如图 6 所示，应用运行生成多个任务，每个任务对应一个任务控制块，在每一个任务控制块中都开辟一块空间用于保存指令集的类型。

S504、调度器 133 获取处理器的指令集集合。

在操作系统启动时，调度器 133 可以获取设备上部署的处理器的支持的指令集集合，例

如，调度器 133 可以通过调用 `cpuinfo` 指令，读取 `modelname` 字段，获取每个处理器的型号，然后再根据处理器型号与指令集集合关系的列表得到每个处理器支持的指令集的类型。又例如，调度器 133 还可以通过调用 `cpuinfo` 指令，读取 `flag` 字段，直接得到每个处理器支持的指令集的类型。

S505、调度器 133 根据处理器的指令集集合设置调度域。

调度器 133 根据处理器的指令集集合，将指令集集合相同的处理器设置为一个物理域，并将指令集集合中所有的指令集的类型以指令集集合的形式写入该物理域的结构体中。

S506、调度器 133 将任务调度至可以执行该任务的处理单元。

当时钟中断触发一次物理域的任务调度时，调度器 133 判断物理域的指令集集合是否包括需要调度的任务的任务控制块中的所有指令集的类型。在可以执行该任务的物理域中根据物理域的调度策略选择一个调度组，即一个处理器。之后，调度器 133 可以根据核心域的调度策略从核心域中选择一个调度组，即一个处理器核，再根据超线程域的调度策略从超线程域中选择一个调度组，即一个处理单元执行该任务。

可选地，当多核处理器不支持多线程并行处理时，调度器 133 可以根据核心域的调度策略从核心域中选择一个调度组，即一个处理核执行该任务。

可选地，当处理器只存在一个核且支持多线程并行处理时，调度器 133 可以直接根据超线程域的调度策略从超线程域中选择一个调度组，即一个处理单元执行该任务。

可选地，可选地，当处理器只存在一个核且不支持多线程并行处理时，调度器 133 直接选择该调度组，即处理器执行该任务。

可选地，当所有的物理域都不能执行该任务时，调度器 133 直接报错，并停止执行该应用。

本申请提出的任务调度的方法，可以应用于同构多指令集的结构中，通过编译器和调度器的配合，支持不同指令集版本的调度组成的多处理器系统在统一的操作系统中灵活调度。

对于大量现存的二进制可执行文件和动态链接库，由于无法通过编译器重新编译，因此并不能将指令集类型写入应用的可执行二进制文件中。可以采用以下方法进行任务调度，图 7 是本申请提供的另一种调度方法，该方法与图 5 类似，可以使用图 2 所示的操作系统 13 执行，具体地，可以使用运行操作系统 13 的主处理器执行，该主处理器可以为用户指定的处理器也可以为运行操作系统 13 的处理器，例如，图 1 中的处理器 140 或者处理器 150。如图 7 所示，具体方法包括：

S701、运行应用的可执行二进制文件，判断任务中指令的执行是否出现异常。

S702、当指令出现异常时，解析器 134 读取异常指令的二进制代码，并解析该指令的指令集的类型。

S703、类似 S403，装载器 132 将指令集的类型放入任务控制块中。

S704、调度器 133 将该异常指令对应的任务放入等待队列。

S705、调度器 133 将任务调度至可以执行该任务的处理单元。

具体地，类似 S504 至 S506，调度器 133 设置调度域的指令集集合，并将任务调度至可以执行的处理单元。

通过上述方法，可以通过动态识别异常指令识别应用的指令集类型，解决老应用程序或者动态链接库的任务被调度到不合适的处理器上造成重复错误的问题。

为了减少调度器判断指令集类型是否相同的时间以及提高调度器执行调度的准确性，还可以在任务的任务控制块中添加“黑名单”或者“白名单”，其中，“黑名单”用于存储执

行任务时出现异常的处理器所关联的调度域。“白名单”用于存储执行任务成功的处理器所关联的调度域。图 8 是本申请提供的一种设置“黑名单”的方法，可以使用图 2 所示的操作系统 13 执行，具体地，可以使用运行操作系统 13 的主处理器执行，该主处理器可以为用户指定的处理器也可以为运行操作系统 13 的处理器，例如，图 1 中的处理器 140 或者处理器 150。如图 8 所示，该方法可以在图 5 所示的方法中的步骤 S506 之后执行，也可以在图 7 所示的方法中的步骤 S702 之后执行，方法具体如下：

S801、运行应用的可执行二进制文件，判断任务中指令的执行是否出现异常。

S802、当指令出现异常时，调度器 133 获取出现该异常指令的处理器所在的物理域的编号。

S803、装载器 132 在任务控制块中设置“黑名单”。

在任务控制块中分配一块存储区域用于存放调度域的“黑名单”，为了便于表述，该存储区域又可以称为第一存储单元，装载器 132 将物理域的编号存入该应用的所有任务的任务控制块的第一存储单元中。

添加“黑名单”后，调度器 133 在下次调度时，不会将任务调度给“黑名单”中存储的编号对应的调度域。可选地，当任务的“黑名单”中包括了设备 100 上所有的物理域时，表示所有的处理器都不能支持执行该任务的指令，调度器 133 可以直接报错，并停止执行该应用。

通过上述方法，可以避免同一个应用的任务再次被调度到不支持执行的处理器上，增加应用成功执行的概率。

图 9 是本申请提供的一种“白名单”的方法，可以使用图 2 所示的操作系统 13 执行，具体地，可以使用运行操作系统 13 的主处理器执行，该主处理器可以为用户指定的处理器也可以为运行操作系统 13 的处理器，例如，图 1 中的处理器 140 或者处理器 150。如图 8 所示，该方法可以在图 5 所示的方法中的步骤 S506 之后执行，方法具体如下：

S901、运行应用的可执行二进制文件，判断任务中指令的执行是否出现异常。

S902、当指令未出现异常时，调度器 133 获取出现该处理器所在的物理域的编号。

S903、装载器 132 在任务控制块中设置“白名单”。

在任务控制块中分配一块存储区域用于存放调度域的“白名单”，为了便于表述，该存储区域又可以称为第二存储单元，装载器 132 将物理域的编号存入该应用的所有任务的任务控制块的第二存储单元中。

添加“白名单”后，调度器 133 在下次调度时，可以直接任务调度给“白名单”中存储的编号对应的调度域。

通过上述方法，可以减少调度器判断调度域是否支持执行任务所需的指令集的时间，提高应用执行的效率。

需要说明的是，可以在任务的任务控制块中添加“黑名单”以及“白名单”中的任意一种，也可以在任务的任务控制块中添加“黑名单”以及“白名单”两种。图 10 是本申请提供的一种添加了“黑名单”以及“白名单”的任务控制块的示意图。

值得说明的是，对于上述方法实施例，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，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，所涉及的动作并不一定是本申请所必须的。

本领域的技术人员根据以上描述的内容，能够想到的其他合理的步骤组合，也属于本中

请的保护范围内。其次，本领域技术人员也应该熟悉，说明书中所描述的实施例均属于优选实施例，所涉及的动作并不一定是本申请所必须的。

上文中结合图 5 至图 10，详细描述了根据本申请实施例所提供的任务调度的方法，下面将结合图 11，描述根据本申请实施例所提供的一种任务调度的装置。

图 11 为本申请实施例提供的一种调度装置 1100 的示意图，包括获取单元 1101 和调度单元 1102。调度装置 1100 用于图 2 所示的操作系统 13 的功能。

其中，获取单元 1101，用于获取应用的指令集的类型，具体实现可参考图 5 中 S501 的步骤，这里不再赘述。在产品的实现过程中，获取单元 1101 可以是图 2 所示的编译器 131，也可以是其他编译器，本实施例不对获取单元 1101 的形态进行限定。

调度单元 1102，用于从计算设备的多个处理器中选择目标处理器，其中，多个处理器中至少一个处理器支持的指令集集合中至少有一个指令集的类型与其他处理器支持的指令集集合中的指令集的类型不同，应用的指令集的类型是目标处理器支持的指令集集合中的多个指令集的子集；调度单元 1102 还用于将应用分配至目标处理器执行。调度单元 1102 具体实现可参考图 5 中 S504 至 S506 的步骤，这里不再赘述。在产品的实现过程中，获取单元 1101 可以是图 2 所示的调度器 133，也可以是其他调度器，本实施例不对调度单元 1102 的形态进行限定。

可选地，调度单元 1102 还用于根据应用的指令集的类型从多个处理器中选择目标处理器。

可选地，在所述选择目标处理器之前，调度单元 1101 还用于，从多个调度域中选择所述目标调度域，应用的指令集的类型是目标调度域支持的指令集集合中的多个指令集的子集。其中，目标处理器属于目标调度域，目标调度域支持的指令集集合与目标处理器支持的指令集集合相同。

可选地，目标调度域仅包括所述目标处理器，或者目标调度域包括目标处理器以及其他处理器，其中，其他处理器支持的指令集集合与目标处理器支持的指令集集合相同。

可选地，调度装置 1100 还包括装载单元 1103，装载单元 1103 用于将任务的指令集的类型写入任务的任务控制块，任务的指令集的类型与所述应用的指令集的类型相同；则获取单元 1101 还用于从任务控制块中获取任务的指令集的类型。其中，应用包括至少一个任务。装载单元 1103 具体实现可参考图 5 中 S503 的步骤，这里不再赘述。在产品的实现过程中，装载单元 1103 可以是图 2 所示的装载器 132，也可以是其他装载器，本实施例不对装载单元 1103 的形态进行限定。

可选地，调度单元 1102 还用于将任务分配至目标处理器执行。

可选地，调度单元 1102 还用于当目标处理器支持多线程并行处理时，将任务分配至目标处理器的一个处理单元执行。其中，目标处理器包括至少两个处理单元。

可选地，调度单元 1102 还用于当目标处理器不支持多线程并行处理时，将任务分配给目标处理器的一个处理器核。其中，目标处理器包括至少一个处理器核。

可选地，装载单元 1103 还用于当目标处理器执行任务失败时，将目标调度域的标识存入任务的任务控制块的第一存储单元中，其中，第一存储单元中存储的标识用于指示目标调度域包括的处理器不能被分配用于执行应用。具体实现可参考图 8 中 S803 的步骤，这里不再赘述。

可选地，装载单元 1103 还用于当目标处理器执行任务成功时，将目标调度域的标识存入任务的任务控制块的第二存储单元中，其中，第二存储单元中存储的标识用于指示目标调度

域包括的处理器能被分配用于执行应用。具体实现可参考图 9 中 S903 的步骤, 这里不再赘述。

可选地, 任务为应用生成的进程或线程。

可选地, 获取单元 1101 还包括编译单元 1104 和解析单元 1105, 当应用可编译时, 编译单元 1104 用于通过编译选项获取所述应用的指令集类型, 具体实现可参考图 5 中 S501 至 S502 的步骤, 这里不再赘述。当应用不可编译时, 解析单元 1105 用于通过异常指令获取应用的指令集类型, 具体实现可参考图 7 中 S702 的步骤, 这里不再赘述。在产品的实现过程中, 编译单元 1104 可以是图 2 所示的编译器 131, 也可以是其他编译器, 本实施例不对编译单元 1104 的形态进行限定。解析单元 1105 可以是图 2 所示的解析器 134, 也可以是其他解析器, 本实施例不对解析单元 1105 的形态进行限定。

综上所述, 本申请实施例提供的调度装置 1100, 可以识别应用的任务中指令集的不同, 将任务调度到可以执行该指令集的处理单元中, 提高应用执行成功的概率。

本申请还提供一种调度系统, 包括用于执行本申请提出的调度方法的调度装置和多个处理器, 其中, 多个处理器中至少一个处理器支持的指令集集合中至少有一个指令集的类型与其他处理器支持的指令集集合中的指令集的类型不同。

可选地, 调度装置可以与多个处理器位于不同的计算设备。

可选地, 多个处理器中至少一个处理器也可以与其他的处理器位于不同的计算设备上。

上述实施例, 可以全部或部分地通过软件、硬件、固件或其他任意组合来实现。当使用软件实现时, 上述实施例可以全部或部分地以计算机程序产品的形式实现。所述计算机程序产品包括一个或多个计算机指令。在计算机上加载或执行所述计算机程序指令时, 全部或部分地产生按照本申请实施例所述的流程或功能。所述计算机可以为通用计算机、专用计算机、计算机网络、或者其他可编程装置。所述计算机指令可以存储在计算机可读存储介质中, 或者从一个计算机可读存储介质向另一个计算机可读存储介质传输, 例如, 所述计算机指令可以从一个网站站点、计算机、服务器或数据中心通过有线(例如同轴电缆、光纤、数字用户线(digital subscriber line, DSL))或无线(例如红外、无线、微波等)方式向另一个网站站点、计算机、服务器或数据中心进行传输。所述计算机可读存储介质可以是计算机能够存取的任何可用介质或者是包含一个或多个可用介质集合的服务器、数据中心等数据存储设备。所述可用介质可以是磁性介质(例如, 软盘、硬盘、磁带)、光介质(例如, DVD)、或者半导体介质。半导体介质可以是固态硬盘(solid state drive, SSD)。

以上所述, 仅为本申请的具体实施方式。熟悉本技术领域的技术人员根据本申请提供的具体实施方式, 可想到变化或替换, 都应涵盖在本申请的保护范围之内。

权 利 要 求 书

1.一种调度方法，其特征在于，所述方法由计算设备执行，所述计算设备包括多个处理器，所述多个处理器中至少一个处理器支持的指令集集合中至少有一个指令集的类型与其他处理器支持的指令集集合中的指令集的类型不同，所述方法包括：

获取应用的指令集的类型；

从所述多个处理器中选择目标处理器，所述应用的指令集的类型是所述目标处理器支持的指令集集合中的多个指令集的类型子集；

将所述应用分配至所述目标处理器执行。

2.根据权利要求1所述的方法，其特征在于，所述从所述多个处理器中选择目标处理器，具体包括：

根据所述应用的指令集的类型从所述多个处理器中选择目标处理器。

3.根据权利要求1-2任一所述的方法，其特征在于，所述目标处理器属于目标调度域，所述目标调度域支持的指令集集合与所述目标处理器支持的指令集集合相同，在所述选择目标处理器之前，还包括：

从多个调度域中选择所述目标调度域，所述应用的指令集的类型是所述目标调度域支持的指令集集合中的多个指令集的类型子集。

4.根据权利要求3所述的方法，其特征在于，

所述目标调度域仅包括所述目标处理器，或者所述目标调度域包括所述目标处理器以及其他处理器，所述其他处理器支持的指令集集合与所述目标处理器支持的指令集集合相同。

5.根据权利要求1-4任一所述的方法，其特征在于，所述应用包括至少一个任务，所述方法还包括：

将所述任务的指令集的类型写入所述任务的任务控制块，所述任务的指令集的类型与所述应用的指令集的类型相同；

所述获取应用的指令集的类型具体包括：从所述任务控制块中获取所述任务的指令集的类型。

6.根据权利要求5所述的方法，其特征在于，所述将所述应用分配至所述目标处理器执行，具体包括：

将所述任务分配至所述目标处理器执行。

7.根据权利要求6所述的方法，其特征在于，所述目标处理器包括至少两个处理单元，所述将所述任务分配至所述目标处理器执行，具体包括：

当所述目标处理器支持多线程并行处理时，将所述任务分配至所述目标处理器的一个处理单元执行。

8.根据权利要求6所述的方法，其特征在于，所述目标处理器包括至少一个处理器核，所述将所述任务分配至所述目标处理器执行，具体包括：

当所述目标处理器不支持多线程并行处理时，将所述任务分配给所述目标处理器的一个处理器核。

9.根据权利要求 6-8 任一所述的方法，其特征在于，所述目标处理器属于目标调度域，所述目标调度域支持的指令集集合与所述目标处理器支持的指令集集合相同，所述方法还包括：

当所述目标处理器执行所述任务失败时，将所述目标调度域的标识存入所述任务的任务控制块的第一存储单元中，所述第一存储单元中存储的所述标识用于指示所述目标调度域包括的处理器不能被分配用于执行所述应用。

10.根据权利要求 6-9 任一所述的方法，其特征在于，所述目标处理器属于目标调度域，所述目标调度域支持的指令集集合与所述目标处理器支持的指令集集合相同，所述方法还包括：

当所述目标处理器执行所述任务成功时，将所述目标调度域的标识存入所述任务的任务控制块的第二存储单元中，所述第二存储单元中存储的所述标识用于指示所述目标调度域包括的处理器能够被分配用于执行所述应用。

11.根据权利要求 4-10 任一所述的方法，其特征在于，所述任务为所述应用生成的进程或线程。

12.根据权利要求 1-11 任一所述的方法，所述获取待执行应用的指令集类型，具体包括：

当所述应用可编译时，通过编译选项获取所述应用的指令集类型；

或者，当所述应用不可编译时，通过异常指令获取所述应用的指令集类型。

13.一种计算设备，其特征在于，所述计算设备包括多个处理器，所述多个处理器中至少一个处理器支持的指令集集合中至少有一个指令集的类型与其他处理器支持的指令集集合中的指令集的类型不同：

所述多个处理器中的主处理器用于：

获取应用的指令集的类型；

从所述多个处理器中选择目标处理器，所述应用的指令集的类型是所述目标处理器支持的指令集集合中的多个指令集的类型子集；

将所述应用分配至所述目标处理器执行。

14.根据权利要求 13 所述的计算设备，其特征在于，所述主处理器还用于：

根据所述应用的指令集的类型从所述多个处理器中选择目标处理器。

15.根据权利要求 13-14 任一所述的计算设备，其特征在于，所述目标处理器属于目标调度域，所述目标调度域支持的指令集集合与所述目标处理器支持的指令集集合相同，在所述选择目标处理器之前，所述主处理器还用于：

从多个调度域中选择所述目标调度域，所述应用的指令集的类型是所述目标调度域支持的指令集集合中的多个指令集的类型子集。

16.根据权利要求 15 所述的计算设备，其特征在于，

所述目标调度域仅包括所述目标处理器，或者所述目标调度域包括所述目标处理器以及其他处理器，所述其他处理器支持的指令集集合与所述目标处理器支持的指令集集合相同。

17.根据权利要求 13-16 任一所述的计算设备，其特征在于，所述应用包括至少一个任务，所述主处理器还用于：

将所述任务的指令集的类型写入所述任务的任务控制块，所述任务的指令集的类型与所述应用的指令集的类型相同；

从所述任务控制块中获取所述任务的指令集的类型。

18.根据权利要求 17 所述的计算设备，其特征在于，所述主处理器还用于：

将所述任务分配至所述目标处理器执行。

19.根据权利要求 18 所述的计算设备，其特征在于，所述目标处理器包括至少两个处理单元，所述主处理器还用于：

当所述目标处理器支持多线程并行处理时，将所述任务分配至所述目标处理器的一个处理单元执行。

20.根据权利要求 18 所述的计算设备，其特征在于，所述目标处理器包括至少一个处理器核，所述主处理器还用于：

当所述目标处理器不支持多线程并行处理时，将所述任务分配给所述目标处理器的一个处理器核。

21.根据权利要求 18-20 任一所述的计算设备，其特征在于，所述目标处理器属于目标调度域，所述目标调度域支持的指令集集合与所述目标处理器支持的指令集集合相同，所述主处理器还用于：

当所述处理器执行所述任务失败时，将所述目标调度域的标识存入所述任务的任务控制块的第一存储单元中，所述第一存储单元中存储的所述标识用于指示所述目标调度域包括的处理器不能被分配用于执行所述应用。

22.根据权利要求 18-21 任一所述的计算设备，其特征在于，所述目标处理器属于目标调度域，所述目标调度域支持的指令集集合与所述目标处理器支持的指令集集合相同，所述主处理器还用于：

当所述处理器执行所述任务成功时，将所述目标调度域的标识存入所述任务的任务控制块的第二存储单元中，所述第二存储单元中存储的所述标识用于指示所述目标调度域包括的处理器能够被分配用于执行所述应用。

23.根据权利要求 15-22 任一所述的计算设备，其特征在于，所述任务为所述应用生成的进程或进程。

24.根据权利要求 13-23 任一所述的计算设备，所述主处理器还用于：

当所述应用可编译时，通过编译选项获取所述应用的指令集类型；
或者，当所述应用不可编译时，通过异常指令获取所述应用的指令集类型。

25.一种调度装置，其特征在于，所述调度装置包括获取单元和调度单元：

所述获取单元，用于获取应用的指令集的类型；

所述调度单元，用于从计算设备的多个处理器中选择目标处理器，所述多个处理器中至少一个处理器支持的指令集集合中至少有一个指令集的类型与其他处理器支持的指令集集合中的指令集的类型不同，所述应用的指令集的类型是所述目标处理器支持的指令集集合中的多个指令集的类型子集；并用于将所述应用分配至所述目标处理器执行。

26. 根据权利要求 25 所述的调度装置，其特征在于，所述调度单元还用于：
根据所述应用的指令集的类型从所述多个处理器中选择目标处理器。

27. 根据权利要求 25-26 任一所述的调度装置，其特征在于，所述目标处理器属于目标调度域，所述目标调度域支持的指令集集合与所述目标处理器支持的指令集集合相同，在所述选择目标处理器之前，所述调度单元还用于：

从多个调度域中选择所述目标调度域，所述应用的指令集的类型是所述目标调度域支持的指令集集合中的多个指令集的类型子集。

28. 根据权利要求 27 所述的调度装置，其特征在于，所述目标调度域仅包括所述目标处理器，或者所述目标调度域包括所述目标处理器以及其他处理器，所述其他处理器支持的指令集集合与所述目标处理器支持的指令集集合相同。

29. 根据权利要求 25-28 任一所述的调度装置，其特征在于，所述应用包括至少一个任务，所述调度装置还包括装载单元，所述装载单元用于：

将所述任务的指令集的类型写入所述任务的任务控制块，所述任务的指令集的类型与所述应用的指令集的类型相同；

所述获取单元还用于：从所述任务控制块中获取所述任务的指令集的类型。

30. 根据权利要求 29 所述的调度装置，其特征在于，所述调度单元还用于：
将所述任务分配至所述目标处理器执行。

31. 根据权利要求 30 所述的调度装置，其特征在于，所述目标处理器包括至少两个处理单元，所述调度单元还用于：

当所述目标处理器支持多线程并行处理时，将所述任务分配至所述目标处理器的一个处理单元执行。

32. 根据权利要求 30 所述的调度装置，其特征在于，所述目标处理器包括至少一个处理器核，所述调度单元还用于：

当所述目标处理器不支持多线程并行处理时，将所述任务分配给所述目标处理器的一个处理器核。

33.根据权利要求 30-32 任一所述的调度装置，其特征在于，所述目标处理器属于目标调度域，所述目标调度域支持的指令集集合与所述目标处理器支持的指令集集合相同，所述装载单元还用于：

当所述目标处理器执行所述任务失败时，将所述目标调度域的标识存入所述任务的任务控制块的第一存储单元中，所述第一存储单元中存储的所述标识用于指示所述目标调度域包括的处理器不能被分配用于执行所述应用。

34.根据权利要求 30-33 任一所述的调度装置，其特征在于，所述目标处理器属于目标调度域，所述目标调度域支持的指令集集合与所述目标处理器支持的指令集集合相同，所述装载单元还用于：

当所述目标处理器执行所述任务成功时，将所述目标调度域的标识存入所述任务的任务控制块的第二存储单元中，所述第二存储单元中存储的所述标识用于指示所述目标调度域包括的处理器能够被分配用于执行所述应用。

35.根据权利要求 29-34 任一所述的调度装置，其特征在于，所述任务为所述应用生成的进程或线程。

36.根据权利要求 25-35 任一所述的调度装置，所述获取单元包括编译单元和解析单元：
当所述应用可编译时，所述编译单元用于通过编译选项获取所述应用的指令集类型；
当所述应用不可编译时，所述解析单元用于通过异常指令获取所述应用的指令集类型。

37.一种调度系统，所述调度系统包括如权利要求 25 至 36 任一所述的调度装置和多个处理器，所述多个处理器中至少一个处理器支持的指令集集合中至少有一个指令集的类型与其他处理器支持的指令集集合中的指令集的类型不同。

38.根据权利要求 37 所述的调度系统，其特征在于，所述调度装置与所述多个处理器位于不同的设备上。

39.根据权利要求 37-38 任一所述的调度系统，其特征在于，所述多个处理器中至少一个处理器与其他处理器位于不同的设备上。

40.一种计算机可读存储介质，其特征在于，所述计算机可读存储介质中包括指令，当其在计算机上运行时，使得计算机执行权利要求 1 至 12 中任一所述的方法的操作步骤。