

说明书

资源调度方法、作业处理方法、调度器、系统及相关设备

本申请要求于 2023 年 4 月 25 日提交国家知识产权局、申请号为 202310473607.0、申请名称为“资源调度方法、作业处理方法、调度器、系统及相关设备”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

本申请涉及计算机技术领域，尤其涉及一种资源调度方法、作业处理方法、调度器、系统及相关设备。

背景技术

目前，诸如消息传递接口（message passing interface, MPI）作业等跨多个节点运行的作业，通常要求每个节点具有相同的计算资源配置，由此，各个节点能够利用配置的计算资源分别执行该作业，并能够近似在同一时刻达到同步点（如同步各个节点得到的作业执行结果）。

通常情况下，在接收到用户提交的作业后，调度器可以为该作业调度同一型号的一个或多个处理器，如调度多个中央处理器（central processing unit, CPU）等，以便利用该同一型号的一个或多个处理器并行处理该作业。

但是，实际应用场景中，即使调度器选择调度同一型号的一个或多个处理器来并行处理该作业，该多个处理器也会在不同的时刻到达同步点，即部分处理器执行作业的速率较快，另一部分处理器执行作业的速率较慢，这使得执行速率较快的处理器需要等待其它处理器执行作业并到达同步点，从而这部分执行速率较快的处理器可能会在较长的等待时间段内处于闲置状态，也即这部分处理器的算力在较长的等待时间段内发生浪费。

因此，目前亟需一种资源调度方法，以尽可能避免部分处理器的算力发生长时间的浪费。

发明内容

本申请提供了一种资源调度方法，以尽可能避免部分处理器的算力发生长时间的浪费。此外，本申请还提供了作业处理方法、调度器、作业处理设备、作业处理系统、计算机可读存储介质以及计算机程序产品。

第一方面，本申请提供一种资源调度方法，包括：调度器获取待处理的第一作业，该第一作业例如可以是 MPI 作业等；然后，调度器根据算力拓扑从资源池包括的一个或多个处理器中确定多个第一处理器，该算力拓扑用于指示一个或多个处理器的算力，其中，处理器的算力例如可以通过处理器执行测试作业所需消耗的时长进行度量，并且，所确定出的多个第一处理器的算力之间的差值不超过阈值；最后，调度器为第一作业调度该多个第一处理器，该第一作业基于多个第一处理器并行执行。

由于为第一作业调度的多个第一处理器的算力之间的偏差较小，具体为不同第一处理器的算力之间的差值不超过阈值，因此，该多个第一处理器并行执行第一作业时，各个第一处理器执行第一作业到达同步点的时刻相近，从而可以减少部分执行作业效率较高的第一处理器的等待时长，以此可以实现避免部分第一处理器的算力被长时间浪费。

在一种可能的实施方式中，调度器在根据算力拓扑确定多个第一处理器时，具体可以根据算力拓扑确定多个候选集合，不同候选集合包括的处理器的数量相同；然后，调度器确定多个候选集合中每个候选集合对应的算力拓扑距离，算力拓扑距离用于指示候选集合包括的一个或多个处理器的算力之间的最大差值，从而调度器可以从多个候选集合中选择第一候选集合，第一候选集合包括多个第一处理器，多个候选集合中第一候选集合的算力拓扑距离最小。如此，调度器通过将算力拓扑距离最小的候选集合中的一个或多个处理器调度给第一作业，可以使得该多个处理器在执行第一作业并到达同步点的时刻之间的偏差尽可能小，从而可以进一步减少部分处理器的算力被浪费的时长，有利于提高资源利用率。

在一种可能的实施方式中，多个第一处理器位于同一计算单元。如此，调度器可以优先将亲和力较高的一个或多个处理器调度给第一作业，这使得后续多个第一处理器之间交互各自针对该第一作业的执行结果

时, 可以提高通信效率、减少通信开销。

在一种可能的实施方式中, 调度器还可以获取待处理的第二作业, 该第二作业也可以是 MPI 作业, 并且, 当资源池中的多个处理器满足资源调整条件, 如当前资源池中剩余可用的处理器之间的差异较大, 不满足第二作业的需求等, 调度器根据算力拓扑将调度给第一作业的多个第一处理器中的第一目标处理器调整为资源池中的第二目标处理器, 并将资源池剩余可用的多个第二处理器调度给第二作业, 多个第二处理器包括第一目标处理器。如此, 通过对已调度给第一作业的部分处理器进行调整以及将其重调度给第二作业, 可以提高调度器针对多个作业进行资源调度的整体效果。

在一种可能的实施方式中, 调度器在将调度给第一作业的多个第一处理器中的第一目标处理器调整为资源池中的第二目标处理器时, 具体可以是当资源池中的多个处理器满足资源调整条件, 且第一作业在第一目标处理器上的执行时长小于预设时长, 根据算力拓扑将调度给第一作业的多个第一处理器中的第一目标处理器调整为资源池中的第二目标处理器。如此, 在调整调度给第一作业的部分处理器之前, 可以结合第一作业的执行时长来综合考虑调整策略, 以此可以进一步提高资源调度的整体效果, 避免在部分处理器执行第一作业较长时间后被重调度给第二作业, 从而可以避免该部分处理器的算力被长时间浪费 (由于该部分处理器执行第一作业未到达同步点, 因此可以视为该部分处理器的算力在执行第一作业的时间段内发生资源浪费)。

在一种可能的实施方式中, 调度器在获取待处理的第一作业时, 具体可以是获取目标队列中的第一作业, 用于处理目标队列中的作业的处理器基于算力拓扑进行确定。如此, 调度器可以针对指定的队列中的作业, 均调度算力相近的多个处理器进行并行处理, 而对于其它队列中的作业, 可以不要求调度算力相近的多个处理器进行并行处理。或者, 调度器响应于用户针对第一作业的使能操作, 获取第一作业。如此, 调度器可以是在用户的指示下, 为第一作业调度算力相近的多个处理器。

在一种可能的实施方式中, 调度器还可以获取测试作业, 并根据测试作业, 对多个处理器分别进行算力测试, 得到算力测试结果, 从而调度器根据多个处理器分别对应的算力测试结果, 生成算力拓扑。如此, 调度器可以利用测试作业来采集各个处理器的算力, 以便通过该算力测试过程得到算力拓扑。

在一种可能的实施方式中, 第一处理器的算力基于第一处理器执行测试作业的耗时进行确定。如此, 可以利用处理器的执行测试作业的耗时, 实现对处理器的算力的度量, 以便于确定各个处理器的算力。

第二方面, 本申请提供一种作业处理方法, 方法包括: 获取待处理的第一作业; 根据算力拓扑, 从资源池中为第一作业调度多个第一处理器, 资源池包括多个处理器, 算力拓扑用于指示多个处理器的算力, 多个第一处理器的算力之间的差值不超过阈值; 利用多个第一处理器并行执行第一作业。由于并行执行第一作业的多个第一处理器的算力之间的偏差较小, 具体为不同第一处理器的算力之间的差值不超过阈值, 因此, 该多个第一处理器并行执行第一作业时, 各个第一处理器执行第一作业到达同步点的时刻相近, 从而可以减少部分执行作业效率较高的第一处理器的等待时长, 以此可以实现避免部分第一处理器的算力被长时间浪费。

第三方面, 本申请提供一种调度器, 调度器包括: 获取模块, 用于获取待处理的第一作业; 确定模块, 用于根据算力拓扑, 从资源池中确定多个第一处理器, 资源池包括多个处理器, 算力拓扑用于指示多个处理器的算力, 多个第一处理器的算力之间的差值不超过阈值; 调度模块, 用于为第一作业调度多个第一处理器, 第一作业基于多个第一处理器并行执行。

在一种可能的实施方式中, 确定模块, 用于: 根据算力拓扑确定多个候选集合, 不同候选集合包括的处理器数量相同; 确定多个候选集合中每个候选集合对应的算力拓扑距离, 算力拓扑距离用于指示候选集合包括的不同处理器的算力之间的最大差值; 从多个候选集合中选择第一候选集合, 第一候选集合包括多个第一处理器, 多个候选集合中第一候选集合的算力拓扑距离最小。

在一种可能的实施方式中, 多个第一处理器位于同一计算单元。

在一种可能的实施方式中, 获取模块, 还用于获取待处理的第二作业; 调度模块, 还用于当资源池中的多个处理器满足资源调整条件, 根据算力拓扑将调度给第一作业的多个第一处理器中的第一目标处理器调整为资源池中的第二目标处理器; 将资源池剩余可用的多个第二处理器调度给第二作业, 所述多个第二处理器包括所述第一目标处理器, 多个第二处理器包括第一目标处理器。

在一种可能的实施方式中, 调度模块, 具体用于: 当资源池中的多个处理器满足资源调整条件, 且

- 第一作业在第一目标处理器上的执行时长小于预设时长，根据算力拓扑将调度给第一作业的多个第一处理器中的第一目标处理器调整为资源池中的第二目标处理器。
- 在一种可能的实施方式中，获取模块，具体用于：获取目标队列中的第一作业，用于处理目标队列中的作业的处理器基于算力拓扑进行确定；或者，响应于用户针对第一作业的使能操作，获取第一作业。
- 5 在一种可能的实施方式中，获取模块，还用于获取测试作业；根据测试作业，对多个处理器分别进行算力测试，得到算力测试结果；根据多个处理器分别对应的算力测试结果，生成算力拓扑。
- 在一种可能的实施方式中，第一处理器的算力基于第一处理器执行测试作业的耗时进行确定。
- 第三方面提供的调度器，对应于第一方面提供的资源调度方法，故第三方面以及第三方面中任意一种实现方式所具有的技术效果，参见第一方面以及第一方面中相应实现方式所具有的技术效果。
- 10 第四方面，本申请提供一种作业处理设备，作业处理设备包括：获取模块，用于获取待处理的第一作业；调度模块，用于根据算力拓扑，从资源池中为第一作业调度多个第一处理器，资源池包括多个处理器，算力拓扑用于指示多个处理器的算力，多个第一处理器的算力之间的差值不超过阈值；执行模块，用于利用多个第一处理器并行执行第一作业。
- 第四方面提供的作业处理设备，对应于第二方面提供的作业处理方法，故第四方面所具有的技术效果，参见第二方面所具有的技术效果。
- 15 第五方面，本申请提供一种调度器，所述调度器包括处理器、存储器和显示器。所述处理器、所述存储器进行相互的通信。所述处理器用于执行存储器中存储的指令，以使得调度器执行如第一方面或第一方面的任一种实现方式中的资源调度方法。需要说明的是，该存储器可以集成于处理器中，也可以是独立于处理器之外。调度器还可以包括总线。其中，处理器通过总线连接存储器。其中，存储器可以包括可读存储器以及随机存取存储器。
- 20 第六方面，本申请提供一种作业处理系统，包括上述第五方面提供的调度器、以及资源池，所述资源池包括多个处理器。
- 第七方面，本申请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，当其在计算设备上运行时，使得计算设备执行上述第一方面或第一方面的任一种实现方式所述的资源调度方法的操作步骤，或者使得计算设备执行上述第二方面所述的作业处理方法的操作步骤。
- 25 第八方面，本申请提供了一种包含指令的计算机程序产品，当其在计算设备上运行时，使得计算设备执行上述第一方面或第一方面的任一种实现方式所述的资源调度方法的操作步骤，或者使得计算设备执行上述第二方面所述的作业处理方法的操作步骤。
- 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。

30

附图说明

- 图 1 为本申请提供的一示例性作业处理系统的结构示意图；
- 图 2 为处理器 A 以及处理器 B 在不同的时刻到达同步点的示意图；
- 图 3 为本申请提供的一种资源调度方法流程示意图；
- 35 图 4 为本申请提供的一示例性算力拓扑的示意图；
- 图 5 为本申请提供的根据算力拓扑确定 4 个候选集合以及算力拓扑距离的示意图；
- 图 6A 为本申请提供的另一种资源调度方法的流程示意图；
- 图 6B 为调度器 101 为第一作业以及第二作业调度计算资源的示意图；
- 图 7 为本申请提供的一种调度器的结构示意图；
- 40 图 8 为本申请提供的一种作业处理设备的结构示意图；
- 图 9 为本申请提供的一种调度器的硬件结构示意图。

40

具体实施方式

- 为了实现避免部分处理器的算力被长时间浪费，本申请提供了一种资源调度方法，基于资源池中多个处理器的算力，从资源池中确定算力相近的多个第一处理器调度给作业。如此，利用算力相近的多个
- 45 第一处理器来并行执行该作业，可以使得多个处理器分别执行作业时到达同步点的时刻相近，从而可以

减少部分执行作业效率较高的处理器的等待时长，以此可以实现避免部分处理器的算力被长时间浪费。

下面将结合本申请实施例中的附图，对本申请中的技术方案进行描述。

参见图1，为本申请实施例提供的一示例性作业处理系统10的结构示意图。

如图1所示，作业处理系统10包括调度器101以及资源池102，调度器101与资源池102可以通过
5 有线或者无线的方式建立通信连接。实际应用时，作业处理系统10可以是开源系统或者商业系统，相应的，调度器101可以是开源系统中的SLURM、OpenPBS、或YARN等调度工具，或者可以是商业系统中的LSF、或PBS Pro等调度工具。

作业处理系统10还可以对外提供客户端103，客户端103可以是部署于用户侧设备的应用
(application)，或者可以是网络浏览器等，用于接收用户提供的作业，并将作业转发给调度器101。其
10 中，用户提交的作业，可以是要求跨多个节点运行的作业，如MPI作业等，要求利用多个节点并行执行该作业，并在到达同步点时汇总各个节点执行作业所得到的执行结果。举例来说，用户可以提交训练人
工作智能(artificial intelligence, AI)模型的作业，由于训练AI模型的样本数量较多，因此，可以利用
多个计算节点并行训练该AI模型，每个计算节点利用部分样本对该AI模型进行训练，并在完成一轮模
型训练时(也即到达同步点)，各个计算节点汇总当前所训练得到的AI模型的参数，并根据汇总结果对
15 各个计算节点上的AI模型进行统一更新，从而各个计算节点可以继续更新的AI模型继续执行模型训练
过程。

调度器101，用于为客户端103发送的作业进行资源调度，具体可以从资源池103中调度相应的
资源来并行执行该作业。

资源池102，包括多个处理器，如包括多个CPU等，该多个处理器的型号可以相同，也可以不同，
20 从而资源池102中的多个处理器，可以在调度器101的调度下，用于并行执行用户提交的作业。实际应
用时，资源池102还可以包括内存、硬盘等存储资源，以及包括带宽等网络资源，从而调度器101还可
以为用户提交的作业一并调度资源池的存储资源以及网络资源。

通常情况下，调度器101会根据处理器的型号评估该处理器的算力，因此，对于资源池102中具有
相同型号的多个处理器，调度器101会确定该多个处理器的算力相同。但是，实际应用场景中，可能因
25 为生产工艺存在波动或者其它原因，导致同一型号的不同处理器的算力存在差异。比如，在生产同一型
号的处理器A的过程中，生产工艺比较稳定，实际的高速电路与预期设计的高速电路偏差较小；但是，
在生产同一型号的处理器B的过程中，生产工作出现波动，实际的高速电路与预期设计的高速电路偏差
较大，这使得处理器A的算力高于处理器B的算力。即，在相同的运行条件下，处理器A执行作业的
效率高于处理器B执行作业的效率。相应地，当调度器101调度同一型号的处理器A以及处理器B并
30 行执行作业时，如处理器A可能耗时12个小时执行作业并到达同步点，并在完成作业后开始等待处理
器B执行完成作业并到达同步点，假设处理器B耗时18个小时执行作业并到达同步点，则处理器A在
执行作业到达同步点后需要额外等待6个小时，如图2上方所示，从而导致处理器A的算力在这6个小
时内会被闲置，也即处理器A的算力被长时间的浪费。并且，当调度器101调度更多数量的同一型号的
处理器来执行该作业时，就会存在更多数量的处理器的算力被长时间的浪费。

因此，本申请提供了一种资源调度方法，以避免处理器A的算力被长时间浪费。具体实现时，调度
器101在获取到用户提交的作业后，可以根据用于指示资源池102中多个处理器的算力的算力拓扑，从
资源池102中确定用于并行执行该作业的多个处理器，该多个处理器的算力之间的差值不超过阈值，如
该多个处理器的算力中最大算力与最小算力之间的偏差小于阈值等，从而调度器101为该作业调度该多
个处理器，以便利用该多个处理器并行执行该作业。

由于为作业调度的多个处理器的算力之间的偏差较小，具体不超过阈值，这使得在多个处理器并行
执行作业时，各个处理器执行作业到达同步点的时刻相近，从而可以减少部分执行作业效率较高的处理
器的等待时长，以此可以实现避免部分处理器的算力被长时间浪费。仍以调度器101调度资源池102中
的两个处理器并行执行作业为例，假设调度器101调度资源池102中的处理器A以及处理器C并行执
行作业，并且，处理器A与处理器C之间的算力差异较小，这使得处理器A可能耗时12个小时执行作
45 业并到达同步点，处理器C可能耗时12.1个小时执行作业并到达同步点，因此，处理器A在执行作业
并到达同步点后仅需额外等待0.1个小时(即6分钟)，如图2下方所示，这相比于处理器A在执行作

业到达同步点后需要额外等待 6 个小时（等待上述处理器 B 执行作业并到达同步点）而言，可以有效减少处理器 A 的算力被浪费的时间，而且，也能提高该作业被执行的整体效率。

示例性地，上述调度器 101 可以通过软件或者硬件实现。

在第一种示例中，当通过软件实现时，调度器 101 例如可以通过虚拟机、容器、计算引擎中的至少一种实现等。

在第二种示例中，当通过硬件实现时，调度器 101 可以通过包括处理器的至少一个物理设备实现。其中，处理器可以是中央处理器（central processing unit, CPU），以及专用集成电路（application-specific integrated circuit, ASIC）、可编程逻辑器件（programmable logic device, PLD）、复杂程序逻辑器件（complex programmable logical device, CPLD）、现场可编程门阵列（field-programmable gate array, FPGA）、通用阵列逻辑（generic array logic, GAL）、片上系统（system on chip, SoC）、软件定义架构（software-defined infrastructure, SDI）芯片、人工智能（artificial intelligence, AI）芯片、数据处理单元（Data processing unit, DPU）等任意一种处理器或其任意组合。并且，调度器 101 中所包括的处理器数量可以是一个或者多个，所包括的处理器种类可以是一种或者多种，具体可以根据实际应用的业务需求设定处理器的数量和种类，本实施例对此并不进行限定。

值得注意的是，上述图 1 所示的作业处理系统 10 仅作为一种示例性说明，在其它可适用的作业处理系统中，作业处理系统可以具有更多的设备，如用于存储调度日志的设备等，以支持作业处理系统具有更多的功能。或者，作业处理系统可以为一部分作业（如付费用户的作业）调度算力相近的多个处理器来并行执行该作业，并可以为另一部分作业（如未付费用户的作业）调度算力差异较大的多个处理器来并行执行该作业等。或者，作业处理系统还可以为用户提交的部分作业（单节点运行的作业）调度资源池 102 中的一个处理器来执行作业等。本申请中对于上述作业处理系统的具体结构及其所适用的应用场景并不进行限定。

为便于理解，下面结合附图，对本申请提供的资源调度方法的实施例进行描述。

参见图 3，图 3 为本申请实施例提供的一种资源调度方法的流程示意图，该方法可以应用于图 1 所述的作业处理系统 10，或者可以应用于其它可适用的作业处理系统。为便于说明，本实施例中应用于图 1 所示的作业处理系统 10 为例进行示例性说明。

其中，图 3 所示的资源调度方法具体可以包括：

S301：调度器 101 获取待处理的第一作业。

其中，第一作业可以是要求跨多个节点运行的作业，如 MPI 作业等，实际应用时，第一作业可以是训练 AI 模型的作业、分布式计算的作业等。

本实施例中，提供了以下几种示例性地实施方式。

在第一种可能的实施方式中，作业处理系统 10 对外提供客户端 103，并且，用户可以通过客户端 103 向作业处理系统 10 提交第一作业。具体实现时，客户端 103 可以向用户呈现用于提交作业的交互界面，以便用户在该交互界面上上传第一作业并进行提交（如通过命令行提交第一作业）。然后，客户端 103 可以将用户提交的第一作业转发给调度器 101，以便调度器 101 为该第一作业调度相应的多个处理器。

实际应用场景中，由于用户并非对所有提交的作业都会要求多个节点执行作业的耗时差异较小，因此，用户在交互界面上提交第一作业时，还可以针对该第一作业执行使能操作，例如为该第一作业添加算力调度的优化标签，该优化标签用于指示基于算力差异较小的多个处理器来并行执行该第一作业。从而，客户端 103 可以将该第一作业发送给调度器 101，并基于用户执行的使能操作（如添加优化标签）请求调度器 101 为该作业调度算力差异较小的多个处理器。

在第二种可能的实施方式中，客户端 103 可以将用户提交的第一作业，写入调度器中的目标队列，从而，调度器 101 可以从该目标队列中获取该第一作业，并针对该第一作业调度算力差异较小的多个处理器。实际应用时，并非用户提交的所有作业都会要求多个节点执行作业的耗时差异较小，因此，调度器 101 中可以配置有多个队列，每个队列可以均可以用于缓存用户提交的作业，该多个队列中的部分队列为上述目标队列。如此，对于目标队列缓存的各个作业，调度器 101 可以为其调度算力差异较小的多个处理器并行执行，而对于其余队列缓存的各个作业，调度器 101 可以调度一个处理器执行该作业，或

者允许为其调度算力差异较大的多个处理器并行执行。其中，调度器 101 中的目标队列，可以通过加载相应的配置文件完成配置。

需要说明的是，上述多种实现方式仅作为一些示例性说明，在其它实施例中，调度器 101 也可以采用其它实现方式获取待处理的第一作业，对此并不进行限定。

- 5 S302：调度器 101 根据算力拓扑，从资源池 102 中确定多个第一处理器，其中，资源池 102 包括多个处理器，该算力拓扑用于指示该多个处理器的算力，所确定的多个第一处理器的算力之间的差值不超过阈值。

S303：调度器 101 为第一作业调度多个第一处理器，该第一作业基于多个第一处理器并行执行。

- 10 本实施例中，针对资源池 102 中的多个处理器，可以预先采集该多个处理器分别具有的算力，并基于各个处理器的算力，生成相应的算力拓扑。其中，处理器的算力，可以通过该处理器执行测试作业的耗时进行度量，耗时越高，表征处理器的算力越低，耗时越短，表征处理器的算力越高。在其它实施例中，处理器的算力，也可以通过其它方式进行表征，对此并不进行限定。测试作业，即为用于对处理器的算力进行测试所使用的作业。

- 15 作为一种生成算力拓扑的实现示例，假设由调度器 101 生成算力拓扑，则，调度器 101 可以获取测试作业，该测试作业例如可以是由技术人员提供给调度器 101。然后，调度器 101 可以根据该测试作业，对资源池 102 中的各个处理器分别进行算力测试，得到算力测试结果。其中，资源池 102 中的多个处理器，可以是同一型号的处理器，也可以是不同型号的处理器。

- 20 具体地，调度器 101 可以将该测试作业分发给各个处理器；各个处理器在具有充足的内存、带宽等资源的条件下，基于相同的作业执行程序、运行功率执行测试作业，并计时该处理器执行测试作业所需的时长，再将执行测试作业所需的时长作为算力测试结果返回给调度器 101。这样，调度器 101 可以根据各个处理器分别对应的算力测试结果，生成算力拓扑。比如，如图 4 所示，调度器 101 可以对各个处理器执行测试作业所需的时长进行排序，并根据排序情况生成算力拓扑，图 4 所示的算力拓扑中包括多个节点，每个节点包括一个处理器的相关信息，该相关信息具体可以是[计算单元标识，处理器标识，计算时长]。其中，计算单元标识，用于指示处理器所属的计算单元，该计算单元例如可以是芯片或者计算机，即，计算单元标识可以用于指示处理器属于哪个芯片或者属于哪个计算机。

- 25 在其它实施方式中，也可以是由作业处理系统 10 中的其它设备生成算力拓扑，并将生成的算力拓扑发送给调度器 101，对此并不进行限定。并且，算力拓扑的表现形式，也不限于上述实现示例。

- 30 在获取到第一作业后，调度器 101 可以根据算力拓扑，从资源池 102 中确定算力相近的多个处理器，具体可以是根据第一作业对于处理器数量的需求，从资源池 102 中的多个处理器中筛选出相应数量、且算力差值不超过阈值的多个处理器，为便于区分和描述，以下称之为第一处理器。

- 35 在一种可能的实施方式中，调度器 101 可以根据算力拓扑，确定多个候选集合，其中，不同候选集合包括的处理器数量相同。示例性地，每个候选集合包括的处理器数量，为并行执行第一作业所要求的处理器的数量。比如，假设算力拓扑如图 4 所示，并且，并行执行第一作业所需的处理器的数量为 3，则调度器 101 可以根据算力拓扑，确定 4 个候选集合，如图 5 所示。

- 40 然后，调度器 101 可以确定该多个候选集合中每个候选集合对应的算力拓扑距离，该算力拓扑距离用于指示该候选集合包括的不同处理器的算力之间的最大差值。以处理器执行测试作业所需的时长来度量处理器的算力为例，每个候选集合对应的算力拓扑距离，即为该候选集合包括的多个处理器对应的最大计算时长与最小计算时长之间的差值，或者根据该差值计算得到算力拓扑距离，如通过对该差值进行归一化处理得到算力拓扑距离等。

- 45 最后，调度器 101 可以从多个候选集合中选择第一候选集合，并将该第一候选集合中的多个处理器作为第一处理器分配给第一作业。具体实现时，调度器 101 可以将该多个第一处理器与第一作业进行绑定，以便各个第一处理器能够根据绑定关系执行第一作业。

- 在第一种实现示例中，该第一候选集合包括多个第一处理器，在多个候选集合中，该第一候选集合的算力拓扑距离最小。如此，调度器 101 可以为第一作业确定资源池 102 中确定算力最接近的多个第一处理器，以便后续为该第一作业调度该多个第一处理器来避免部分处理器的算力被长时间浪费。比如，假设图 5 中的 4 个候选集合分别对应的算力拓扑距离依次为：2、4、1、3，则，由于候选集合 3 对应的

算力拓扑距离最小，因此，调度器 101 可以将候选集合 3 包括的处理器作为上述第一处理器分配给第一作业。

其中，当多个候选集合中，同时存在两个或者两个以上的候选集合的算力拓扑距离最小。则，调度器 101 可以从这两个或者两个以上的候选集合中随机选择一个候选集合，作为上述第一候选集合；或者，
5 调度器 101 可以从这两个或者两个以上的候选集合中，选择多个处理器的平均算力较大的候选集合，作为上述第一候选集合；或者，调度器 101 可以从这两个或者两个以上的候选集合中，优先选择多个处理器位于同一计算单元的候选集合，作为上述第一候选集合等。

在第二种实现示例中，实际应用场景中，第一作业可能对于处理器的最低算力可能存在要求，或者对于执行该第一作业的整体耗时可能存在要求，因此，调度器 101 也可以先从多个候选集合中确定能够
10 满足该第一作业的算力需求的 M 个候选集合，M 为大于 0 的正整数，再根据该 M 个候选集合分别对应的算力拓扑距离，从 M 个候选集合中确定出算力拓扑距离最小的候选集合，并将其作为上述第一候选集合。

在第三种实现示例中，调度器 101 所确定出的多个候选集合，均为处理器的算力满足第一作业的需求的候选集合，则，调度器 101 可以确定该多个候选集合中，算力拓扑距离小于阈值的一个或者多个候
15 选集合，再从该一个或者多个候选集合中，随机选择一个候选集合作为上述第一候选集合。或者，调度器 101 可以从该一个或者多个候选集合中选择多个处理器的平均算力较大的候选集合作为上述第一候选集合。比如，假设候选集合 1 以及候选集合 3 分别对应的算力拓扑距离均小于阈值，则调度器 101 可以优先选择平均算力较大的候选集合 3 作为上述第一候选集合。或者，调度器 101 可以从该一个或者多个
20 候选集合中选择多个处理器位于同一计算单元的候选集合，作为上述第一候选集合等。比如，假设候选集合 1 以及候选集合 3 分别对应的算力拓扑距离均小于阈值，并且，候选集合 1 中的多个处理器均位于同一芯片，候选集合 2 中的多个处理器位于不同的芯片，则调度器 101 可以选择候选集合 1 作为上述第一候选集合。这样，后续多个第一处理器之间交互各自针对该第一作业的执行结果时，可以提高通信效率、减少通信开销

可以理解，上述关于确定第一候选集合的多种实现示例，仅作为一些示例性说明，实际应用时，调度器 101 也可以采用其它方式确定第一候选集合，本实施例对此并不进行限定。
25

调度器 101 在为第一作业调度多个第一处理器后，该多个第一处理器可以开始并行执行第一作业。因此，本实施例还可以进一步包括：

S304：多个第一处理器并行执行第一作业。

可以理解，为第一作业调度的多个第一处理器的算力之间的偏差较小，具体为不同第一处理器的算力之间的差值不超过阈值，因此，该多个第一处理器并行执行第一作业时，各个第一处理器执行第一作
30 业到达同步点的时刻相近，从而可以减少部分执行作业效率较高的第一处理器的等待时长，以此可以实现避免部分第一处理器的算力被长时间浪费。

实际应用时，资源池 102 中除了可以包括多个处理器之外，还可以包括存储资源、网络资源等，因此，调度器 101 在为第一作业调度资源时，除了可以基于上述过程为第一作业调度多个处理器之外，还
35 可以根据第一作业对于存储资源以及网络资源的需求，为第一作业调度相应的存储资源以及网络资源。在已经确定为第一作业调度的多个处理器的情况下，基于该多个处理器调度相应的存储资源以及网络资源，在实际场景中存在相关应用，在此不做展开赘述。

上述图 3 所示的实施例中，主要介绍了调度器 101 在资源池 102 中的计算资源充足的情况下，为第一作业调度多个处理器的实现过程。实际应用场景中，随着调度器持续为多个作业进行资源调度，资源池 102 中的多个处理器的资源占用情况不断变化，这使得调度器 101 在为其它作业调度计算资源时，资源池 102 中剩余可调度的多个处理器的算力之间的差异较大，这使得调度器 101 从剩余的多个处理器中
40 为其它作业调度至少一个处理器，可能会出现该各个处理器执行该其它作业并到达同步点的时刻之间存在较大差异。因此，在进一步可能的实施方式中，调度器 101 还可以通过调整已分配的计算资源，来优化为多个作业进行资源调度的整体效果。
45

参见图 6A，示出了本申请实施例提供的另一种资源调度方法的流程示意图。如图 6A 所示，该方法

具体可以包括：

S601：调度器 101 为第一作业调度多个第一处理器。

本实施例中，步骤 S601 的具体实现过程，可以参见上述图 3 所示实施例的相关之处描述，在此不做赘述。

5 S602：调度器 101 获取待处理的第二作业。

本实施例中，第二作业仍为要求跨多个节点运行的作业，如 MPI 作业。

其中，步骤 S602 的具体实现过程，可参见前述步骤 S301 的相关之处描述，在此不做赘述。

S603：当资源池 102 中的多个处理器满足资源调整条件，调度器 101 根据算力拓扑，将调度给第一作业的多个第一处理器中的第一目标处理器调整为资源池 102 中的第二目标处理器。

10 S604：调度器 101 将资源池 102 剩余可用的多个第二处理器调度给第二作业，该多个第二处理器包括第一目标处理器。

调度器 101 在获取到第二作业后，可以先根据算力拓扑，确定资源池 102 当前剩余的多个处理器中，是否存在算力相近（算力之间的差值不超过阈值）的多个处理器来执行第二作业。

15 如果存在，则调度器 101 可以按照上述图 3 所示实施例的资源调度方法，为第二作业调度多个第二处理器。

如果不存在，则，调度器 101 可以确定当前资源池 102 中的多个处理器满足资源调整条件，并以此调整已调度给第一作业的多个处理器。

20 在一种可能的实施方式中，调度器 101 可以先判断将已调度给第一作业的多个第一处理器中的 N（N 为正整数）个第一处理器进行回收后（回收到资源池 102），资源池 102 中的剩余可用的处理器中是否存在算力差值不超过阈值的多个处理器能够满足第二作业的需求。为便于区分，以下将该 N 个第一处理器称之为 N 个第一目标处理器。如果存在，则调度器 101 可以将已调度给第一作业的 N 个第一目标处理器重新调度给第二作业，从当前资源池 102 剩余可用的处理器中选择 N 个第二目标处理器调度给第一作业，该 N 个第二目标处理器可以重新开始执行第一作业。并且，调度器 101 还根据算力拓扑，从资源池 102 剩余可用的处理器中调度一定数量的处理器给第二作业，以使得调度给第二作业的处理器数量满足并行执行第二作业的要求。为便于区分，以下将调度给第二作业的多个处理器称之为第二处理器。其中，调度给第二作业的所有第二处理器的算力之间的差值不超过阈值。如果不存在，则调度器 101 可以从当前资源池 102 中剩余可用的处理器中调度算力尽可能相近的多个第二处理器调度给第二作业，此时，调度给第二作业的多个第二处理器的算力之间的差值允许超过阈值。

30 举例来说，假设调度器 101 为第一作业调度的多个第一处理器为图 6B 所示的处理器 2、处理器 3 以及处理器 4，并且，第二作业要求利用 3 个处理器来并行执行作业，此时，调度器 101 可以判断将处理器 4（即上述第一目标处理器）进行回收后，资源池 102 剩余可用的处理器 1、处理器 4、处理器 5 以及处理器 6 是否存在算力相近的 3 个处理器能够满足第二作业的需求。由于根据算力拓扑，调度器 101 可以确定处理器 4、处理器 5 以及处理器 6 之间的算力距离相近，因此，调度器 101 可以将处理器 4、处理器 5 以及处理器 6 调度给第二作业，并将处理器 1（即上述第二目标处理器）调度给第一作业，此时，调度给第一作业的多个处理器变更为处理器 1、处理器 2 以及处理器 3。

40 示例性地，在从当前资源池 102 中剩余可用的处理器中调度 N 个处理器给第一作业的过程中，调度器 101 可以先判断已调度给第一作业的多个第一处理器所在的计算单元中，是否存在其它 N 个可用的处理器。如果存在，则调度器 101 可以根据算力拓扑，从该计算单元包括的可用的处理器中（即未被调度给其它作业的处理器），选择 N 个处理器调度给该第一作业。这样，在调整为第一作业调度的计算资源后，调度给第一作业的多个处理器的亲和力较高，这使得后续多个第一处理器之间交互各自针对该第一作业的执行结果时，通信效率较高、通信开销较小。如果不存在，则调度器 101 可以根据算力拓扑，从其它计算单元包括的可用的处理器中，选择 N 个处理器调度给第一作业。

45 实际应用时，由于调度给第一作业的第一目标处理器已经执行第一作业但并未到达同步点，因此，新调度给第一作业的第二目标处理器通常会重新开始执行第一作业，这使得第一目标处理器针对该第一作业所执行的操作内容无效，即可以视为该第一目标处理器的算力在执行第一作业的时间段内发生资源浪费。

因此,在进一步可能的实施方式中,调度器 101 还可以结合第一目标处理器执行第一作业的时长,来确定是否将第一目标处理器重新调度给第二作业。具体地,调度器 101 在确定回收 N 个第一目标处理器后,资源池 102 中的剩余可用的处理器中存在算力差值不超过阈值的多个处理器能够满足第二作业的需求的情况下,还可以判断 N 个第一目标处理器执行第一作业的时长是否小于预设时长,该预设时长例如由技术人员预先完成配置,例如可以是根据第一作业被允许拖延执行的最大时长进行确定。若是,调度器 101 可以将该 N 个第一目标处理器作为第二处理器调度给第二作业,从当前资源池 102 中剩余可用的处理器中选择 N 个第二目标处理器调度给第一作业。并且,调度器 101 还根据算力拓扑,从资源池 102 剩余可用的处理器中调度一定数量的第二处理器给第二作业,以使得调度给第二作业的第二处理器的数量满足并行执行第二作业的要求,其中,调度给第二作业的所有第二处理器的算力之间的差值不超过阈值。若否,或者,在回收 N 个第一目标处理器后,资源池 102 中的剩余可用的处理器中不存在算力差值不超过阈值的多个处理器能够满足第二作业的需求,则,调度器 101 可以从当前资源池 102 中剩余可用的处理器中调度算力尽可能相近的多个第二处理器调度给第二作业,此时,调度给第二作业的多个第二处理器的算力之间的差值允许超过阈值。

值得注意的是,上述各实现方式仅作为一些示例性说明,实际应用时,调度器 101 也可以采用其它方式确定资源池 102 中的多个处理器满足资源调整条件,并为第一作业以及第二作业调度相应的计算资源。

本实施例中,调度器 101 在为不同的作业调度算力相近的多个处理器时,不同作业可以对应于不同的阈值,每个作业对应的阈值用于约束调度给该作业的多个处理器的算力之间的最大差值;或者,不同作业可以对应于统一的阈值,对此并不进行限定。

S605: 多个第二处理器并行执行第二作业。

并且,第一作业也是通过多个处理器并行执行,该多个处理器具体为已调度给第一作业但是并未被调整的第一处理器、以及调度器 101 调度给第一作业的 N 个第二目标处理器。

本实施例中,调度器 101 可以对已调度给第一作业的部分处理器进行调整以及将其重调度给第二作业,可以从全局优化为多个作业做出的资源调度,以此可以提高调度器 101 针对多个作业进行资源调度的整体效果。

值得注意的是,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,也属于本申请的保护范围内。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。

以上结合图 1 至图 6B 对本申请实施例提供的资源调度方法进行介绍,接下来结合附图对本申请实施例提供的调度器以及作业调度设备的结构进行介绍。

参见图 7,示出了一种调度器的结构示意图,该调度器 700 包括:

获取模块 701,用于获取待处理的第一作业;

确定模块 702,用于根据算力拓扑,从资源池中确定多个第一处理器,资源池包括多个处理器,算力拓扑用于指示多个处理器的算力,多个第一处理器的算力之间的差值不超过阈值;

调度模块 703,用于为第一作业调度多个第一处理器,第一作业基于多个第一处理器并行执行。

在一种可能的实施方式中,确定模块 702,用于:

根据算力拓扑确定多个候选集合,不同候选集合包括的处理器的数量相同;

确定多个候选集合中每个候选集合对应的算力拓扑距离,算力拓扑距离用于指示候选集合包括的不同处理器的算力之间的最大差值;

从多个候选集合中选择第一候选集合,第一候选集合包括多个第一处理器,多个候选集合中第一候选集合的算力拓扑距离最小。

在一种可能的实施方式中,多个第一处理器位于同一计算单元。

在一种可能的实施方式中,获取模块 701,还用于获取待处理的第二作业;

调度模块 703,还用于:

当资源池中的多个处理器满足资源调整条件,根据算力拓扑将调度给第一作业的多个第一处理器中的第一目标处理器调整为资源池中的第二目标处理器;

将资源池剩余可用的多个第二处理器调度给第二作业，多个第二处理器包括第一目标处理器。

在一种可能的实施方式中，调度模块 703，具体用于：

当资源池中的多个处理器满足资源调整条件，且第一作业在第一目标处理器上的执行时长小于预设时长，根据算力拓扑将调度给第一作业的多个第一处理器中的第一目标处理器调整为资源池中的第二目标处理器。

在一种可能的实施方式中，获取模块 701，具体用于：

获取目标队列中的第一作业，用于处理目标队列中的作业的处理器基于算力拓扑进行确定；

或者，响应于用户针对第一作业的使能操作，获取第一作业。

在一种可能的实施方式中，获取模块 701，还用于：

获取测试作业；

根据测试作业，对多个处理器分别进行算力测试，得到算力测试结果；

根据多个处理器分别对应的算力测试结果，生成算力拓扑。

在一种可能的实施方式中，第一处理器的算力基于第一处理器执行测试作业的耗时进行确定。

由于图 7 所示的调度器 700，对应于上述图 3 以及图 6A 所示实施例中的调度器 101，故图 7 所示的调度器 700 的具体实现方式及其所具有的技术效果，参见上述图 3 以及图 6A 所示实施例中的相关之处描述，在此不做赘述。

参见图 8，示出了一种作业处理设备结构示意图，该作业处理设备 800 包括：

获取模块 801，用于获取待处理的第一作业；

调度模块 802，用于根据算力拓扑，从资源池中为第一作业调度多个第一处理器，资源池包括多个处理器，算力拓扑用于指示多个处理器的算力，多个第一处理器的算力之间的差值不超过阈值；

执行模块 803，用于利用多个第一处理器并行执行第一作业。

图 9 为本申请提供的一种调度器 900 的硬件结构示意图，该调度器 900 例如可以是上述图 3 以及图 6A 所示实施例中的调度器 101 等。

如图 9 所示，所述调度器 900 包括处理器 901、存储器 902、通信接口 903。其中，处理器 901、存储器 902、通信接口 903 通过总线 904 进行通信，也可以通过无线传输等其他手段实现通信。该存储器 902 用于存储指令，该处理器 901 用于执行该存储器 902 存储的指令。进一步的，调度器 900 还可以包括内存单元 905，该内存单元 905 可以通过总线 904 与处理器 901、存储介质 902 以及通信接口 903 连接。其中，该存储器 902 存储程序代码，且处理器 901 可以调用存储器 902 中存储的程序代码执行以下操作：

获取待处理的第一作业；

根据算力拓扑，从资源池中确定多个第一处理器，所述资源池包括多个处理器，所述算力拓扑用于指示所述多个处理器的算力，所述多个第一处理器的算力之间的差值不超过阈值；

为所述第一作业调度所述多个第一处理器，所述第一作业基于所述多个第一处理器并行执行。

应理解，在本申请实施例中，该处理器 901 可以是 CPU，该处理器 901 还可以是其他通用处理器、数字信号处理器（DSP）、专用集成电路（ASIC）、现场可编程门阵列（FPGA）或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立器件组件等。通用处理器可以是微处理器或者是任何常规的处理器等。

该存储器 902 可以包括只读存储器和随机存取存储器，并向处理器 901 提供指令和数据。存储器 902 还可以包括非易失性随机存取存储器。

该存储器 902 可以是易失性存储器或非易失性存储器，或可包括易失性和非易失性存储器两者。其中，非易失性存储器可以是只读存储器（read-only memory，ROM）、可编程只读存储器（programmable ROM，PROM）、可擦除可编程只读存储器（erasable PROM，EPROM）、电可擦除可编程只读存储器（electrically EPROM，EEPROM）或闪存。易失性存储器可以是随机存取存储器（random access memory，RAM），其用作外部高速缓存。通过示例性但不是限制性说明，许多形式的 RAM 可用，例如静态随机存取存储器（static RAM，SRAM）、动态随机存取存储器（DRAM）、同步动态随机存取存储器（synchronous DRAM，SDRAM）、双倍数据速率同步动态随机存取存储器（double data rate SDRAM，DDR SDRAM）、

增强型同步动态随机存取存储器 (enhanced SDRAM, ESDRAM)、同步连接动态随机存取存储器 (synchlink DRAM, SLDRAM) 和直接内存总线随机存取存储器 (direct rambus RAM, DR RAM)。

该通信接口 903 用于与调度器 900 连接的其它设备进行通信。该总线 904 除包括数据总线之外，还可以包括电源总线、控制总线和状态信号总线等。但是为了清楚说明起见，在图中将各种总线都标为总线 904。

应理解，根据本申请实施例的调度器 900 可对应于本申请实施例中的调度器 700，并可以对应于执行根据本申请实施例中图 3 以及图 6A 所示方法中的调度器 101 所执行的方法，调度器 900 所实现的上述和其它操作和/或功能分别为了实现图 3 以及图 6A 中的相应方法的流程，为了简洁，在此不再赘述。

本申请实施例还提供了一种计算机可读存储介质。所述计算机可读存储介质可以是计算设备能够存储的任何可用介质或者是包含一个或多个可用介质的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 (例如固态硬盘) 等。该计算机可读存储介质包括指令，所述指令指示计算设备执行上述资源调度方法。

本申请实施例还提供了一种计算机程序产品。所述计算机程序产品包括一个或多个计算机指令。在计算设备上加载和执行所述计算机指令时，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、计算机或数据中心通过有线 (例如同轴电缆、光纤、数字用户线 (DSL)) 或无线 (例如红外、无线、微波等) 方式向另一个网站站点、计算机或数据中心进行传输。

所述计算机程序产品可以作为一个软件安装包，在需要使用前述资源调度方法的任一方法的情况下，可以下载该计算机程序产品并在计算设备上执行该计算机程序产品。

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 (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。

以上所述，仅为本申请的具体实施方式，但本申请的保护范围并不局限于此，任何熟悉本技术领域的技术人员在本申请揭露的技术范围内，可轻易想到各种等效的修改或替换，这些修改或替换都应涵盖在本申请的保护范围之内。因此，本申请的保护范围应以权利要求的保护范围为准。

权 利 要 求 书

1. 一种资源调度方法，其特征在于，所述方法包括：
获取待处理的第一作业；
- 5 根据算力拓扑，从资源池中确定多个第一处理器，所述资源池包括多个处理器，所述算力拓扑用于指示所述多个处理器的算力，所述多个第一处理器的算力之间的差值不超过阈值；
为所述第一作业调度所述多个第一处理器，所述第一作业基于所述多个第一处理器并行执行。
2. 根据权利要求1所述的方法，其特征在于，所述根据算力拓扑，从资源池中确定多个第一处理器，包括：
- 10 根据所述算力拓扑确定多个候选集合，不同候选集合包括的处理器数量相同；
确定所述多个候选集合中每个候选集合对应的算力拓扑距离，所述算力拓扑距离用于指示候选集合包括的不同处理器的算力之间的最大差值；
从所述多个候选集合中选择第一候选集合，所述第一候选集合包括所述多个第一处理器，所述多个候选集合中所述第一候选集合的算力拓扑距离最小。
- 15 3. 根据权利要求2所述的方法，其特征在于，所述多个第一处理器位于同一计算单元。
4. 根据权利要求1至3任一项所述的方法，其特征在于，所述方法还包括：
获取待处理的第二作业；
当所述资源池中的多个处理器满足资源调整条件，根据所述算力拓扑将调度给所述第一作业的多个第一处理器中的第一目标处理器调整为所述资源池中的第二目标处理器；
- 20 将所述资源池剩余可用的多个第二处理器调度给所述第二作业，所述多个第二处理器包括所述第一目标处理器。
5. 根据权利要求4所述的方法，其特征在于，所述当所述资源池中的多个处理器满足资源调整条件，根据所述算力拓扑将调度给所述第一作业的多个第一处理器中的第一目标处理器调整为所述资源池中的第二目标处理器，包括：
- 25 当所述资源池中的多个处理器满足资源调整条件，且所述第一作业在所述第一目标处理器上的执行时长小于预设时长，根据所述算力拓扑将调度给所述第一作业的多个第一处理器中的第一目标处理器调整为所述资源池中的第二目标处理器。
6. 根据权利要求1至5任一项所述的方法，其特征在于，所述获取待处理的第一作业，包括：
获取目标队列中的所述第一作业，用于处理所述目标队列中的作业的处理器基于算力拓扑进行确定；
- 30 或者，响应于用户针对所述第一作业的使能操作，获取所述第一作业。
7. 根据权利要求1至6任一项所述的方法，其特征在于，所述方法还包括：
获取测试作业；
根据所述测试作业，对所述多个处理器分别进行算力测试，得到算力测试结果；
根据所述多个处理器分别对应的算力测试结果，生成所述算力拓扑。
- 35 8. 根据权利要求1至7任一项所述的方法，其特征在于，所述第一处理器的算力基于所述第一处理器执行所述测试作业的耗时进行确定。
9. 一种作业处理方法，其特征在于，所述方法包括：
获取待处理的第一作业；
根据算力拓扑，从资源池中为所述第一作业调度多个第一处理器，所述资源池包括多个处理器，所述算力拓扑用于指示所述多个处理器的算力，所述多个第一处理器的算力之间的差值不超过阈值；
利用所述多个第一处理器并行执行所述第一作业。
- 40 10. 一种调度器，其特征在于，所述调度器包括：
获取模块，用于获取待处理的第一作业；
确定模块，用于根据算力拓扑，从资源池中确定多个第一处理器，所述资源池包括多个处理器，所述算力拓扑用于指示所述多个处理器的算力，所述多个第一处理器的算力之间的差值不超过阈值；
- 45

- 调度模块，用于为所述第一作业调度所述多个第一处理器，所述第一作业基于所述多个第一处理器并行执行。
11. 一种作业处理设备，其特征在于，所述作业处理设备包括：
- 获取模块，用于获取待处理的第一作业；
- 5 调度模块，用于根据算力拓扑，从资源池中为所述第一作业调度多个第一处理器，所述资源池包括多个处理器，所述算力拓扑用于指示所述多个处理器的算力，所述多个第一处理器的算力之间的差值不超过阈值；
- 执行模块，用于利用所述多个第一处理器并行执行所述第一作业。
12. 一种调度器，其特征在于，包括处理器、存储器；
- 10 所述处理器用于执行所述存储器中存储的指令，以使所述调度器执行如权利要求 1 至 8 任一项所述方法的步骤。
13. 一种作业处理系统，其特征在于，包括权利要求 12 所述的调度器、以及资源池，所述资源池包括多个处理器。
14. 一种计算机可读存储介质，其特征在于，包括指令，当其在计算设备上运行时，使得所述计算
- 15 设备执行如权利要求 1 至 9 中任一项所述方法的步骤。