TASK SCHEDULING METHOD AND SYSTEM, AND STORAGE MEDIUM

说　明　书

一种任务调度方法、系统及存储介质

技术领域

本说明书涉及资源分配领域，尤其涉及一种任务调度方法、系统及存储介质。

背景技术

随着互联网技术的发展以及数据规模的不断扩大，对于任务处理的要求也不断提高。为了提升数据处理的效率，将计算任务分配到多台物理机上进行并行处理的方式应运而生。当多个相关任务之间存在上下游的依赖关系时，如何合理分配算力资源，提高任务的整体执行效率显得尤为重要。相关技术中采取随机分配的方式将任务分配到相应的物理机上，这样会忽略依赖任务之间的关联关系，往往会增加不必要的耗时。

因此，需要提供一种新的任务调度的方法，能够基于上下游任务之间的依赖关系进行任务分配，降低整体的任务耗时。

背景技术部分的内容仅仅是申请人个人所知晓的信息，并不代表上述信息在本公开申请日之前已经进入公共领域，也不代表其可以成为本公开的现有技术。

发明内容

第一方面，本说明书提供一种任务调度方法，包括获得目标有向图，所述目标有向图是基于 N 个任务间的依赖关系形成的有向无环图，所述 N 为大于 1 的整数；基于所述依赖关系，按照可执行时刻从早到晚的顺序，依次将所述 N 个任务分配到 M 个机器上，以生成调度表，所述 M 为大于 1 的整数，在每次任务分配中，将被分配的任务分配给预计最早能够执行完任务的机器，其中，所述可执行时刻是基于每个任务所依赖的上游任务全部完成的时刻以及所述 M 个机器的预计空闲时刻确定的；以及输出所述调度表。

第二方面，本说明书还提供一种任务调度系统，包括至少一个存储介质，存储有至少一个指令集，用于进行任务调度；以及至少一个处理器，同所述至少一个存储介质通信连接，其中，所述至少一个处理器运行时读取所述至少一个指令集，以根据所述至少一个指令集的指示：获得目标有向图，所述目标有向图是基于 N 个任务间的依赖关系形成的有向无环图，所述 N 为大于 1 的整数；基于所述依赖关系，按照可执行时刻从早到晚的顺序，依次将所述 N 个任务分配到 M 个机器上，以生成调度表，所述 M 为大于 1 的整数，在每次任务分配中，将被分配的任务分配给预计最早能够执行完任务的机器，其中，所述可执行时刻是基于每个任务所依赖的上游任务全部完成的时刻以及所述 M 个机器的预计空闲时刻确定的；以及输出所述调度表。

第三方面，本说明书还提供一种计算机可读非易失性存储介质，其中，所述计算机可读非易失性存储介质中存储有至少一个指令集，所述至少一个指令集被至少一个处理器执行时，实现如第一方面中任一项所述的任务调度方法。

由以上技术方案可知，本说明书提供的任务调度方法、系统及存储介质，在对目标有向图中的多个任务进行调度时，基于任务之间的依赖关系，按照可执行时刻从早到晚的顺序依次对多个任务进行任务分配。在每一次任务分配中，将被分配的任务分配给预计最早能够执行完任务的机器。因此，本说明书提供的方法、系统和存储介质能够利用依赖任务之间的关联关系优化分配逻辑，整体上缩短多任务的完成时长。

本说明书提供的任务调度方法、系统及存储介质的其他功能将在以下说明中部分列出。根据描述，以下数字和示例介绍的内容将对那些本领域的普通技术人员显而易见。本说明书提供的任务调度方法、系统及存储介质的创造性方面可以通过实践或使用下面详细示例中所提供的方法、装置和组合得到充分解释。

附图说明

为了更清楚地说明本说明书实施例中的技术方案，下面将对实施例描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本说明书的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

图 1 示出了根据本说明书的一些实施例提供的任务调度系统的应用场景示意图；

图 2 示出了根据本说明书的一些实施例提供的计算设备的硬件结构示意图；

图 3 示出了根据本说明书的一些实施例提供的一种任务调度方法的流程示意图；

图 4 示出了根据本说明书的一些实施例提供的每次任务分配的流程示意图；

图 5 示出了根据本说明书的一些实施例提供的确定目标任务的示例图；

图 6 示出了根据本说明书的一些实施例提供的确定预计完成时刻的示例图；以及

图 7 示出了根据本说明书的一些实施例提供的任务调度示例图。

具体实施方式

以下描述提供了本说明书的特定应用场景和要求，目的是使本领域技术人员能够制造和使用本说明书中的内容。对于本领域技术人员来说，对所公开的实施例的各种局部修改是显而易见的，并且在不脱离本说明书的精神和范围的情况下，可以将这里定义的一般原理应用于其他实施例和应用。因此，本说明书不限于所示的实施例，而是与权利要求一致的最宽范围。

这里使用的术语仅用于描述特定示例实施例的目的，而不是限制性的。比如，除非上下文另有明确说明，这里所使用的，单数形式“一”，“一个”和“该”也可以包括复数形式。当在本说明书中使用时，术语“包括”、“包含”和/或“含有”意思是指所关联的整数，步骤、操作、元素和/或组件存在，但不排除一个或多个其他特征、整数、步骤、操作、元素、组件和/或组的存在或在该系统/方法中可以添加其他特征、整数、步骤、操作、元素、组件和/或组。

考虑到以下描述，本说明书的这些特征和其他特征、以及结构的相关元件的操作和功能、以及部件的组合和制造的经济性可以得到明显提高。参考附图，所有这些形成本说明书的一部分。然而，应该清楚地理解，附图仅用于说明和描述的目的，并不旨在限制本说明书的范围。还应理解，附图未按比例绘制。

本说明书中使用的流程图示出了根据本说明书中的一些实施例的系统实现的操作。应该清楚地理解，流程图的操作可以不按顺序实现。相反，操作可以以反转顺序或同时实现。此外，可以向流程图添加一个或多个其他操作。可以从流程图中移除一个或多个操作。

在本说明书中，“X 包括 A、B 或 C 中的至少一个”所表达的意思是 X 至少包括 A，或 X 至少包括 B，或 X 至少包括 C。也就是说，X 可以只包括 A、B、C 的任意一个，或者同时包括 A、B、C 的任意组合以及其他可能的内容/元素。所述 A、B、C 的任意组合可以是 A、B、C、AB、AC、BC、或 ABC。

在本说明书中，除非明确说明，否则结构之间产生的关联关系可以是直接的关联关

系也可以是间接的关联关系。比如，当描述“A 与 B 连接”时，除非明确说明了 A 与 B 直接连接，否则应当理解成 A 可以与 B 直接连接，也可以与 B 间接地连接；再比如，当描述“A 在 B 之上”时，除非明确说明了 A 直接在 B 的上方（AB 相邻且 A 在 B 的上方），否则应当理解成 A 可以直接在 B 的上方，A 也可以间接地在 B 之上（AB 之间隔着其他元素，且 A 在 B 的上方）。以此类推。

为了方便描述，首先对说明书中出现的术语进行如下解释：

有向无环图（Directed Acyclic Graph，DAG）：指的是一个由顶点和边组成的有向图，其中顶点表示任务或事件，边表示任务或事件之间的依赖关系。有向无环图中的图是有向的，表示任务或事件之间的执行顺序，而无环表示不存在循环依赖关系，保证了任务或事件的有序性。

任务调度：对多个任务按照一定的规则和算法进行排队、分配和执行的过程。任务调度的目标是合理地利用系统资源，提高系统的效率和响应性，同时满足任务的各项要求和约束。

隐私计算（privacy-preserving computation）：是指在保护数据本身不对外泄露的前提下实现数据分析计算的技术集合，达到对数据可用、不可见的目的。在充分保护数据隐私安全的前提下，实现数据价值的转化和释放。由此可见，隐私计算是一种保护数据（隐私数据/敏感数据）的计算方法，通过对数据进行加密处理，使得在计算过程中无法直接访问和获取数据，从而保护用户的隐私。

本说明书提供一种任务调度方法及任务调度系统。所述任务调度系统可以执行上述任务调度方法，从而实现高效合理的任务调度安排。比如，在多任务的应用场景中，所述任务调度系统可以执行上述任务调度方法，从而可以使用多个机器进行任务的并行处理。

本说明书提供的任务调度方法可以应用于多种架构的系统。比如，上述多个机器可以是分布式系统、微服务系统、或者云架构系统等系统中的计算节点机器。比如，上述多个机器可以是部署在同一计算机集群中的机器，也可以是部署在不同机构的计算机集群中的机器。在一些实施例中，上述多个机器之间可以进行通信，以实现数据传输，比如实现中间计算数据或者计算结果数据的传输，从而使得具有依赖关系的任务可以执行。上述多个机器之间的通信可以通过不同的方式实现。比如，上述多个机器之间可以直接通信连接，或者间接通信连接。直接通信连接可以是上述多个机器之间直接进行数据传输，无需经过中转。间接通信连接可以是上述多个机器与中心平台通信连接，通过中心

平台实现多个机器之间的通信。在间接通信连接中，上述多个机器可以将中间文件或者结果文件上传至中心平台的存储空间。存储空间可以是云存储服务器，可以使用文件存储系统或者是对象存储系统。

本说明书提供的任务调度方法可以适用于多种任务调度应用场景。比如，在一些大数据分析的任务中，可以将不同数据库中的数据的计算任务分配到不同的机器上进行并行计算，以提高计算效率。比如，在物联网系统中，存在大量的机器人、智能设备等，通过任务调度系统进行合理安排，可以将原本分散的任务进行集中统一的调度，提高生产、服务效率。再比如，在隐私计算中，各个机构可以将各自经过加密的文件上传至云存储平台，再由计算平台进行统一的计算资源调度，既可以保证数据的安全，也充分利用了边缘端的算力资源。

下面以隐私计算平台的应用场景为例进行具体说明。在隐私计算平台中，隐私计算任务往往需要多个机构共同参与、协同计算，并获得最终的计算结果。而不同机构之间的原始数据属于隐私数据。隐私数据通常不允许在不同机构之间传输。比如，在大模型的训练过程中，为了使得模型具有更好的适应性以及通用性，通常需要各个机构同时参与训练。此时，各个机构各自持有部分训练数据，且训练数据不允许在不同机构之间传输。在一些实施例中，各个机构可以将各自的训练数据经过清洗、对齐、脱敏、加密之后上传至云端服务器，以供后续模型训练时使用。在一些实施例中，各个机构也可以在各自的机器上使用自己的训练数据进行模型的训练，并将各自训练得到的模型参数上传至云端服务器，以供后续模型训练时使用。比如，在信息推荐模型的训练中需要使用用户画像数据，用户画像数据涉及多个机构的用户数据，用户数据不允许泄露。各个机构可以将各自的用户数据经过脱敏、加密之后上传各自云存储服务器，以供后续模型训练时使用。各个机构也可以基于各自的用户数据进行信息推荐模型的训练，将各自训练得到的模型参数上传各自云存储服务器，以供后续模型训练时使用。再比如，金融机构的信用风险评估模型的训练中需要使用多方的合规数据。合规数据不允许泄露。各个机构可以将各自的合规数据经过脱敏、加密之后上传各自云存储服务器，从而使金融机构可以和优质的数据机构打通，以供后续模型训练时使用。各个机构也可以基于各自的合规数据进行信用风险评估模型的训练，将各自训练得到的模型参数上传各自云存储服务器，以供后续模型训练时使用。

大模型的训练可以离线训练，也可以在线训练。模型训练任务的有向图将整个训练的大任务细化成多个存在依赖关系的小任务。在此基础上，计算平台通过本说明书提供

的任务调度方法，将多个小任务分配到多个机构的机器上，以充分利用各个机构的算力资源。

需要说明的是，上述有关隐私计算平台的应用场景只是本说明书提供的多种应用场景中的一个。本领域技术人员应当明白，本说明书所述的任务调度方法及系统应用于其它使用场景也在本说明书的保护范围内。

图 1 示出了根据本说明书的一些实施例提供的任务调度系统 003 的应用场景示意图。为了方便理解，后文描述中以隐私计算场景为例进行举例说明。如图 1 所示，应用场景中可以包括计算平台 001、计算节点 002 以及任务调度系统 003。任务调度系统 003 可以用于对计算平台 001 中的多个任务进行调度计算，从而将多个任务安排至计算节点 002 中的多个机器上。

计算节点 002 可以是机构部署的计算资源及配套设施。不同机构可以部署不同的计算节点 002。图 1 所示的示意图中，一个计算机点 002 可以对应一个机构。以隐私计算为例，在不同的应用场景中可以有不同类型的机构。比如，在信息推荐的模型训练中，所述机构可以包括有信息推荐需求的商家、媒体平台以及第三方的数据管理平台。媒体平台可以提供丰富的用户画像数据。多家机构可以通过用户数据匹配，筛选出共有用户的数据作为训练数据，以供后续的模型训练使用。再比如，在金融机构的信用风险评估模型的训练中，所述机构可以包括金融机构以及第三方数据源机构。第三方数据源机构可以有一个或者多个，用于提供优质合规的用户数据。结合多个数据源机构的数据可以为风险评估模型提供更加全面的训练数据。

机构可以通过私有化部署、集群化部署、或者 SaaS（Software as a Service，软件即服务）免部署等多种方式将计算节点 002 接入计算平台 001。计算平台 001 可以对计算节点 002 中的计算设备进行调度。为方便描述，我们将多个计算节点 002 的数量定义为 M。M 为大于 1 的整数。每个计算节点 002 可以包括至少一台计算设备。其中，每台计算设备可以为服务器或者客户端。每个计算节点 002 的多台计算设备可以形成计算机集群。为了方便描述，我们将每个计算节点 002 对应的计算机集群定义为一台机器。M 个计算节点 002 对应 M 台机器。计算节点 002 可以采用任意形式的系统架构。比如，计算节点 002 可以采用云原生架构、分布式架构、集中式架构、或者微服务架构等系统架构。

计算节点 002 可以部署云存储空间。比如，计算节点 002 可以通过云存储服务的厂商使用其提供的云存储空间租借、云存储数据管理等服务，将数据存储在云服务厂商提

供的云存储服务器中。比如，计算节点 002 可以通过自建存储，将数据存储在节点的云存储服务器中。存储空间可以使用文件存储系统或者是对象存储（Object Storage Service, OSS）系统，本说明书对于存储空间的系统架构不作限定。在隐私计算中，不同的计算节点 002 可以通过计算平台 001 获取中间结果。计算节点 002 不允许将隐私数据存储至计算平台 001 的云存储服务器中。因此，计算节点 002 无法直接获得其它机构的隐私数据。

计算平台 001 可以与计算节点 002 通信连接，并向计算节点 002 提供各种服务。计算平台 001 可以解决多个计算节点 002 之间的协作问题。以隐私计算为例，计算平台 001 可以提供标准化的隐私计算产品，不同的机构可以根据自身的使用需求和系统架构选择合适的接入方式。计算平台 001 可以基于隐私计算技术，实现数据可用不可见，解决企业之间数据协同计算的安全性和隐私保护问题，实现跨机构的任务合作。计算平台 001 可以包括至少一台计算设备。所述至少一台计算设备可以是一台服务器，或者是由多台服务器组成的服务器集群。计算平台 001 可以是任意架构的系统。比如，计算平台 001 可以采用云原生架构、分布式架构、集中式架构、或者微服务架构等系统架构。

在需要多方机构共同参与的隐私计算（比如模型训练）中，计算平台 001 可以调用 M 个计算节点 002 对应的 M 台机器共同执行计算任务，以充分利用各个机构的算力资源，同时提高计算任务的执行效率。任务调度系统 003 可以对计算平台 001 中的计算任务进行任务调度，从而将子任务分配至 M 个计算节点 002 中的 M 台机器上执行。需要注意的是，不同的子任务之间可能存在依赖关系，即下游子任务的执行依赖上游子任务的输出结果。不同的计算节点 002 的机器之间的数据传输需要通过云存储服务器实现。也就是说，当下游子任务与其依赖的上游子任务不在同一台机器上执行时，上游子任务需要将执行结果上传至云存储服务器，下游子任务需要从云存储服务器上下载上游子任务的执行结果才可以执行。当下游子任务与其依赖的上游子任务在同一台机器上执行时，下游子任务可以直接从执行机器的本地存储器中获取上游子任务的执行结果。对于相同大小的数据，直接从本地存储器获取所需的时长小于从云存储服务器上下载的时长。

任务调度系统 003 可以基于有向图进行任务调度。计算平台 001 可以将计算任务划分为 N 个任务，并基于 N 个任务之间的依赖关系生成目标有向图 100。任务调度系统 003 可以基于目标有向图 100 中的任务依赖关系，按照可执行时刻从早到晚的顺序，依次将 N 个任务分配到 M 个机器上，并在每次任务分配中将任务分配至最早能够执行完任务的机器上，从而生成调度表。本说明书提供的任务调度系统 003，在进行任务调度

的过程中充分利用任务间的依赖关系，避免相关联的任务被随机分配从而产生不必要的通信耗时，进而可以降低任务的整体耗时。

任务调度系统 003 可以包括至少一台计算设备。所述至少一台计算设备运行时可以执行本说明书提供的任务调度方法。在一些实施例中，任务调度系统 003 与计算平台 001 可以是相互独立并进行通讯连接的设备。在一些实施例中，任务调度系统 003 也可以是计算平台 001。

图 2 示出了根据本说明书的一些实施例提供的计算设备 200 的硬件结构示意图。在一些实施例中，计算设备 200 可以是任务调度系统 003 中的计算设备。如图 2 所示，计算设备 200 包括至少一个存储介质 230 和至少一个处理器 220。在一些实施例中，计算设备 200 还可以包括内部通信总线 210。在一些实施例中，计算设备 200 还可以包括通信端口 250。在一些实施例中，计算设备 200 还可以包括 I/O 组件 260。

内部通信总线 210 可以连接不同的系统组件，包括存储介质 230 和处理器 220。I/O 组件 260 支持计算设备 200 和其他组件之间的输入/输出。

存储介质 230 可以包括数据存储装置。数据存储装置可以是非暂时性存储介质，也可以是暂时性存储介质。比如，数据存储装置可以包括磁盘 232、只读存储介质（ROM）234 或随机存取存储介质（RAM）236 中的一种或多种。存储介质 230 还包括存储在数据存储装置中的至少一个指令集。指令集是计算机程序代码，计算机程序代码可以包括执行本说明书提供的任务调度方法的程序、例程、对象、组件、数据结构、过程、模块等等。

通信端口 250 用于计算设备 200 同外界的数据通讯。比如，计算设备 200 可以通过通信端口 250 连接网络 130。至少一个处理器 220 同至少一个存储介质 230 通过内部通信总线 210 通信连接。至少一个处理器 220 用以执行上述至少一个指令集。任务调度系统 001 运行时，至少一个处理器 220 读取至少一个指令集，并且根据至少一个指令集的指示执行本说明书提供的任务调度方法。

处理器 220 可以执行任务调度方法包含的所有步骤。处理器 220 可以是一个或多个处理器的形式。处理器 220 可以发出执行指令。处理器 220 可以包括一个或多个硬件处理器，例如微控制器，微处理器，精简指令集计算机（RISC），专用集成电路（ASIC），特定于应用的指令集处理器（ASIP），中央处理单元（CPU），图形处理单元（GPU），物理处理单元（PPU），微控制器单元，数字信号处理器（DSP），现场可编程门阵列（FPGA），高级 RISC 机器（ARM），可编程逻辑器件（PLD），能够执行一个或多个功能的任何电

路或处理器等，或其任何组合。

仅仅为了说明问题，在本说明书中计算设备 200 中仅描述了一个处理器 220。然而，应当注意，本说明书中计算设备 200 还可以包括多个处理器，因此，本说明书中披露的操作和/或方法步骤可以如本说明书所述的由一个处理器执行，也可以由多个处理器联合执行。例如，如果在本说明书中计算设备 200 的处理器 220 执行步骤 A 和步骤 B，则应该理解，步骤 A 和步骤 B 也可以由两个不同处理器 220 联合或分开执行（例如，第一处理器执行步骤 A，第二处理器执行步骤 B，或者第一和第二处理器共同执行步骤 A 和 B）。

图 3 示出了根据本说明书的一些实施例提供的一种任务调度方法 P100 的流程示意图。如前所述，任务调度系统 003 的至少一台计算设备 200 可以执行任务调度方法 P100。具体地，存储介质 230 可以存储有至少一组指令集用于任务调度。处理器 220 可以与存储介质 230 通信连接。处理器 220 可以读取存储在其本地存储介质和/或数据库中的指令集，然后根据指令集的规定，执行任务调度方法 P100。

如图 3 所示，任务调度方法 P100 可以包括：

S110：获得目标有向图 100。

所述目标有向图 100 是基于 N 个任务间的依赖关系形成的有向无环图，所述 N 为大于 1 的整数。如前所述，有向无环图可以通过顶点和边表征多个任务之间的依赖关系。每个顶点表示一个任务，其中至少一个顶点为起始顶点用于表示起始任务。每条边表示两个任务之间的依赖关系。其中，边是带方向的边，即由上游任务指向下游任务，表示下游任务的执行依赖上游任务的执行结果。需要注意的是，起始顶点表示的起始任务不存在上游任务。有向无环图中不存在环，即不存在循环依赖，保证了任务的有序执行。

目标有向图 100 可以是基于 N 个任务间的依赖关系形成的有向无环图。在一些实施例中，N 个任务可以是一个任务分解成的多个子任务，N 为大于 1 的整数。比如，N 个任务可以是一个大模型的离线训练任务分解成的多个子任务。其中，多个子任务可以包括数据增广等预处理子任务。多个子任务还可以包括对数据集进行划分，可以将数据集划分为训练集和验证集。对于规模较大的训练集，可以将其划分成多个部分，从而使用多台机器执行训练子任务。多个子任务还可以包括使用指定的训练集对模型进行训练。多个子任务还可以包括对多个训练子任务的结果进行整合。多个子任务还可以包括对训练得到的大模型在验证集上进行验证。

在一些实施例中，所述依赖关系可以包括下游任务的执行依赖上游任务的输出结果。

继续以上述大模型的离线训练任务为例，比如，模型训练的子任务依赖数据预处理和/或数据集划分子任务的执行结果。数据预处理子任务的执行结果可以是输出经过数据增广等处理之后的数据集。数据集划分子任务的执行结果可以是输出划分之后的一个或者多个训练集。比如，训练结果整合的子任务依赖多个训练子任务的训练结果。训练子任务的执行结果可以是输出经过对应的训练集训练得到的训练模型。再比如，大模型验证的子任务依赖训练结果整合的子任务。训练结果整合的子任务将多个训练子任务得到的训练模型进行整合输出此次离线训练任务得到的大模型。每个子任务的输出结果既可以保存在执行子任务的本地机器中，也可以上传至云存储服务器中。当下游任务与其上游任务不在同一台机器上执行时，下游任务可以从云存储服务器中下载所需的依赖任务的输出数据。当下游任务与其上游任务在同一台机器上执行时，下游任务可以直接从执行子任务的本地机器的存储器中下载所需的依赖任务的输出数据。

在一些实施例中，所述 N 个任务可以包括至少一个起始任务。每个起始任务没有依赖的上游任务。对于起始任务来说，其执行不依赖上游任务的输出结果。起始任务执行时所需的数据可以从指定的存储空间中获取。比如，当起始任务是一个训练子任务时，需要使用到训练数据以及预训练模型，此时可以从云存储服务器中获取所需的数据及模型。

目标有向图 100 可以使用任意形式的数据结构表示。比如，目标有向图 100 可以是邻接矩阵。比如，目标有向图 100 还可以是邻接链表。本说明书对目标有向图 100 的数据结构不作限定。在一些实施例中，目标有向图 100 可以由计算平台 001 生成，并预先存储在计算平台 001 中，在任务调度系统 003 执行所述任务调度方法 P100 时直接使用目标有向图 100。在一些实施例中，目标有向图 100 也可以由其它系统生成，在任务调度系统 003 执行所述任务调度方法 P100 时从外部进行调用。

如图 3 所示，任务调度方法 P100 还可以包括：

S130：基于所述依赖关系，按照可执行时刻从早到晚的顺序，依次将所述 N 个任务分配到 M 个机器上，以生成调度表。

其中，所述可执行时刻是基于每个任务所依赖的上游任务全部完成的时刻以及所述 M 个机器的预计空闲时刻确定的。在每次任务分配中，将被分配的任务分配给预计最早能够执行完任务的机器，所述 M 为大于 1 的整数。

在对步骤 S130 进行具体介绍之前，先厘清两个概念。在本说明书中，使用“时刻”表示瞬时的时间点，使用“时长”表示两个时刻之间的时间长度。

任务调度方法 P100 在确定每个任务的可执行时刻时，综合考虑当前任务的所有上游任务全部完成时刻，以及 M 个计算节点 002 对应的 M 个机器的预计空闲时刻。在本说明书中，以可被分配的机器数量为 M 进行举例说明，其中 M 为大于 1 的整数。如前文所述，依赖关系可以指的是下游任务的执行依赖上游任务的输出结果。因此，对于每个任务来说，在当前任务的依赖的上游任务没有全部完成之前，当前任务是无法开始执行的。需要特别说明的是，由于起始任务没有上游任务，可以认为起始任务的上游任务数量为零。在一个目标有向图 100 中可以存在一个或多个起始任务。确定每个任务可被执行的时刻，除了要考虑当前任务的全部上游任务的完成时刻，还需要考虑可被分配的 M 个机器的预计空闲时刻。换句话说，即使当前任务的全部上游任务都已执行完毕，但是所有上游依赖任务完成的时刻没有机器空闲，则依旧无法执行当前任务。同样的，即使有机器存在空闲，但是当前任务的全部上游任务还未执行完毕，则也依旧无法执行当前任务。

任务调度方法 P100 按照任务可执行时刻从早到晚的顺序进行任务的分配，在每一轮任务分配时，选择最早可执行的任务进行分配。任务可执行时刻并不是在初始确定后一成不变的，而是可以根据每一轮的任务分配情况发生变化。

图 4 示出了根据本说明书的一些实施例提供的每次任务分配的流程示意图。如图 4 所示，步骤 S130 可以包括：

S131：从待分配任务中确定最早可执行的目标任务。

所述待分配任务是在前轮任务分配之后目标有向图 100 中还未被分配的任务。对于第一轮任务分配来说，不存在前轮任务分配，待分配任务可以包括所有 N 个任务。所述目标任务是所述待分配任务中最早可执行的任务。最早可执行的任务是指目标任务的可执行时刻是所有待分配任务中最早的。在每次任务分配中，确定当轮任务分配中的目标任务，并将所述目标任务分配到相应的机器上。

步骤 S131 可以包括：从所述待分配任务中确定可执行的至少一个候选任务；以及从所述至少一个候选任务中确定所述目标任务。

其中，每个候选任务所依赖的全部上游任务都已被分配完毕。在每一次分配时，所述待分配任务中的每个任务可以按照依赖任务是否全部被分配完毕分为候选任务或者非候选任务。其中，候选任务的全部上游任务都已被分配完毕。对于起始任务而言，由于其不存在上游依赖任务，默认起始任务的上游任务已全部分配完毕。每次分配时，从待分配任务中确定至少一个候选任务。

图 5 示出了根据本说明书的一些实施例提供的确定目标任务的示例图。图 5 中目标有向图 100 中包括 8 个任务，即任务数量 N 等于 8。箭头从上游任务指向下游任务，表示任务之间的依赖关系。虚线箭头表示箭头起始端的上游任务已被分配。图 5 所示为经历过第一轮分配过后，任务 E 已被分配的情况。在第二轮分配中，除了已被分配的任务 E，其余的所有任务都为待分配任务。其中，任务 C 的全部上游任务 E 已被分配，任务 A 和任务 B 不存在上游任务。因此，确定任务 C、任务 A 和任务 B 为第二轮分配中的候选任务。接下来，从至少一个候选任务中确定一个目标任务。

如前所述，目标任务是待分配任务中最早可执行的任务。同样，目标任务也是至少一个候选任务中最早可执行的任务。在一些实施例中，当候选任务数量为 1 时，该候选任务即为目标任务。当候选任务数量大于 1 时，为了确定目标任务，需要确定每个候选任务的可执行时刻。在一些实施例中，所述从所述至少一个候选任务中确定所述目标任务可以包括：确定所述 M 个机器在前轮任务分配之后的最早空闲时刻，所述最早空闲时刻为所述 M 个机器各自执行完已分配任务的 M 个预计空闲时刻中最早的时刻；确定所述每个候选任务所依赖的上游任务全部完成的时刻；对所述每个候选任务，将所述最早空闲时刻与当前候选任务所依赖的上游任务全部完成的时刻中的较大者确定为所述当前候选任务的可执行时刻；以及从所述至少一个候选任务中确定所述可执行时刻最早的时刻对应的候选任务为所述目标任务。也就是说，对于每个候选任务来说，可执行时刻是在当前候选任务所依赖的上游任务全部完成且有机器存在空闲的时刻中最早的时刻。在一些实施例中，当所述可执行时刻最早的时刻对应的候选任务不止一个时，可以确定其中任意一个任务作为目标任务。

如前所述，每个任务的可执行时刻是由上游依赖任务的执行完成时刻和 M 个机器的预计空闲时刻共同决定的。继续以图 5 为例，t0 表示任务调度的起始时刻，假设在 t0 时刻所有可被分配的机器均处于空闲状态。图 5 中以可被分配的机器数量为 3，即 M 等于 3 为例进行说明。在进行第二轮任务分配时可以确定，经过前轮的任务分配之后，机器 1 执行完任务 E 后的预计空闲时刻为 t1，机器 2 和机器 3 未被分配任务，预计空闲时刻都为 t0。从 3 个机器的 3 个预计空闲时刻中确定最早的预计空闲时刻 t0 为最早空闲时刻。

对于本轮的候选任务——任务 C、任务 A 和任务 B 而言，任务 C 的上游任务全部完成时刻为 t1，任务 A 和任务 B 不存在上游任务，默认上游任务全部完成时刻为 t0。对于每个候选任务，将所述最早空闲时刻与当前候选任务所依赖的上游任务全部完成的

时刻中的较大者确定为所述当前候选任务的可执行时刻。据此可以确定，任务 C 的可执行时刻为 t0 与 t1 中的较大者 t1，任务 A 和任务 B 的可执行时刻为 t0 与 t0 中的较大者 t0。任务 C、任务 A、任务 B 的可执行时刻 t1、t0、t0 中最早的可执行时刻为 t0。最早可执行时刻 t0 对应的候选任务为任务 A 和任务 B，可以选择其中任一任务作为目标任务。图 5 中选择任务 A 作为本轮任务分配的目标任务。

任务调度方法 P100 根据任务的可执行时刻从早到晚的顺序对任务进行分配，在每一轮的分配中都选择可执行时刻最早的任务作为目标任务，对所述目标任务进行分配。

如图 4 所示，步骤 S130 还可以包括：

S133：确定所述目标任务在所述 M 个机器上执行时的 M 个预计完成时刻。

目标有向图 100 中的每个任务执行时需要使用上游任务的执行结果。上游任务的执行结果可以存储在执行该上游任务的计算节点 002 的机器上以及云存储空间中。起始任务没有上游任务，其所需的数据存储在云存储空间中。每个任务执行前需要获取所需的数据。当下游任务与所有上游任务在同一台机器上执行时，上游任务的执行结果存储在同一台机器上，不需要从云存储空间中下载。对于有多个上游任务的下游任务来说，需要判断其每个上游任务是否都在当前机器上执行过，如果是在其它机器上执行的需要从云存储空间获取当前上游任务的执行结果。

在一些实施例中，所述确定所述目标任务在所述 M 个机器上执行时的 M 个预计完成时刻可以包括：确定所述目标任务在所述 M 个机器中的每个机器上执行时的预计开始时刻；确定所述目标任务在所述每个机器上执行时所需的预计时长，所述预计时长包括预计执行时长以及预计通信时长，所述预计通信时长包括所述目标任务获取所需数据的时长；以及将所述目标任务在所述每个机器上执行时的所述预计开始时刻与所述预计时长叠加，得到所述目标任务在所述每个机器上执行时的预计完成时刻，从而得到所述 M 个预计完成时刻。

所述目标任务在每台机器上的预计开始时刻指的是当所述目标任务被分配到当前机器上时，最早可以开始执行的时刻。一方面，所述目标任务的执行依赖上游任务的执行结果，因此需要等待全部上游任务执行完毕才可以执行；另一方面，所述目标任务在当前机器上执行时需要等待当前机器执行完所有已被分配的任务，才能够释放算力资源。因此，在确定预计开始时刻时，需要综合考虑全部上游任务的执行完毕时刻以及每台机器的预计空闲时刻。

在一些实施例中，所述确定所述目标任务在所述 M 个机器中的每个机器上执行时

的预计开始时刻可以包括：确定所述每个机器执行完已分配任务的预计空闲时刻；确定所述目标任务所依赖的上游任务全部完成的时刻；以及对所述每个机器，将当前机器的所述预计空闲时刻与所述目标任务所依赖的上游任务全部完成的时刻中的较大者，作为所述目标任务在所述当前机器上执行时的预计开始时刻。

图 6 示出了根据本说明书的一些实施例提供的确定预计完成时刻的示例图。如图 6 所示，在之前轮次的分配中，任务 E、任务 A 和任务 B 已分别被分配到机器 1、机器 2 和机器 3 上。任务 C 为本轮分配中的目标任务。机器 1、机器 2、机器 3 的预计空闲时刻分别是 t1、t2、t3。目标任务 C 依赖的上游任务 A 完成时刻为 t1。如果目标任务 C 在机器 1 上执行，预计开始时刻为机器 1 的预计空闲时刻 t1 和上游任务全部完成时刻 t1 中的较大者，仍然为 t1。如果目标任务 C 在机器 2 上执行，预计开始时刻为机器 2 的预计空闲时刻 t2 和上游任务全部完成时刻 t1 中的较大者 t2。如果目标任务 C 在机器 3 上执行，预计开始时刻为机器 3 的预计空闲时刻 t3 和上游任务全部完成时刻 t1 中的较大者 t3。因此，目标任务 C 在机器 1、机器 2、机器 3 上执行时的预计开始时刻分别是 t1、t2、t3。

所述目标任务在每台机器上执行时的预计时长可以指所述目标任务在当前机器上从任务执行之前的准备工作开始到任务执行结束的预计总耗时。在一些实施例中，所述预计时长包括预计通信时长以及预计执行时长。预计通信时长可以是任务执行之前的准备工作可以包括数据准备工作。即使对于同一个任务来说，在不同的机器上执行时的预计时长也可能是不同的。

在一些实施例中，所述预计通信时长可以包括所述目标任务获取所需数据的时长。所述目标任务所需的数据可以包括上游任务的执行结果。当目标任务与其依赖的一个上游任务不在同一台机器上执行时，这个上游任务执行结果存储在计算平台 001 的云端存储器中，所述目标任务需要从计算平台 001 的云端存储器中获取数据。当目标任务与其依赖的一个上游任务在同一台机器上执行时，这个上游任务执行结果存储在执行所述目标任务的机器上，所述目标任务可以直接从执行的机器上获取执行结果数据而不需要从计算平台 001 的云端存储器中获取数据。直接从本地机器上获取数据的通信时间小于从云端存储器下载数据所花费的时间。当所述目标任务与上游任务在同一台机器上执行时，可以认为所需的预计通信时长为零。在一些实施例中，不同的机器从云端下载相同数据的预计通信时长相同。

以隐私计算中的大模型离线训练为例，大模型的训练中，往往涉及到大规模的模型

参数和训练数据。从云存储平台中下载训练模型和训练数据成为训练过程中耗时的重要方面。如果将下游任务和上游任务安排在同一台机器上执行，则可以大大减少下载数据的耗时。

继续参考图 6，当目标任务 C 与上游任务在同一机器上执行时，预计通信时长为零，当目标任务与上游任务在不同的机器上执行时，从云存储空间中下载上游任务的执行结果的预计通信时长为 T_D。目标任务 C 的上游任务是任务 E，当将目标任务 C 安排在机器 1 上执行时，由于上游任务 E 在机器 1 上执行过，预计通信时长为零。当将目标任务 C 安排在机器 2 上执行时，由于上游任务 E 未在机器 2 上执行过，预计通信时长为 T_D。同样的，当将目标任务 C 安排在机器 3 上执行时，由于上游任务 E 未在机器 3 上执行过，预计通信时长也为 T_D。在图 6 中目标任务 C 只有一个上游任务，当目标任务存在多个上游任务时，需要考虑多个上游任务中每个上游任务的执行情况。

在一些实施例中，在所述目标任务依赖至少一个上游任务时，所述预计通信时长可以包括所述至少一个上游任务对应的至少一个通信时长中最大的时长。当所述目标任务获取所述至少一个上游任务的执行结果时，对于每一个上游任务，如果当前上游任务与执行所述目标任务的机器不是同一台机器，则需要从云存储服务器中下载当前上游任务对应的执行结果数据。如果当前上游任务与执行所述目标任务的机器是同一台机器，则直接从所述同一台机器本地获取执行结果数据。从本地获取数据所需的预计通信时长小于从云存储服务器中下载数据的预计通信时长。也就是说，对所述至少一个上游任务中的每个上游任务，当所述目标任务与当前上游任务在同一个机器上执行时所需的通信时长小于所述目标任务与所述当前上游任务不在同一个机器上执行时所需的通信时长。在一些实施例中，对所述目标任务的每个上游任务，当所述目标任务与所述当前上游任务在同一个机器上执行时所需的通信时长为零或者接近于零。在一些实施例中，当所述目标任务需要获取多个执行结果时，获取数据的操作可以并行执行，预计通信时长为至少一个上游任务对应的至少一个预计通信时长中最大的时长。

所述预计执行时长可以指所述目标任务从开始执行到执行结束的时间长度。在一些实施例中，所述预计执行时长与任务复杂度成正相关，任务越复杂，预计执行时长越长。在一些实施例中，所述预计执行时长与数据规模成正相关，数据规模越大，预计执行时间越长。

在一些实施例中，所述目标任务在不同的机器上执行时的预计执行时长相同。继续参考图 6，目标任务 C 在 3 台机器上的预计执行时间相同，都为 T_E。将所述目标任务在

每个机器上执行时的预计开始时刻与预计时长叠加，得到所述目标任务在每个机器上执行时的预计完成时刻。据此可得，目标任务 C 在机器 1 上执行时的预计完成时刻为预计开始时刻 t1 与预计时长 T_E 相加得到，即 t4。目标任务 C 在机器 2 上执行时的预计完成时刻为预计开始时刻 t2 与预计时长（预计通信时长 T_D+预计执行时长 T_E）相加得到，即 t5。目标任务 C 在机器 3 上执行时的预计完成时刻为预计开始时刻 t3 与预计时长（预计通信时长 T_D+预计执行时长 T_E）相加得到，即 t6。至此，可以确定在本轮任务分配中所述目标任务 C 在机器 1、机器 2、机器 3 上执行时的 3 个预计完成时刻分别为 t4、t5、t6。

在一些实施例中，所述目标任务在不同的机器上执行时的预计执行时长与机器处理速度负相关。不同的机器执行所述目标任务时，由于机器的性能存在差异，所述目标任务在不同的机器上执行时的预计执行时长可以不同。执行相同的任务时，处理速度更快的机器花费的预计执行时长越短，反之，处理速度越慢的机器花费的预计执行时长越长。

如图 4 所示，步骤 S130 还可以包括：

S135：将所述目标任务分配给目标机器，所述目标机器为所述 M 个预计完成时刻中最早的时刻对应的机器。

如前所述，目标任务在 M 个机器上对应的 M 个预计完成时刻可以由所述目标任务在 M 个机器上的预计开始时刻和执行时的预计时长共同确定。在得到 M 个预计完成时刻之后，将所述目标任务分配给所述 M 个预计完成时刻中最早的时刻对应的机器。继续参考图 6，在本轮任务分配中，目标任务 C 在机器 1、机器 2、机器 3 上执行时的 3 个预计完成时刻分别为 t4、t5、t6。3 个预计完成时刻中最早的时刻为 t4，t4 对应的机器 1 即为目标机器。将目标任务 C 分配给目标机器，即机器 1。

综上，步骤 S130 基于目标有向图 100 中 N 个任务间的依赖关系，按照可执行时刻从早到晚的顺序，依次将所述 N 个任务分配到 M 个机器上。在每次任务分配中，从至少一个候选任务中选择一个目标任务，将所述目标任务分配给预计最早能够执行完任务的机器。

如图 3 所示，任务调度方法 P100 还可以包括：

S150：输出所述调度表。

所述调度表可以表示 M 个机器中每个机器以及与每个机器对应的被分配的任务。所述调度表可以是任意格式的数据。所述调度表可以使用数据库进行存储。所述调度表可以采用任意的数据结构。比如，所述调度表可以是字典。比如，所述调度表可以是矩

阵。本说明书中对所述调度表的格式以及数据结构不做限定。

综上所述，本说明书提供的任务调度方法 P100 基于目标有向图 100 中任务间的依赖关系，在进行任务调度时充分考量下游任务获取上游任务的执行结果的通信耗时，在每一轮分配中将最早可执行的任务分配到最早可执行完所述任务的机器上。在一些相关技术中，采用随机分配的方式将任务分配到 M 个机器中的任意一个上，未考虑下游任务与其上游任务如果不在同一机器上执行会产生额外的数据传输耗时。与随机分配的方式相比，任务调度方法 P100 可以减少从云存储器中下载数据的次数，从而整体上减少任务完成的时间。

图 7 示出了根据本说明书的一些实施例提供的任务调度示例图。如图 7 所示，目标有向图 100 中包括 6 个任务，任务之间的依赖关系通过图中的箭头表示，箭头从上游任务指向下游任务。为了方便描述每轮分配是如何进行的，每个任务的右侧标注出预计通信时长以及预计执行时长。“预计通信时长 T_D：x1，x2”表示该下游任务有两个上游依赖任务，当执行该下游任务的机器从云端获取两个上游任务的执行结果数据时的预计通信时长分别是 x1 和 x2。如果下游任务与上游任务在同一台机器上执行过，则下游任务获取该上游任务的执行结果所需的预计通信时长为 0。起始任务不存在上游依赖任务，默认需要从云端获取所需数据。“预计执行时长 T_E：y”表示该任务在每台机器上执行时的预计执行时长都为 y。本示例中以同一个任务在不同的机器上执行时的预计执行时间相同为例进行说明。

下面将对每一轮分配是如何进行的做具体说明。

在第一轮分配中，所有任务都为待分配任务。任务 A 和任务 F 不存在上游任务，是本轮分配中的候选任务。机器 1 和机器 2 的预计空闲时刻都为 0，最早空闲时刻为两个预计空闲时刻中的较小者，即为 0。任务 A 的可执行时刻为全部上游任务执行完毕时刻 0 和机器的最早空闲时刻 0 中的较大者，即为 0。同理，任务 B 的可执行时刻为 0。由于两个候选任务的可执行时刻相同，因此可以从两个任务中选择任何一个任务作为目标任务。示例中选择任务 A 作为目标任务。任务 A 在机器 1 上的预计开始时刻为上游任务执行完毕的时刻 0 与机器 1 的预计空闲时刻 0 中的较大者，即为 0。任务 A 在机器 1 上执行时的预计时长为预计通信时长 4 加上预计执行时长 2，即为 6。任务 A 在机器 1 上执行时的预计完成时刻为预计开始时刻 0 加上预计时长 6，即为 6。同理，任务 A 在机器 2 上的预计开始时刻为 0，预计时长为 6，预计完成时刻为 6。任务 A 在两个机器上的预计完成时刻相同，因此，可以从两个机器中选择任何一台机器作为目标机器执行

任务 A。示例中选择机器 1 作为目标机器。在第一轮分配中，将任务 A 分配到机器 1 上。

在第二轮分配中，除了任务 A 之外的任务都为待分配任务。任务 F 不存在上游任务，任务 B 的全部上游任务执行完毕。任务 F 和任务 B 是本轮分配中的候选任务。机器 1 和机器 2 的预计空闲时刻分别为 6 和 0，最早空闲时刻为两个预计空闲时刻中的较小者，即为 0。任务 F 的可执行时刻为全部上游任务执行完毕时刻 0 和机器的最早空闲时刻 0 中的较大者，即为 0。任务 B 的可执行时刻为全部上游任务执行完毕时刻 6 和机器的最早空闲时刻 0 中的较大者，即为 6。因此选择可执行时刻最早的任务 F 作为目标任务。任务 F 在机器 1 上的预计开始时刻为上游任务执行完毕的时刻 0 与机器 1 的预计空闲时刻 6 中的较大者，即为 6。任务 F 在机器 1 上执行时的预计时长为预计通信时长 4 加上预计执行时长 3，即为 7。任务 F 在机器 1 上执行时的预计完成时刻为预计开始时刻 6 加上预计时长 7，即为 13。同理，任务 F 在机器 2 上的预计开始时刻为 0，预计时长为 7，预计完成时刻为 7。因此，选择预计完成时刻最早的机器 2 作为目标机器。在第二轮分配中，将任务 F 分配到机器 2 上。

接下来的描述中，对重复的流程不再赘述。

在第三轮分配中，任务 B 和任务 C 是本轮分配中的候选任务。机器 1 和机器 2 的最早空闲时刻为两个预计空闲时刻中的较小者 6。任务 B 和任 C 的可执行时刻分别为 6 和 7。因此选择任务 B 作为目标任务。任务 B 在机器 1 上的预计开始时刻为 6。任务 B 在机器 1 上执行时的预计时长为预计通信时长 0（上游任务 A 在机器 1 上执行过）加上预计执行时长 5，即为 5。任务 B 在机器 1 上执行时的预计完成时刻为预计开始时刻 6 加上预计时长 5，即为 11。同理，任务 B 在机器 2 上的预计开始时刻为 7，预计时长为 9，预计完成时刻为 16。因此，选择机器 1 作为目标机器。在第三轮分配中，将任务 B 分配到机器 1 上。

在第四轮分配中，任务 C 和任务 D 是本轮分配中的候选任务。机器 1 和机器 2 的最早空闲时刻为 7。任务 C 和任 D 的可执行时刻分别为 7 和 11。因此选择任务 C 作为目标任务。任务 C 在机器 1 上的预计开始时刻为 11。任务 C 在机器 1 上执行时的预计时长为预计通信时长 3（获取上游任务 F 的执行结果）加上预计执行时长 2，即为 5。任务 C 在机器 1 上执行时的预计完成时刻为预计开始时刻 11 加上预计时长 5，即为 16。同理，任务 C 在机器 2 上的预计开始时刻为 7，预计时长为 6，预计完成时刻为 13。因此，选择机器 2 作为目标机器。在第四轮分配中，将任务 C 分配到机器 2 上。

在第五轮分配中，任务 D 和任务 E 是本轮分配中的候选任务。机器 1 和机器 2 的

最早空闲时刻为 11。任务 D 和任务 E 的可执行时刻分别为 11 和 13。因此选择任务 D 作为目标任务。任务 D 在机器 1 上的预计开始时刻为 11。任务 D 在机器 1 上执行时的预计时长为预计通信时长 0 加上预计执行时长 1，即为 1。任务 D 在机器 1 上执行时的预计完成时刻为预计开始时刻 11 加上预计时长 1，即为 12。同理，任务 D 在机器 2 上的预计开始时刻为 13，预计时长为 3，预计完成时刻为 16。因此，选择机器 1 作为目标机器。在第五轮分配中，将任务 D 分配到机器 1 上。

在第六轮分配中，任务 E 是本轮分配中的唯一的候选任务，也是目标任务。任务 E 在机器 1 上的预计开始时刻为 13（全部上游任务执行完毕的时刻 13 比机器 1 的预计空闲时刻 12 更晚）。任务 E 在机器 1 上执行时的预计时长为预计通信时长 3 加上预计执行时长 2，即为 5。任务 E 在机器 1 上执行时的预计完成时刻为预计开始时刻 13 加上预计时长 5，即为 18。同理，任务 E 在机器 2 上的预计开始时刻为 13，预计时长为 4，预计完成时刻为 17。因此，选择机器 2 作为目标机器。在第五轮分配中，将任务 E 分配到机器 2 上。

综上所述，在经过六轮任务分配之后，得到任务调度表：{机器 1：任务 A、任务 B、任务 D；机器 2：任务 F、任务 C、任务 E}。可以看出，经过本说明书提供的任务调度方法的分配之后，有上下游依赖关系的任务被尽可能分配到同一台机器上，减少了通信耗时，整体上提高了任务的执行效率。

本说明书另一方面提供一种计算机可读非暂时性存储介质，存储有至少一组用来进行任务调度的可执行指令。当所述可执行指令被处理器执行时，所述可执行指令指导所述处理器实施本说明书所述的任务调度方法 P100 的步骤。在一些可能的实施方式中，本说明书的各个方面还可以实现为一种程序产品的形式，其包括程序代码。当所述程序产品在任务调度系统 003 上运行时，所述程序代码用于使任务调度系统 003 执行本说明书描述的任务调度方法 P100 的步骤。用于实现上述方法的程序产品可以采用便携式紧凑盘只读存储器（CD-ROM）包括程序代码，并可以在任务调度系统 003 上运行。然而，本说明书的程序产品不限于此，在本说明书中，可读存储介质可以是任何包含或存储程序的有形介质，该程序可以被指令执行系统使用或者与其结合使用。所述程序产品可以采用一个或多个可读介质的任意组合。可读介质可以是可读信号介质或者可读存储介质。可读存储介质例如可以为但不限于电、磁、光、电磁、红外线、或半导体的系统、装置或器件，或者任意以上的组合。可读存储介质的更具体的例子包括：具有一个或多个导线的电连接、便携式盘、硬盘、随机存取存储器（RAM）、只读存储器（ROM）、可擦式

可编程只读存储器（EPROM 或闪存）、光纤、便携式紧凑盘只读存储器（CD-ROM）、光存储器件、磁存储器件、或者上述的任意合适的组合。所述计算机可读存储介质可以包括在基带中或者作为载波一部分传播的数据信号，其中承载了可读程序代码。这种传播的数据信号可以采用多种形式，包括但不限于电磁信号、光信号或上述的任意合适的组合。可读存储介质还可以是可读存储介质以外的任何可读介质，该可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。可读存储介质上包含的程序代码可以用任何适当的介质传输，包括但不限于无线、有线、光缆、RF 等等，或者上述的任意合适的组合。可以以一种或多种程序设计语言的任意组合来编写用于执行本说明书操作的程序代码，所述程序设计语言包括面向对象的程序设计语言—诸如 Java、C++等，还包括常规的过程式程序设计语言—诸如“C”语言或类似的程序设计语言。

上述对本申请特定实施例进行了描述。其他实施例在所附权利要求书的范围内。在一些情况下，在权利要求书中记载的动作或步骤可以按照不同于实施例中的顺序来执行并且仍然可以实现期望的结果。另外，在附图中描绘的过程不一定要求示出特定顺序或者连续顺序才能实现期望的结果。在某些实施方式中，多任务处理和并行处理也是可以的或者是可能有利的。

综上所述，在阅读本详细公开内容之后，本领域技术人员可以明白，前述详细公开内容可以仅以示例的方式呈现，并且可以不是限制性的。尽管这里没有明确说明，本领域技术人员可以理解本申请需求囊括对实施例的各种合理改变，改进和修改。这些改变，改进和修改旨在由本申请提出，并且在本申请的示例性实施例的精神和范围内。

此外，本申请中的某些术语已被用于描述本申请的实施例。例如，“一个实施例”，“实施例”和/或“一些实施例”意味着结合该实施例描述的特定特征，结构或特性可以包括在本申请的至少一个实施例中。因此，可以强调并且应当理解，在本申请的各个部分中对“实施例”或“一个实施例”或“替代实施例”的两个或更多个引用不一定都指代相同的实施例。此外，特定特征，结构或特性可以在本申请的一个或多个实施例中适当地组合。

应当理解，在本申请的实施例的前述描述中，为了帮助理解一个特征，出于简化本申请的目的，本申请将各种特征组合在单个实施例、附图或其描述中。然而，这并不是说这些特征的组合是必须的，本领域技术人员在阅读本申请的时候完全有可能将其中一部分设备标注出来作为单独的实施例来理解。也就是说，本申请中的实施例也可以理解

为多个次级实施例的整合。而每个次级实施例的内容在于少于单个前述公开实施例的所有特征的时候也是成立的。

本文引用的每个专利，专利申请，专利申请的出版物和其他材料，例如文章，书籍，说明书，出版物，文件，物品等，除了与其相关的任何历史起诉文件、可能与本文件不一致或相冲突的任何相同的、或者任何可能对权利要求的最宽范围具有限制性影响的任何相同的历史起诉文件，均可以通过引用结合于此，并用于现在或以后与本文件相关联的所有目的。此外，如果在与任何所包含的材料相关联的术语的描述、定义和/或使用与本文档相关的术语、描述、定义和/或之间存在任何不一致或冲突时，使用本文件中的术语为准。

最后，应理解，本文公开的申请的实施方案是对本申请的实施方案的原理的说明。其他修改后的实施例也在本申请的范围内。因此，本申请披露的实施例仅仅作为示例而非限制。本领域技术人员可以根据本申请中的实施例采取替代配置来实现本申请中的申请。因此，本申请的实施例不限于申请中被精确地描述过的实施例。

权　利　要　求　书

1. 一种任务调度方法，包括：

获得目标有向图，所述目标有向图是基于 N 个任务间的依赖关系形成的有向无环图，所述 N 为大于 1 的整数；

基于所述依赖关系，按照可执行时刻从早到晚的顺序，依次将所述 N 个任务分配到 M 个机器上，以生成调度表，所述 M 为大于 1 的整数，在每次任务分配中，将被分配的任务分配给预计最早能够执行完任务的机器，其中，所述可执行时刻是基于每个任务所依赖的上游任务全部完成的时刻以及所述 M 个机器的预计空闲时刻确定的；以及

输出所述调度表。

2. 如权利要求 1 所述的方法，其中，所述依赖关系包括下游任务的执行依赖上游任务的输出结果，所述 N 个任务包括至少一个起始任务，每个起始任务没有依赖的上游任务。

3. 如权利要求 1 所述的方法，其中，所述每次任务分配包括：

从待分配任务中确定最早可执行的目标任务；

确定所述目标任务在所述 M 个机器上执行时的 M 个预计完成时刻；以及

将所述目标任务分配给目标机器，所述目标机器为所述 M 个预计完成时刻中最早的时刻对应的机器。

4. 如权利要求 3 所述的方法，其中，所述从待分配任务中确定最早可执行的目标任务，包括：

从所述待分配任务中确定可执行的至少一个候选任务，其中，每个候选任务所依赖的全部上游任务都已被分配完毕；以及

从所述至少一个候选任务中确定所述目标任务。

5. 如权利要求 4 所述的方法，其中，所述从所述至少一个候选任务中确定所述目标任务，包括：

确定所述 M 个机器在前轮任务分配之后的最早空闲时刻，所述最早空闲时刻为所

述 M 个机器各自执行完已分配任务的 M 个预计空闲时刻中最早的时刻；

确定所述每个候选任务所依赖的上游任务全部完成的时刻；

对所述每个候选任务，将所述最早空闲时刻与当前候选任务所依赖的上游任务全部完成的时刻中的较大者确定为所述当前候选任务的可执行时刻；以及

从所述至少一个候选任务中确定所述可执行时刻最早的时刻对应的候选任务为所述目标任务。

6. 如权利要求 3 所述的方法，其中，所述确定所述目标任务在所述 M 个机器上执行时的 M 个预计完成时刻，包括：

确定所述目标任务在所述 M 个机器中的每个机器上执行时的预计开始时刻；

确定所述目标任务在所述每个机器上执行时所需的预计时长，所述预计时长包括预计执行时长以及预计通信时长，所述预计通信时长包括所述目标任务获取所需数据的时长；以及

将所述目标任务在所述每个机器上执行时的所述预计开始时刻与所述预计时长叠加，得到所述目标任务在所述每个机器上执行时的预计完成时刻，从而得到所述 M 个预计完成时刻。

7. 如权利要求 6 所述的方法，其中，所述确定所述目标任务在所述 M 个机器中的每个机器上执行时的预计开始时刻，包括：

确定所述每个机器执行完已分配任务的预计空闲时刻；

确定所述目标任务所依赖的上游任务全部完成的时刻；以及

对所述每个机器，将当前机器的所述预计空闲时刻与所述目标任务所依赖的上游任务全部完成的时刻中的较大者，作为所述目标任务在所述当前机器上执行时的预计开始时刻。

8. 如权利要求 6 所述的方法，其中，在所述目标任务依赖至少一个上游任务时，所述预计通信时长包括所述至少一个上游任务对应的至少一个通信时长中最大的时长，

其中，对所述至少一个上游任务中的每个上游任务，当所述目标任务与当前上游任务在同一个机器上执行时所需的通信时长小于所述目标任务与所述当前上游任务不在同一个机器上执行时所需的通信时长。

9. 如权利要求 8 所述的方法，其中，对所述每个上游任务，当所述目标任务与所述当前上游任务在同一个机器上执行时所需的通信时长为零。

10. 如权利要求 6 所述的方法，其中，所述目标任务在不同的机器上执行时的预计执行时长相同。

11. 如权利要求 6 所述的方法，其中，所述目标任务在不同的机器上执行时的预计执行时长与机器处理速度负相关。

12. 一种任务调度系统，包括：

至少一个存储介质，存储有至少一个指令集，用于进行任务调度；以及

至少一个处理器，同所述至少一个存储介质通信连接，其中，所述至少一个处理器运行时读取所述至少一个指令集，以根据所述至少一个指令集的指示：

获得目标有向图，所述目标有向图是基于 N 个任务间的依赖关系形成的有向无环图，所述 N 为大于 1 的整数；

基于所述依赖关系，按照可执行时刻从早到晚的顺序，依次将所述 N 个任务分配到 M 个机器上，以生成调度表，所述 M 为大于 1 的整数，在每次任务分配中，将被分配的任务分配给预计最早能够执行完任务的机器，其中，所述可执行时刻是基于每个任务所依赖的上游任务全部完成的时刻以及所述 M 个机器的预计空闲时刻确定的；以及

输出所述调度表。

13. 如权利要求 12 所述的系统，其中，所述依赖关系包括下游任务的执行依赖上游任务的输出结果，所述 N 个任务包括至少一个起始任务，每个起始任务没有依赖的上游任务。

14. 如权利要求 12 所述的系统，其中，在所述每次任务分配中，所述至少一个处理器：

从待分配任务中确定最早可执行的目标任务；

确定所述目标任务在所述 M 个机器上执行时的 M 个预计完成时刻；以及

将所述目标任务分配给目标机器，所述目标机器为所述 M 个预计完成时刻中最早的时刻对应的机器。

15. 如权利要求 14 所述的系统，其中，为了从待分配任务中确定最早可执行的目标任务，所述至少一个处理器：

从所述待分配任务中确定可执行的至少一个候选任务，其中，每个候选任务所依赖的全部上游任务都已被分配完毕；以及

从所述至少一个候选任务中确定所述目标任务。

16. 如权利要求 15 所述的系统，其中，为了从所述至少一个候选任务中确定所述目标任务，所述至少一个处理器：

确定所述 M 个机器在前轮任务分配之后的最早空闲时刻，所述最早空闲时刻为所述 M 个机器各自执行完已分配任务的 M 个预计空闲时刻中最早的时刻；

确定所述每个候选任务所依赖的上游任务全部完成的时刻；

对所述每个候选任务，将所述最早空闲时刻与当前候选任务所依赖的上游任务全部完成的时刻中的较大者确定为所述当前候选任务的可执行时刻；以及

从所述至少一个候选任务中确定所述可执行时刻最早的时刻对应的候选任务为所述目标任务。

17. 如权利要求 14 所述的系统，其中，为了确定所述目标任务在所述 M 个机器上执行时的 M 个预计完成时刻，所述至少一个处理器：

确定所述目标任务在所述 M 个机器中的每个机器上执行时的预计开始时刻；

确定所述目标任务在所述每个机器上执行时所需的预计时长，所述预计时长包括预计执行时长以及预计通信时长，所述预计通信时长包括所述目标任务获取所需数据的时长，在所述目标任务依赖至少一个上游任务时，所述预计通信时长包括所述至少一个上游任务对应的至少一个通信时长中最大的时长，其中，对所述至少一个上游任务中的每个上游任务，当所述目标任务与当前上游任务在同一个机器上执行时所需的通信时长小于所述目标任务与所述当前上游任务不在同一个机器上执行时所需的通信时长；以及

将所述目标任务在所述每个机器上执行时的所述预计开始时刻与所述预计时长叠加，得到所述目标任务在所述每个机器上执行时的预计完成时刻，从而得到所述 M 个

预计完成时刻。

18. 如权利要求 17 所述的系统，其中，为了确定所述目标任务在所述 M 个机器中的每个机器上执行时的预计开始时刻，所述至少一个处理器：

确定所述每个机器执行完已分配任务的预计空闲时刻；

确定所述目标任务所依赖的上游任务全部完成的时刻；以及

对所述每个机器，将当前机器的所述预计空闲时刻与所述目标任务所依赖的上游任务全部完成的时刻中的较大者，作为所述目标任务在所述当前机器上执行时的预计开始时刻。

19. 如权利要求 17 所述的系统，其中，所述目标任务在不同的机器上执行时的预计执行时长相同或所述预计执行时长与机器处理速度负相关。

20. 一种计算机可读非易失性存储介质，其中，所述计算机可读非易失性存储介质中存储有至少一个指令集，所述至少一个指令集被至少一个处理器执行时，实现如权利要求 1-11 中任一项所述的任务调度方法。