

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2020 年 11 月 26 日 (26.11.2020)



(10) 国际公布号
WO 2020/232875 A1

- (51) 国际专利分类号:
G06F 9/52 (2006.01)
- (21) 国际申请号: PCT/CN2019/102741
- (22) 国际申请日: 2019 年 8 月 27 日 (27.08.2019)
- (25) 申请语言: 中文
- (26) 公布语言: 中文
- (30) 优先权:
201910421282.5 2019年5月20日 (20.05.2019) CN
- (71) 申请人: 平安科技(深圳)有限公司(PING AN TECHNOLOGY (SHENZHEN) CO., LTD.) [CN/CN];
中国广东省深圳市福田区福田街道福安社区益田路 5033 号平安金融中心 23 楼, Guangdong 518000 (CN)。
- (72) 发明人: 袁康(YUAN, Kang); 中国广东省深圳市福田区福田街道福安社区益田路 5033 号平安金融中心 23 楼, Guangdong 518000 (CN)。

- (74) 代理人: 广州三环专利商标代理有限公司 (SCIHEAD IP LAW FIRM); 中国广东省广州市越秀区先烈中路 80 号汇华商贸大厦 1508 室, Guangdong 510070 (CN)。
- (81) 指定国(除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。
- (84) 指定国(除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG,

(54) **Title:** ACTOR MODEL-BASED TASK SCHEDULING METHOD AND APPARATUS, AND STORAGE MEDIUM

(54) 发明名称: 一种基于Actor模型的任务调度方法、装置及存储介质

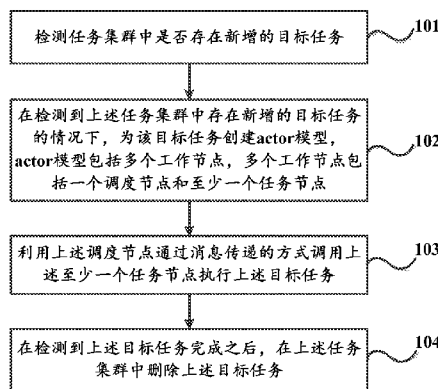


图 1

- 101 Detect whether a newly added target task is present in a task cluster
- 102 If detected that a newly added target task is present in the task cluster, create an Actor model for the target task, the Actor model comprising a plurality of work nodes, and the plurality of work nodes comprising one scheduling node and at least one task node
- 103 Use the scheduling node to call the at least one task node by means of a message delivery manner to execute the target task
- 104 Once detected that the target task is completed, delete the target task in the task cluster

(57) **Abstract:** Disclosed by the present application are an Actor model-based task scheduling method and apparatus and a storage medium, which are applied in the field of data processing, the method comprising: detecting whether a newly added target task is present in a task cluster; if detected that a newly added target task is present in the task cluster, creating an Actor model for the target task, the Actor model comprising a plurality of work nodes, and the plurality of work nodes comprising one scheduling node and at least one task node; using the scheduling node to call the at least one task node by means of a message delivery manner to execute the target task; once detected that the target task is completed, deleting the target task in the task cluster. Provided in the present application is an Actor model-based task scheduling method. Since the scheduling node and task node are both Actor nodes, the present application does not require locks to be able to complete task scheduling, which greatly improves the efficiency of task scheduling.

CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU,
IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT,
RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI,
CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

— 包括国际检索报告(条约第21条(3))。

(57) 摘要: 本申请公开了一种基于Actor模型的任务调度方法、装置及存储介质, 应用于数据处理领域, 其中方法包括: 检测任务集群中是否存在新增的目标任务; 在检测到任务集群中存在新增的目标任务的情况下, 为目标任务创建Actor模型, Actor模型包括多个工作节点, 多个工作节点包括一个调度节点和至少一个任务节点; 利用调度节点通过消息传递的方式调用至少一个任务节点执行目标任务; 在检测到目标任务完成之后, 在任务集群中删除目标任务。本申请提供了一种基于Actor模型的任务调度方法, 由于调度节点和任务节点都是Actor节点, 使得本申请不需要锁便能完成任务调度, 大大的提高了任务调度的效率。

一种基于 Actor 模型的任务调度方法、装置及存储介质

本申请要求于 2019 年 05 月 20 日提交中国专利局、申请号为 201910421282.5、申请名称为“一种基于 Actor 模型的任务调度方法、装置及存储介质”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。

5

技术领域

本申请涉及任务调度领域，尤其涉及一种基于 Actor 模型的任务调度方法、装置及存储介质。

10 背景技术

任务调度常常因为多线程或多进程并发访问某个临界资源而陷入死锁，这时可通过对该临界资源进行加锁，来建立线程或进程之间的互斥同步机制，以保障临界资源的完整性和一致性。单机的进程同步和线程同步有很多可用的互斥方案，比如互斥锁、信号量和条件变量等。但这些单机下的互斥方案在多机上是无法使用的，因为分布式系统的组件运行在不同的机器上，使得其不在统一的运行环境中。

15

分布式任务调度中，常常通过分布式锁解决对临界资源的互斥访问，比如 Redis 分布式锁和 Zookeeper 分布式锁的等。

但是分布式锁的实现比较复杂，且维护比较麻烦，需要较多维护成本，因此分布式锁不是一个解决分布式任务调度的死锁问题的理想方案，还是缺少了一种可以更高效的分布式任务调度方法。

20

发明内容

本申请实施例提供一种基于 Actor 模型的任务调度方法，可以不使用锁便实现分布式任务调度和处理。

25

第一方面，本申请实施例提供了一种基于 Actor 模型的任务调度方法，该方法包括：检测任务集群中是否存在新增的目标任务；

在检测到所述任务集群中存在新增的目标任务的情况下，为所述目标任务创建 Actor 模型，所述 Actor 模型包括多个工作节点，所述多个工作节点之间通过消息传递的方式通信，所述多个工作节点中至少存在两个工作节点分别分布于不同的终端上，所述多个工作节点包括一个调度节点和至少一个任务节点，所述调度节点用于调用所述任务节点执行任务；

30

利用所述调度节点通过所述消息传递的方式调用所述至少一个任务节点执行所述目标任务；

在检测到所述目标任务完成之后，在所述任务集群中删除所述目标任务。

第二方面，本申请实施例提供了一种任务调度装置，该任务调度装置包括用于执行上述第一方面的任务调度方法的单元，该任务调度装置包括：

检测单元，用于检测任务集群中是否存在新增的目标任务；

创建单元，用于在检测到任务集群中存在新增的目标任务的情况下，为所述目标任务创建 Actor 模型，所述 Actor 模型包括多个工作节点，所述多个工作节点之间通过消息传递的方式通信，所述多个工作节点中至少存在两个工作节点分别分布于不同的终端上，所述多个工作节点包括一个调度节点和至少一个任务节点，所述调度节点用于调用所述任务节点执行任务；

执行单元，用于利用所述调度节点通过所述消息传递的方式调用所述至少一个任务节点执行所述目标任务；

删除单元，用于在检测到所述目标任务完成之后，在所述任务集群中删除所述目标任务。

第三方面，本申请实施例提供了另一种任务调度装置，包括处理器和存储器，所述处理器和存储器相互连接，其中，所述存储器用于存储支持任务调度装置执行上述任务调度方法的计算机程序，所述计算机程序包括程序指令，所述处理器被配置用于调用所述程序指令，用以执行：

检测任务集群中是否存在新增的目标任务；

在检测到所述任务集群中存在新增的目标任务的情况下，为所述目标任务创建 Actor 模型，所述 Actor 模型包括多个工作节点，所述多个工作节点之间通过消息传递的方式通信，所述多个工作节点中至少存在两个工作节点分别分布于不同的终端上，所述多个工作节点包括一个调度节点和至少一个任务节点，所述调度节点用于调用所述任务节点执行任务；

利用所述调度节点通过所述消息传递的方式调用所述至少一个任务节点执行所述目标任务；

在检测到所述目标任务完成之后，在所述任务集群中删除所述目标任务。

第四方面，本申请实施例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存储有计算机程序，所述计算机程序包括程序指令，所述程序指令当被处理器执行，用以执行上述第一方面至第一方面的任意一种实现方式的任务调度方法。

本申请不需要借助于锁便可以实现复杂的分布式任务调度，保证分布式任务调度过程中不会产生数据竞争和死锁，并且大大提高分布式任务处理的并发性能。

附图说明

为了更清楚地说明本申请实施例技术方案，下面将对实施例描述中所需要使用的附图作简单地介绍。

图 1 是本申请实施例提供的一种任务调度方法的示意流程图；

图 2 是本申请另一实施例提供的一种任务调度方法的示意流程图；

图 3 是本申请实施例提供的一种任务调度装置的示意性框图；

图 4 是本申请实施例提供的一种任务调度装置的结构框图。

5 具体实施方式

下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行清楚、完整地描述。

本申请主要应用于任务调度装置，该任务调度装置可以是传统任务调度装置、大型存储系统、台式电脑、笔记本电脑、平板电脑、掌上电脑、智能手机、便携式数字播放器、智能手表以及智能手环等等，本申请对此不做限制。

本申请实施例中描述的终端包括但不限于带通讯功能的设备、智能手机、平板电脑、笔记本电脑、台式电脑、便携式数字播放器、智能手环以及智能手表等。当终端向任务调度装置发送数据的时候，按照预设格式进行对数据的特性进行记录并传送，其中，数据的特性包括时间、地点、类型等。

参见图 1，是本申请实施例提供一种任务调度方法的示意流程图，如图 1 所示任务调度方法可包括：

101：检测任务集群中是否存在新增的目标任务。

在本申请实施例中，检测任务集群中是否存在新增的任务，若存在新增的任务，则将该新增的任务作为目标任务。其中，任务集群指的是待处理的任务的集合，且该任务集群中的不同任务可能由不同的终端所提出，而任务集群中的任务可能是由多个终端共同处理的分布式任务，也可能是仅由一个终端处理的单机任务，检测的过程指的是，本端的任务调度装置定时检测任务集群中是否存在新增的任务，或者任务集群中新增一个任务，本端装置便能接收到用于指示任务集群中新增任务的消息。

在另一种可实施的方式中，上述目标任务不是任务集群中新增的任意任务，而是任务集群中新增的分布式任务，具体的，上述检测任务集群中是否存在新增的目标任务指的是，检测任务集群中的新增的任务，若确定该新增的任务为分布式任务，则确定检测到任务集群中存在新增的目标任务。

在本申请实施例中，上述检测任务集群中新增的任务为分布式任务的过程指的是，获取上述任务集群中新增的任务的类型，若该新增的任务的类型为分布式任务，则表示该新增的任务的类型为分布式任务，或者，获取上述新增的任务的难度等级，若确定该新增的任务的难度等级超过预设级别，则说明该新增的任务需要占用较多的计算资源，并确定该新增的任务为分布式任务，也即是将该难度级别超过预设级别的任务作为目标任务，并接下来采用分布式任务的处理方法对其进行处理。

可见，本申请实施例只有在确定任务集群新增了分布式任务的时候，才确定任务集群中有新增的目标任务，上述检测到的任务集群中新增的分布式任务即目标任务。

102 :在检测到上述任务集群中存在新增的目标任务的情况下,为该目标任务创建 Actor 模型,Actor 模型包括多个工作节点,多个工作节点包括一个调度节点和至少一个任务节点。

在本申请实施例中,Actor 模型为一个用于任务处理的模型,具体来说 Actor 模型是一个概念模型,用于处理并发计算,它定义了一系列系统组件应该如何动作和交互的通用规则。Actor 模型中包含了多个 Actor,Actor 在本申请实施例中被称为工作节点,每个工作节点是 Actor 模型中进行任务处理的最基本的计算单元,它能接收一个消息并且基于其执行计算。于是上述根据该目标任务的执行内容创建包含多个工作节点的 Actor 模型指的是,对目标任务进行分解,然后基于分解之后的目标任务的任务内容,创建多个工作节点来执行对应的任务内容,而该用于执行目标任务的工作节点即任务节点,而在执行任务的时候,多个任务节点可能是并行执行的,如果调度任务节点时调度不当,可能会造成多个任务节点之间的资源竞争,于是为了确定调度的顺序,并避免资源竞争,上述 Actor 模型中还包
5 括用于执行任务调度的工作节点,该用于执行任务调度的工作节点在本申请实施例中被称为调度节点,调度节点用于安排各个任务节点的执行顺序,并调度各个任务节点有序执行任务。总的来说,为上述目标任务创建的 Actor 模型中包含多个工作节点,该多个工作节点中包含一个调度节点和至少一个任务节点。
10

需要说明的是,由于本申请实施例采用的调度节点和任务节点都是 Actor,而 Actor 之间是通过消息传递的方式进行通信,因此不管是在任务调度,还是任务执行的时候,都不会产生资源竞争的问题。其中,消息传递的方式指的是在消息传递系统中,进程间的数据交换是以格式化的消息(Message)为单位的,Actors 的一大重要特征在于 Actors 之间相互隔离,它们并不互相共享内存,一个 Actor 能维持一个私有的状态,并且这个状态不可能被另一个 Actor 所改变,而消息传递也保证了 Actor 之间的异步处理,因此 Actor 之间也不会存在数据竞争。若通信的进程之间不存在可直接访问的共享空间,则必须利用操作系统提供的消息传递方法实现进程通信,进程通过系统提供的发送消息和接收消息两个原语进行数据交换。简单来说,Actor 之间通过消息传递的方式来进行资源共享,当 Actor 需要另一个 Actor 的资源的时候,只需要将消息发送给另一个 Actor,但不关心另一个 Actor 什么时候处理,因此 Actor 之间的消息是异步传输的,类似邮件通信。具体的,在 Actor 模型里每个 Actor 都有地址,所以它们才能够相互发送消息,尽管许多 Actor 同时运行,但是一个 Actor 只能顺序地处理消息。也就是说其它三个 Actor 发送了给一个 Actor 分别发送了一条消息,这个 Actor 只能一次处理一条,如果需要并行处理三条消息,则需要三个 Actor 分别
20 对这三条消息进行处理,并且如果 Actor 在处理消息的过程中又接收到一条新的消息,则该新的消息会被存储到 Actor 的邮箱 Mailbox 中,每个 Actor 对应有一个 Mailbox。
25

可见,Actor 通过消息传递的方式将不同 Actor 的可访问的资源进行了分隔,并保障了 Actor 之间的消息的异步处理,从而避免了使用锁来处理数据条件竞争(data race)的问题。消息传递机制最大的优点就是不会产生数据竞争状态(data race)。

35 还需要说明的是,上述多个工作节点中至少存在两个工作节点分别分布于不同的终端

上，于是本申请实施例应用于分布式任务的调度和处理，可见，由于位于不同的终端上的工作节点在工作的时候，只会占用其所在终端的计算资源，于是在一个终端集群中的多个终端共同处理一个分布式任务，相比单机处理，分布式处理可以大大提升了计算能力和计算速度。需要注意的是，由于本申请实施例采用了 Actor 模型中的 Actor 来作为工作节点，
5 才可以不需要分布式锁便能实现分布式任务调度和处理，这是因为 Actor 只关注接收到的消息的内容，而不关注消息的发送方是本地的 Actor 还是终端上的 Actor，于是只要保证多个 Actor 之间的消息能够顺利达到即可，也允许了本申请实施例可以不及住锁便能在包含多个终端的终端集群中构建 Actor 模型，并实现分布式任务的调度和处理。

在另一种可实施的方式中，上述多个工作节还包括至少一个备用调度节点，备用调度节点用于替换宕机的调度节点。若检测到上述调度节点宕机，则从上述至少一个备用调度节点中选取一个备用调度节点作为新的调度节点。
10

在本申请实施例中，为目标任务创建的 Actor 模型中除了上述任务节点和调度节点以外，还包括至少一个备用调度节点，该备用调度节点用于在调度节点宕机的时候，替换该宕机的调度节点作为新的调度节点，然后新的调度节点可以从头开始执行调度，或者接着宕机的调度节点在宕机之前的调度进度，继续执行调度。其中，备用调度节点是按照调度节点的初始化信息创建的，因此备用调度节点可以替代调度节点。
15

可见，本申请实施例通过创建至少一个备用调度节点，可以在调度节点宕机时，启用该备用调度节点，以解决调度节点可能宕机的问题。

在一种可实施的方式中，若上述 Actor 模型中只包括一个备用调度节点，则将在调度节点宕机的时候，直接将上述 Actor 模型中仅包括的一个备用调度节点作为新的调度节点，若上述 Actor 模型中包括多个备用调度节点，则在调度节点宕机的时候，从该多个备用调度节点中选取任意一个备用调度节点作为新的调度节点，或者，分别获取每个备用调度节点的创建时间，并确定 Actor 模型包括的备用调度节点中创建时间最早的调度节点，然后将创建时间最早的备用调度节点作为新的调度节点。
20

可见，如果是根据备用调度时间的早晚来确定用于替换宕机的调度节点的备用调度节点，好处在于，便于本端装置通过读取当前使用的调度节点的创建时间，可以快速判定已宕机的调用节点的个数，从而获知调用节点的宕机情况。
25

在一种可实施的方式中，上述为目标任务创建 Actor 模型指的是，获取目标任务的标签，标签用于指示目标任务的重要程度，并获取目标任务的标签对应的目标数量，目标数量用于指示 Actor 模型中的备用调度节点的个数，然后创建包括目标数量的备用调度节点的 Actor 模型。
30

在本申请实施例中，在创建 Actor 模型的时候，Actor 模型中的备用调度节点的个数多少与目标任务的重要程度有关，目标任务越重要，就为目标任务创建更多的备用调度节点。具体的，在为目标任务创建 Actor 模型的时候，获取目标任务的标签，并在数据库中获取该目标任务的标签对应的目标数量，然后创建包含目标数量的备用调度节点。其中，数
35

数据库中保存有任务的标签与备用调度节点的个数的对应关系。

在另一种可实施的方式中，上述多个工作节点还包括至少一个监控节点，监控节点用于初始化宕机的调度节点，具体的，若检测到调度节点宕机，则利用监控节点对该宕机的调度节点进行初始化，使得该宕机的调度节点恢复正常。

5 本申请实施例描述了另一种解决调度节点宕机问题的方法，即如果调度节点宕机，则使用该监控设备对宕机的调度节点进行初始化。具体的，在检测到调度节点宕机之后，使用监控节点根据调度节点的初始化信息对宕机的调度节点进行初始化，使得宕机的调度节点恢复正常。或者，监控节点用于将宕机的调度节点恢复到宕机前一个时刻的正常状态，具体的，在检测到上述调度节点宕机之后，获取记录在数据库中的该宕机的调度节点在宕机前一个时刻的状态信息，并使用监控节点根据上述调度节点在宕机前一个时刻的状态信息对调度节点进行恢复，使得调度节点恢复到上述状态信息对应的正常状态。

10 在另一种可实施的方式中，在为目标任务创建 Actor 模型的时候，根据目标任务的重要程度决定是否在上述 Actor 模型中创建监控节点。具体的，在任务集群中检测到目标任务之后，获取该目标任务的标签，该标签用于指示目标任务的重要程度。获取目标任务的标签对应的目标任务的重要等级，若目标任务的重要等级超过了预设等级，则在创建 Actor 模型的时候，创建上述监控节点，若目标任务的重要等级未超过预设等级，则不创建上述监控节点。

在另一种可实施的方式中，上述监控节点还用于对宕机的任务节点进行初始化，以使得该宕机的任务节点恢复正常。

20 在另一种可实施的方式中，上述为目标任务创建的 Actor 模型针对于该模型中包含的每个任务节点分别包括至少一个备用任务节点，该备用任务节点用于替换宕机的任务节点。也即是说 Actor 模型中的一个任务节点对应一个备用任务节点，任务节点与备用任务节点是一一对应的。具体的，本方案还包括：检测宕机的任务节点；若检测到宕机的任务节点，则从 Actor 模型中的至少一个备用任务节点中选取与宕机的任务节点对应的备用任务节点，并将该选出的备用任务节点作为新的任务节点。

25 103：利用上述调度节点通过消息传递的方式调用上述至少一个任务节点执行上述目标任务。

30 在本申请实施例中，利用调度节点通过消息传递的方式，调用上述至少一个任务节点执行上述目标任务。具体的，调度节点将用于触发任务节点执行任务的消息发送给任务节点，使得任务节点执行任务，在任务节点执行完任务之后，会向调度节点反馈一个用于指示任务完成的信息，使得调度节点获取完成任务的任务节点，并继续按照执行顺序调度执行其他任务节点，从而最终完成目标任务。

104：在检测到上述目标任务完成之后，在上述任务集群中删除上述目标任务。

35 在本申请实施例中，在检测到上述目标任务完成之后，便在上述任务集群中删除上述目标任务，并检测和执行任务集群中的下一个目标任务。

在另一种可实施的方式中，上述在任务集群中删除目标任务之后，检测已创建的工作节点的个数，并获取所述已创建的工作节点的创建时间；若所述已创建的工作节点的个数大于预设个数，则按照所述创建时间从早到晚的排序，依次删除所述已创建的工作节点，直到剩下的已创建的工作节点的个数小于或等于所述预设个数。

5 在本申请实施例中，系统中的工作节点可能太多而导致本端装置存储空间不足或者系统运行缓慢，于是在完成目标任务之后，检测已创建的工作节点的个数，如果已创建的工作节点的个数超过预设个数，就对工作节点进行删减，直到剩下的工作节点的个数小于等于预设个数。其中，按照工作节点的创建时间由早到晚顺序进行删除，这是因为早创建的工作节点，相较于晚创建的工作节点，被再次使用的几率更小。

10 上述删除已创建的工作节点的详细过程为，检测已创建的工作节点的个数，并获取该已创建的工作节点的创建时间，若该已创建的工作节点的个数大于预设个数，则确定该已创建的工作节点中创建时间最早的时间节点，并删除所述已创建的工作节点中创建时间最早的工作节点，然后若在删除该已创建的工作节点中创建时间最晚的工作节点之后，已创建的工作节点的个数仍大于预设个数，则触发执行上述确定已创建的工作节点中创建时间
15 最早的工作节点，至删除已创建的工作节点中创建时间最早的工作节点的步骤，直到已创建的工作节点的个数在删除了创建时间最晚的工作节点之后小于或等于预设个数。

在本申请实施例中，每当检测到目标任务，就为该目标任务建立包括调度节点和任务节点的 Actor 模型，并利用调度节点来调度任务节点执行目标任务，在检测到目标任务完成之后，删除该目标任务，可见，本申请所提供的任务调度方法中，由于调度节点和任务
20 节点都是 Actor，于是调度节点和任务节点是通过信息传递的方式进行通信的，而信息传递的好处就是不会产生数据竞争的问题，在处理任务的时候无需同步阻塞等待，于是本申请也便不需要借助于锁便可以实现复杂的分布式任务调度，并且大大提高分布式任务处理的并发性能。可见本申请实施例提供了一种高效的任务调度方法。

25 参见图 2，是本申请实施例提供另一种任务调度方法的示意图，如图 2 所示任务调度方法可包括：

201：检测任务集群中的新增的任务。

在本申请实施例中，检测任务集群中是否存在新增的任务，具体的检测过程指的是，本端的任务调度装置定时检测任务集群中是否存在新增的任务，或者任务集群中新增一个
30 任务，本端装置便能接收到用于指示任务集群中新增任务的消息。其中，任务集群指的是待处理的任务的集合，且该任务集群中的不同任务可能由不同的终端所提出，而任务集群中的任务可能是由多个终端共同处理的分布式任务，也可能是仅由一个终端处理的单机任务。

202：若确定上述新增的任务为分布式任务，则确定检测到上述任务集群中存在新增的
35 目标任务。

在本申请实施例中，在检测到任务集群中的新增的任务之后，进一步确定该新增的任务是否为目标任务。具体的，确定该新增的任务是否为分布式任务，若是则确定检测到任务集群中存在新增的目标任务。

在本申请实施例中，上述检测任务集群中新增的任务为分布式任务的过程指的是，获取上述任务集群中新增的任务的类型，若该新增的任务的类型为分布式任务，则表示该新增的任务的类型为分布式任务，或者，获取上述新增的任务的难度等级，若确定该新增的任务的难度等级超过预设级别，则说明该新增的任务需要占用较多的计算资源，并确定该新增的任务为分布式任务，也即是将该难度级别超过预设级别的任务作为目标任务，并接下来采用分布式任务的处理方法对其进行处理。

可见，本申请实施例只有在确定任务集群新增了分布式任务的时候，才确定任务集群中有新增的目标任务，上述检测到的任务集群中新增的分布式任务即目标任务。

203 :在检测到上述任务集群中存在新增的目标任务的情况下，为该目标任务创建 actor 模型，actor 模型包括多个工作节点，多个工作节点包括一个调度节点、至少一个任务节点和至少一个备用调度节点。

在本申请实施例中，Actor 模型为一个用于任务处理的模型，具体来说 Actor 模型是一个概念模型，用于处理并发计算，它定义了一系列系统组件应该如何动作和交互的通用规则。Actor 模型中包含了多个 Actor，Actor 在本申请实施例中被称为工作节点，每个工作节点是 Actor 模型中进行任务处理的最基本的计算单元，它能接收一个消息并且基于其执行计算。于是上述根据该目标任务的执行内容创建包含多个工作节点的 Actor 模型指的是，对目标任务进行分解，然后基于分解之后的目标任务的任务内容，创建多个工作节点来执行对应的任务内容，而该用于执行目标任务的工作节点即任务节点，而在执行任务的时候，多个任务节点可能是并行执行的，如果调度任务节点时调度不当，可能会造成多个任务节点之间的资源竞争，于是为了确定调度的顺序，并避免资源竞争，上述 Actor 模型中还包括用于执行任务调度的工作节点，该用于执行任务调度的工作节点在本申请实施例中被称为调度节点，调度节点用于安排各个任务节点的执行顺序，并调度各个任务节点有序执行任务。此外，由于调度节点只有一个，于是为了应对调度节点的情况，Actor 中还包括至少一个备用调度节点，备用调度节点用于替换宕机的调度节点。总的来说，为上述目标任务创建的 Actor 模型中包含多个工作节点，该多个工作节点中包含一个调度节点、至少一个任务节点和至少一个备用调度节点。

需要说明的是，由于本申请实施例采用的调度节点和任务节点都是 Actor，而 Actor 之间是通过消息传递的方式进行通信，因此不管是在任务调度，还是任务执行的时候，都不会产生资源竞争的问题。其中，消息传递的方式指的是在消息传递系统中，进程间的数据交换是以格式化的消息(Message)为单位的，Actors 的一大重要特征在于 Actors 之间相互隔离，它们并不互相共享内存，一个 Actor 能维持一个私有的状态，并且这个状态不可能被另一个 Actor 所改变，而消息传递也保证了 Actor 之间的异步处理，因此 Actor 之间也不会

存在数据竞争。若通信的进程之间不存在可直接访问的共享空间，则必须利用操作系统提供的消息传递方法实现进程通信，进程通过系统提供的发送消息和接收消息两个原语进行数据交换。简单来说，Actor 之间通过消息传递的方式来进行资源共享，当 Actor 需要另一个 Actor 的资源的时候，只需要将消息发送给另一个 Actor，但不关心另一个 Actor 什么时候处理，因此 Actor 之间的消息是异步传输的，类似邮件通信。具体的，在 Actor 模型里每个 Actor 都有地址，所以它们才能够相互发送消息，尽管许多 Actor 同时运行，但是一个 Actor 只能顺序地处理消息。也就是说其它三个 Actor 发送了给一个 Actor 分别发送了一条消息，这个 Actor 只能一次处理一条，如果需要并行处理三条消息，则需要三个 Actor 分别对这三条消息进行处理，并且如果 Actor 在处理消息的过程中又接收到一条新的消息，则该新的消息会被存储到 Actor 的邮箱 Mailbox 中，每个 Actor 对应有一个 Mailbox。

可见，Actor 通过消息传递的方式将不同 Actor 的可访问的资源进行了分隔，并保障了 Actor 之间的消息的异步处理，从而避免了使用锁来处理数据条件竞争(data race)的问题。消息传递机制最大的优点就是不会产生数据竞争状态(data race)。

还需要说明的是，上述多个工作节点中至少存在两个工作节点分别分布于不同的终端上，于是本申请实施例应用于分布式任务的调度和处理，可见，由于位于不同的终端上的工作节点在工作的时候，只会占用其所在终端的计算资源，于是在一个终端集群中的多个终端共同处理一个分布式任务，相比单机处理，分布式处理可以大大提升了计算能力和计算速度。需要注意的是，由于本申请实施例采用了 Actor 模型中的 Actor 来作为工作节点，才可以不需要分布式锁便能实现分布式任务调度和处理，这是因为 Actor 只关注接收到的消息的内容，而不关注消息的发送方是本地的 Actor 还是终端上的 Actor，于是只要保证多个 Actor 之间的消息能够顺利达到即可，也允许了本申请实施例可以不及住锁便能在包含多个终端的终端集群中构建 Actor 模型，并实现分布式任务的调度和处理。

204 :利用上述调度节点通过消息传递的方式调用上述至少一个任务节点执行上述目标任务。

在本申请实施例中，利用调度节点通过消息传递的方式，调用上述至少一个任务节点执行上述目标任务。具体的，调度节点将用于触发任务节点执行任务的消息发送给任务节点，使得任务节点执行任务，在任务节点执行完任务之后，会向调度节点反馈一个用于指示任务完成的信息，使得调度节点获取完成任务的任务节点，并继续按照执行顺序调度执行其他任务节点，从而最终完成目标任务。

205 :若检测到上述调度节点宕机，则从上述至少一个备用调度节点中选取一个备用调度节点作为新的调度节点。

在本申请实施例中，Actor 模型中的备用调度节点用于在调度节点宕机的时候，替换该宕机的调度节点作为新的调度节点，然后新的调度节点可以从头开始执行调度，或者接着宕机的调度节点在宕机之前的调度进度，继续执行调度。其中，备用调度节点是按照调度节点的初始化信息创建的，因此备用调度节点可以替代调度节点。

需要说明的是，步骤 205 与步骤 204 之间不存在先后关系，步骤 205 是在步骤 204 的执行过程中进行，从步骤 204 开始，步骤 204 结束之间的任意一个时间点，只要检测到调度节点宕机，就从上述至少一个备用调度节点中选取一个备用调度节点作为新的调度节点，然后继续利用新的调度节点执行上述步骤 204，在步骤 204 执行结束之后，即表示目标任务完成，于是步骤 205 也结束了。

可见，本申请实施例通过创建至少一个备用调度节点，可以在调度节点宕机时，启用该备用调度节点，以解决调度节点可能宕机的问题。

在一种可实施的方式中，若上述 Actor 模型中只包括一个备用调度节点，则将在调度节点宕机的时候，直接将上述 Actor 模型中仅包括的一个备用调度节点作为新的调度节点，若上述 Actor 模型中包括多个备用调度节点，则在调度节点宕机的时候，从该多个备用调度节点中选取任意一个备用调度节点作为新的调度节点，或者，分别获取每个备用调度节点的创建时间，并确定 Actor 模型包括的备用调度节点中创建时间最早的调度节点，然后将创建时间最早的备用调度节点作为新的调度节点。

可见，如果是根据备用调度时间的早晚来确定用于替换宕机的调度节点的备用调度节点，好处在于，便于本端装置通过读取当前使用的调度节点的创建时间，可以快速判定已宕机的调用节点的个数，从而获知调用节点的宕机情况。

在一种可实施的方式中，上述为目标任务创建 Actor 模型指的是，获取目标任务的标签，标签用于指示目标任务的重要程度，并获取目标任务的标签对应的目标数量，目标数量用于指示 Actor 模型中的备用调度节点的个数，然后创建包括目标数量的备用调度节点的 Actor 模型。

在本申请实施例中，在创建 Actor 模型的时候，Actor 模型中的备用调度节点的个数多少与目标任务的重要程度有关，目标任务越重要，就为目标任务创建更多的备用调度节点。具体的，在为目标任务创建 Actor 模型的时候，获取目标任务的标签，并在数据库中获取该目标任务的标签对应的目标数量，然后创建包含目标数量的备用调度节点。其中，数据库中保存有任务的标签与备用调度节点的个数的对应关系。

在另一种可实施的方式中，上述多个工作节点还包括至少一个监控节点，监控节点用于初始化宕机的调度节点，具体的，若检测到调度节点宕机，则利用监控节点对该宕机的调度节点进行初始化，使得该宕机的调度节点恢复正常。

本申请实施例描述了另一种解决调度节点宕机问题的方法，即如果调度节点宕机，则使用该监控设备对宕机的调度节点进行初始化。具体的，在检测到调度节点宕机之后，使用监控节点根据调度节点的初始化信息对宕机的调度节点进行初始化，使得宕机的调度节点恢复正常。或者，监控节点用于将宕机的调度节点恢复到宕机前一个时刻的正常状态，具体的，在检测到上述调度节点宕机之后，获取记录在数据库中的该宕机的调度节点在宕机前一个时刻的状态信息，并使用监控节点根据上述调度节点在宕机前一个时刻的状态信息对调度节点进行恢复，使得调度节点恢复到上述状态信息对应的正常状态。

在另一种可实施的方式中，在为目标任务创建 Actor 模型的时候，根据目标任务的重要程度决定是否在上述 Actor 模型中创建监控节点。具体的，在任务集群中检测到目标任务之后，获取该目标任务的标签，该标签用于指示目标任务的重要程度。获取目标任务的标签对应的目标任务的重要等级，若目标任务的重要等级超过了预设等级，则在创建 Actor 模型的时候，创建上述监控节点，若目标任务的重要等级未超过预设等级，则不创建上述监控节点。

在另一种可实施的方式中，上述监控节点还用于对宕机的任务节点进行初始化，以使得该宕机的任务节点恢复正常。

在另一种可实施的方式中，上述为目标任务创建的 Actor 模型针对于该模型中包含的每个任务节点分别包括至少一个备用任务节点，该备用任务节点用于替换宕机的任务节点。也即是说 Actor 模型中的一个任务节点对应一个备用任务节点，任务节点与备用任务节点是一一对应的。具体的，本方案还包括：检测宕机的任务节点；若检测到宕机的任务节点，则从 Actor 模型中的至少一个备用任务节点中选取与宕机的任务节点对应的备用任务节点，并将该选出的备用任务节点作为新的任务节点。

206：在检测到上述目标任务完成之后，在上述任务集群中删除上述目标任务。

在本申请实施例，由于调度节点可以感知到目标任务是否完成，于是本端装置可以检测到上述目标任务是否已完成，若在检测到上述目标任务完成，则在上述任务集群中删除上述目标任务，并检测和执行任务集群中的下一个目标任务。

207：检测已创建的工作节点的个数，并获取该已创建的工作节点的创建时间。

在本申请实施例，在上述目标任务完成，检测上述已创建的工作节点，统计工作节点的个数，以及每个已创建的工作节点的创建时间。基于创建时间从早到晚的顺序，对已创建的工作节点进行排序。

208：若上述已创建的工作节点的个数大于预设个数，则按照上述创建时间从早到晚的排序，依次删除已创建的工作节点，直到剩下的已创建的工作节点的个数小于或等于预设个数。

在本申请实施例，系统中的工作节点可能太多而导致本端装置存储空间不足或者系统运行缓慢，于是在完成目标任务之后，检测已创建的工作节点的个数，如果已创建的工作节点的个数超过预设个数，就对工作节点进行删减，直到剩下的工作节点的个数小于等于预设个数。其中，按照工作节点的创建时间由早到晚顺序进行删除，这是因为早创建的工作节点，相较于晚创建的工作节点，被再次使用的几率更小。

上述删除已创建的工作节点的详细过程为，检测已创建的工作节点的个数，并获取该已创建的工作节点的创建时间，若该已创建的工作节点的个数大于预设个数，则确定该已创建的工作节点中创建时间最早的时间节点，并删除所述已创建的工作节点中创建时间最早的工作节点，然后若在删除该已创建的工作节点中创建时间最晚的工作节点之后，已创建的工作节点的个数仍大于预设个数，则触发执行上述确定已创建的工作节点中创建时间

最早的工作节点，至删除已创建的工作节点中创建时间最早的工作节点的步骤，直到已创建的工作节点的个数在删除了创建时间最晚的工作节点之后小于或等于预设个数。

本申请实施例相对于上一申请实施例来说更加的详细，且详细描述了在利用调度节点调用任务节点完成目标任务的过程中，若检测到调度节点宕机，则从 Actor 模型中包括的至少一个备用调度节点中选取一个备用调度节点，替换宕机的调度节点，作为新的调度节点，此外，还详细描述了在目标任务完成之后，若已创建的工作节点超过预设个数，就对多余的工作节点进行清除，清除创建时间较早的工作节点，使得删除之后已创建的工作节点的个数小于等于预设个数。因此本申请实施例一方面在调度节点宕机时，利用备用调度节点来作为新的调度节点，使得任务调度的过程更加可靠，也增加了任务调度的成功率，另一方面，本申请实施例通过在目标任务完成之后及时得对多余的工作节点进行清除，来保障了本装置的系统的顺畅运行，并且保障了下一个目标任务的调度和处理。总的来说任务调度的效率进一步被提高了。

需要说明的是，上文对各个实施例的描述倾向于强调各个实施例之间的不同之处，其相同或相似之处可以互相参考，为了简洁，本文不再赘述。

本申请实施例还提供一种任务调度装置，该任务调度装置用于执行前述任一项的任务调度方法的单元。具体地，参见图 3，是本申请实施例提供的一种任务调度装置的示意框图。本实施例的任务调度装置包括：检测单元 310、创建单元 320、执行单元 330 以及删除单元 340。具体的：

检测单元 310，用于检测任务集群中是否存在新增的目标任务；

创建单元 320，用于在检测到任务集群中存在新增的目标任务的情况下，为上述目标任务创建 Actor 模型，上述 Actor 模型包括多个工作节点，上述多个工作节点之间通过消息传递的方式通信，上述多个工作节点中至少存在两个工作节点分别分布于不同的终端上，上述多个工作节点包括一个调度节点和至少一个任务节点，上述调度节点用于调用上述任务节点执行任务；

执行单元 330，用于利用上述调度节点通过上述消息传递的方式调用上述至少一个任务节点执行上述目标任务；

删除单元 340，用于在检测到上述目标任务完成之后，在上述任务集群中删除上述目标任务。

在另一种可实施的方式中，上述多个工作节还包括至少一个备用调度节点，上述备用调度节点用于替换宕机的调度节点，上述检测单元 310，还用于检测上述调度节点是否宕机；上述任务调度装置还包括替换单元 350，用于从上述至少一个备用调度节点中选取一个备用调度节点作为新的调度节点。

在另一种可实施的方式中，上述任务调度方式还包括获取单元 370，用于分别获取上述至少一个备用调度节点的创建时间；还包括确定单元 360，用于确定上述至少一个备用

调度节点中创建时间最早的调度节点；上述替换单元 350，具体用于将上述创建时间最早的备用调度节点作为上述新的调度节点。

在另一种可实施的方式中，上述获取单元 370，还用于获取上述目标任务的标签，上述标签用于指示上述目标任务的重要程度；上述获取单元 370，还用于获取上述目标任务的标签对应的目标数量，上述目标数量用于指示上述 Actor 模型中的备用调度节点的个数；上述创建单元 320，还用于创建包括上述目标数量的备用调度节点的 Actor 模型。

在另一种可实施的方式中，上述多个工作节点还包括至少一个监控节点，上述监控节点用于初始化宕机的调度节点，上述检测单元 310，还用于检测上述调度节点宕机；上述任务调度装置还包括修复单元 380，用于利用上述监控节点对上述宕机的调度节点进行初始化，使得上述宕机的调度节点恢复正常。

在另一种可实施的方式中，上述检测单元 310，用于检测已创建的工作节点的个数；上述获取单元 370，用于获取上述已创建的工作节点的创建时间；上述删除单元 340，还用于若上述已创建的工作节点的个数大于预设个数，则按照上述创建时间从早到晚的排序，依次删除上述已创建的工作节点，直到剩下的已创建的工作节点的个数小于或等于上述预设个数。

在另一种可实施的方式中，上述检测单元 310，具体用于检测上述任务集群中的新增的任务；上述检测单元 310，还具体用于若确定上述新增的任务为分布式任务，则确定检测到上述任务集群中存在新增的目标任务。

在本申请实施例，每当检测单元检测到目标任务，创建单元就为该目标任务建立包括调度节点和任务节点的 Actor 模型，然后执行单元利用调度节点来调度任务节点执行目标任务，在检测单元检测到目标任务完成之后，删除单元删除该目标任务，可见，本申请所提供的任务调度方法中，由于调度节点和任务节点都是 Actor，于是调度节点和任务节点是通过信息传递的方式进行通信的，而信息传递的好处就是不会产生数据竞争的问题，在处理任务的时候无需同步阻塞等待，于是本申请也便不需要借助于锁便可以实现复杂的分布式任务调度，并且大大提高分布式任务处理的并发性能。可见本申请实施例提供了一种高效的任务调度方法。

参见图 4，是本申请另一实施例提供的一种任务调度装置示意框图。如图所示的本实施例中的任务调度装置可以包括：一个或多个处理器 410 和存储器 420。上述处理器 410 和存储器 420 通过总线 430 连接。存储器 420 用于存储计算机程序，计算机程序包括程序指令，处理器 410 用于执行存储器 420 存储的程序指令。具体的：

处理器 410，用于执行检测单元 310 的功能，用于检测任务集群中是否存在新增的目标任务；还用于执行创建单元 320 的功能，用于在检测到任务集群中存在新增的目标任务的情况下，为上述目标任务创建 Actor 模型，上述 Actor 模型包括多个工作节点，上述多个工作节点之间通过消息传递的方式通信，上述多个工作节点中至少存在两个工作节点分别

分布于不同的终端上，上述多个工作节点包括一个调度节点和至少一个任务节点，上述调度节点用于调用上述任务节点执行任务；还用于执行执行单元 330 的功能，用于利用上述调度节点通过上述消息传递的方式调用上述至少一个任务节点执行上述目标任务；还用于执行删除单元 340 的功能，用于在检测到上述目标任务完成之后，在上述任务集群中删除上述目标任务。

在另一种可实施的方式中，上述多个工作节点还包括至少一个备用调度节点，上述备用调度节点用于替换宕机的调度节点，上述处理器 410，还用于检测上述调度节点是否宕机；处理器 410 还用于执行替换单元 350 的功能，用于从上述至少一个备用调度节点中选取一个备用调度节点作为新的调度节点。

在另一种可实施的方式中，上述处理器 410 还用于执行获取单元 370 的功能，用于分别获取上述至少一个备用调度节点的创建时间；上述处理器 410 还用于执行确定单元 360 的功能，用于确定上述至少一个备用调度节点中创建时间最早的调度节点；上述处理器 410 还具体用于将上述创建时间最早的备用调度节点作为上述新的调度节点。

在另一种可实施的方式中，上述处理器 410 还用于获取上述目标任务的标签，上述标签用于指示上述目标任务的重要程度；还用于获取上述目标任务的标签对应的目标数量，上述目标数量用于指示上述 Actor 模型中的备用调度节点的个数；还用于创建包括上述目标数量的备用调度节点的 Actor 模型。

在另一种可实施的方式中，上述多个工作节点还包括至少一个监控节点，上述监控节点用于初始化宕机的调度节点，上述处理器 410 还用于检测上述调度节点宕机；上述处理器 410 还用于执行修复单元 380 的功能，用于利用上述监控节点对上述宕机的调度节点进行初始化，使得上述宕机的调度节点恢复正常。

在另一种可实施的方式中，上述处理器 410 还用于检测已创建的工作节点的个数；还用于获取上述已创建的工作节点的创建时间；还用于若上述已创建的工作节点的个数大于预设个数，则按照上述创建时间从早到晚的排序，依次删除上述已创建的工作节点，直到剩下的已创建的工作节点的个数小于或等于上述预设个数。

在另一种可实施的方式中，上述处理器 410，具体用于检测上述任务集群中的新增的任务；还具体用于若确定上述新增的任务为分布式任务，则确定检测到上述任务集群中存在新增的目标任务。

应当理解，在本申请实施例中，所称处理器 410 可以是中央处理单元（Central Processing Unit, CPU），该处理器还可以是其他通用处理器、数字信号处理器（Digital Signal Processor, DSP）、专用集成电路（Application Specific Integrated Circuit, ASIC）、现成可编程门阵列（Field-Programmable Gate Array, FPGA）或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件等。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

该存储器 420 可以包括只读存储器和随机存取存储器，并向处理器 410 提供指令和数

据。存储器 420 的一部分还可以包括非易失性随机存取存储器。例如，存储器 420 还可以存储设备类型的信息。

具体实现中，本申请实施例中所描述的处理器 410 可执行本申请实施例提供的任务调度方法的第一实施例和第二实施例中所描述的实现方式，也可执行本申请实施例所描述的

5 任务调度装置的实现方式，在此不再赘述。

在本申请的另一实施例中提供一种计算机可读存储介质，计算机可读存储介质存储有计算机程序，计算机程序包括程序指令，程序指令被处理器执行。

计算机可读存储介质可以是前述任一实施例的任务调度装置的内部存储单元，例如任务调度装置的硬盘或内存。计算机可读存储介质也可以是任务调度装置的外部存储设备，
10 例如任务调度装置上配备的插接式硬盘，智能存储卡（Smart Media Card, SMC），安全数字（Secure Digital, SD）卡，闪存卡（Flash Card）等。进一步地，计算机可读存储介质还可以既包括任务调度装置的内部存储单元也包括外部存储设备。计算机可读存储介质用于存储计算机程序以及任务调度装置所需的其他程序和数据。计算机可读存储介质还可以用于暂时地存储已经输出或者将要输出的数据。

15

权 利 要 求

1、一种基于 Actor 模型的任务调度方法，其特征在于，包括：

检测任务集群中是否存在新增的目标任务；

5 在检测到所述任务集群中存在新增的目标任务的情况下，为所述目标任务创建 Actor 模型，所述 Actor 模型包括多个工作节点，所述多个工作节点之间通过消息传递的方式通信，所述多个工作节点中至少存在两个工作节点分别分布于不同的终端上，所述多个工作节点包括一个调度节点和至少一个任务节点，所述调度节点用于调用所述任务节点执行任务；

10 利用所述调度节点通过所述消息传递的方式调用所述至少一个任务节点执行所述目标任务；

在检测到所述目标任务完成之后，在所述任务集群中删除所述目标任务。

2、根据权利要求 1 所述的方法，其特征在于，所述多个工作节点还包括至少一个备用调度节点，所述备用调度节点用于替换宕机的调度节点，所述方法还包括：

15 若检测到所述调度节点宕机，则从所述至少一个备用调度节点中选取一个备用调度节点作为新的调度节点。

3、根据权利要求 2 所述的方法，其特征在于，所述从所述至少一个备用调度节点中选取一个备用调度节点作为新的调度节点，包括：

20 分别获取所述至少一个备用调度节点的创建时间，并确定所述至少一个备用调度节点中创建时间最早的调度节点；

将所述创建时间最早的备用调度节点作为所述新的调度节点。

4、根据权利要求 2 所述的方法，其特征在于，所述为所述目标任务创建 Actor 模型，包括：

获取所述目标任务的标签，所述标签用于指示所述目标任务的重要程度；

25 获取所述目标任务的标签对应的目标数量，所述目标数量用于指示所述 Actor 模型中的备用调度节点的个数；

创建包括所述目标数量的备用调度节点的 Actor 模型。

5、根据权利要求 1 所述的方法，其特征在于，所述多个工作节点还包括至少一个监控节点，所述监控节点用于初始化宕机的调度节点，所述方法还包括：

30 若检测到所述调度节点宕机，则利用所述监控节点对所述宕机的调度节点进行初始化，使得所述宕机的调度节点恢复正常。

6、根据权利要求 1 所述的方法，其特征在于，所述在所述任务集群中删除所述目标任务之后，还包括：

检测已创建的工作节点的个数，并获取所述已创建的工作节点的创建时间；

35 若所述已创建的工作节点的个数大于预设个数，则按照所述创建时间从早到晚的排序，

依次删除所述已创建的工作节点，直到剩下的已创建的工作节点的个数小于或等于所述预设个数。

7、根据权利要求1所述的方法，其特征在于，所述检测任务集群中是否存在新增的目标任务，包括：

5 检测所述任务集群中的新增的任务；

若确定所述新增的任务为分布式任务，则确定检测到所述任务集群中存在新增的目标任务。

8、根据权利要求1所述的方法，其特征在于，所述在检测到所述任务集群中存在新增的目标任务的情况下，所述方法还包括：

10 获取所述目标任务的标签，所述标签用于指示所述目标任务的重要程度；

获取所述目标任务的标签对应所述目标任务的重要等级；

若所述目标任务的重要等级超过了预设等级，则创建包括有至少一个监控节点的 Actor 模型，所述监控节点用于初始化宕机的调度节点。

9、根据权利要求1所述的方法，其特征在于，所述多个工作节还包括与所述至少一个任务节点分别对应的至少一个备用任务节点，所述备用任务节点用于替换宕机的任务节点，所述方法还包括：

检测宕机的任务节点；

20 若检测到宕机的任务节点，则从所述至少一个备用任务节点中选取与宕机的任务节点对应的备用任务节点，并将所述与宕机的任务节点对应的备用任务节点作为新的任务节点。

10、一种基于 Actor 模型的任务调度装置，其特征在于，包括：

检测单元，用于检测任务集群中是否存在新增的目标任务；

25 创建单元，用于在检测到任务集群中存在新增的目标任务的情况下，为所述目标任务创建 Actor 模型，所述 Actor 模型包括多个工作节点，所述多个工作节点之间通过消息传递的方式通信，所述多个工作节点中至少存在两个工作节点分别分布于不同的终端上，所述多个工作节点包括一个调度节点和至少一个任务节点，所述调度节点用于调用所述任务节点执行任务；

执行单元，用于利用所述调度节点通过所述消息传递的方式调用所述至少一个任务节点执行所述目标任务；

30 删除单元，用于在检测到所述目标任务完成之后，在所述任务集群中删除所述目标任务。

11、一种任务调度装置，其特征在于，包括处理器和存储器，所述处理器和存储器相互连接；所述存储器用于存储计算机程序，所述计算机程序包括程序指令；所述处理器被配置用于调用所述程序指令，用以执行：

35 检测任务集群中是否存在新增的目标任务；

在检测到所述任务集群中存在新增的目标任务的情况下，为所述目标任务创建 Actor 模型，所述 Actor 模型包括多个工作节点，所述多个工作节点之间通过消息传递的方式通信，所述多个工作节点中至少存在两个工作节点分别分布于不同的终端上，所述多个工作节点包括一个调度节点和至少一个任务节点，所述调度节点用于调用所述任务节点执行任务；

利用所述调度节点通过所述消息传递的方式调用所述至少一个任务节点执行所述目标任务；

在检测到所述目标任务完成之后，在所述任务集群中删除所述目标任务。

12、根据权利要求 11 所述的装置，其特征在于，所述多个工作节点还包括至少一个备用调度节点，所述备用调度节点用于替换宕机的调度节点，所述处理器还用于：

若检测到所述调度节点宕机，则从所述至少一个备用调度节点中选取一个备用调度节点作为新的调度节点。

13、根据权利要求 12 所述的装置，其特征在于，所述处理器具体用于：

分别获取所述至少一个备用调度节点的创建时间，并确定所述至少一个备用调度节点中创建时间最早的调度节点；

将所述创建时间最早的备用调度节点作为所述新的调度节点。

14、根据权利要求 12 所述的装置，其特征在于，所述处理器具体用于：

获取所述目标任务的标签，所述标签用于指示所述目标任务的重要程度；

获取所述目标任务的标签对应的目标数量，所述目标数量用于指示所述 Actor 模型中的备用调度节点的个数；

创建包括所述目标数量的备用调度节点的 Actor 模型。

15、根据权利要求 11 所述的装置，其特征在于，所述多个工作节点还包括至少一个监控节点，所述监控节点用于初始化宕机的调度节点，所述处理器还用于：

若检测到所述调度节点宕机，则利用所述监控节点对所述宕机的调度节点进行初始化，使得所述宕机的调度节点恢复正常。

16、根据权利要求 1 所述的装置，其特征在于，所述处理器还用于：

检测已创建的工作节点的个数，并获取所述已创建的工作节点的创建时间；

若所述已创建的工作节点的个数大于预设个数，则按照所述创建时间从早到晚的排序，依次删除所述已创建的工作节点，直到剩下的已创建的工作节点的个数小于或等于所述预设个数。

17、根据权利要求 1 所述的装置，其特征在于，所述处理器具体用于：

检测所述任务集群中的新增的任务；

若确定所述新增的任务为分布式任务，则确定检测到所述任务集群中存在新增的目标任务。

18、根据权利要求 11 所述的装置，其特征在于，所述处理器还用于：

获取所述目标任务的标签，所述标签用于指示所述目标任务的重要程度；

获取所述目标任务的标签对应所述目标任务的重要等级；

若所述目标任务的重要等级超过了预设等级，则创建包括有至少一个监控节点的 Actor 模型，所述监控节点用于初始化宕机的调度节点。

5 19、根据权利要求 11 所述的装置，其特征在于，所述多个工作节还包括与所述至少一个任务节点分别对应的至少一个备用任务节点，所述备用任务节点用于替换宕机的任务节点，所述处理器还用于：

检测宕机的任务节点；

10 若检测到宕机的任务节点，则从所述至少一个备用任务节点中选取与与宕机的任务节点对应的备用任务节点，并将所述与宕机的任务节点对应的备用任务节点作为新的任务节点。

20、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存储有计算机程序，所述计算机程序包括程序指令，所述程序指令当被处理器执行时使所述处理器执行如权利要求 1-9 任一项所述的方法。

15

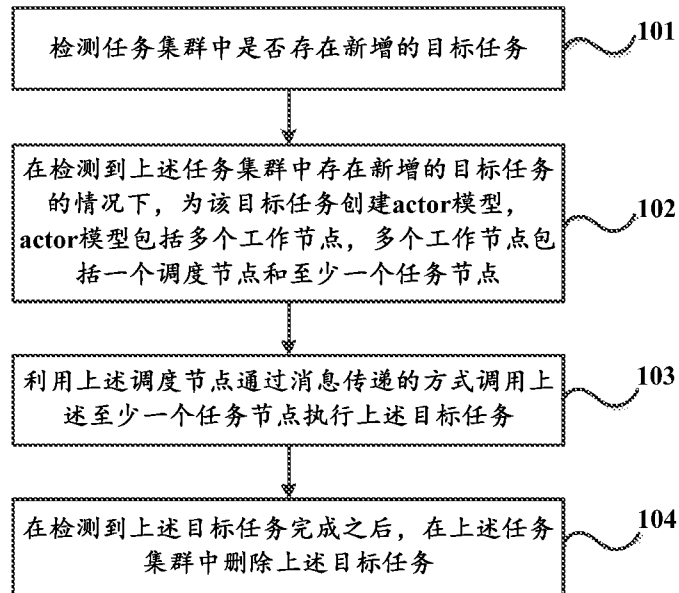


图 1

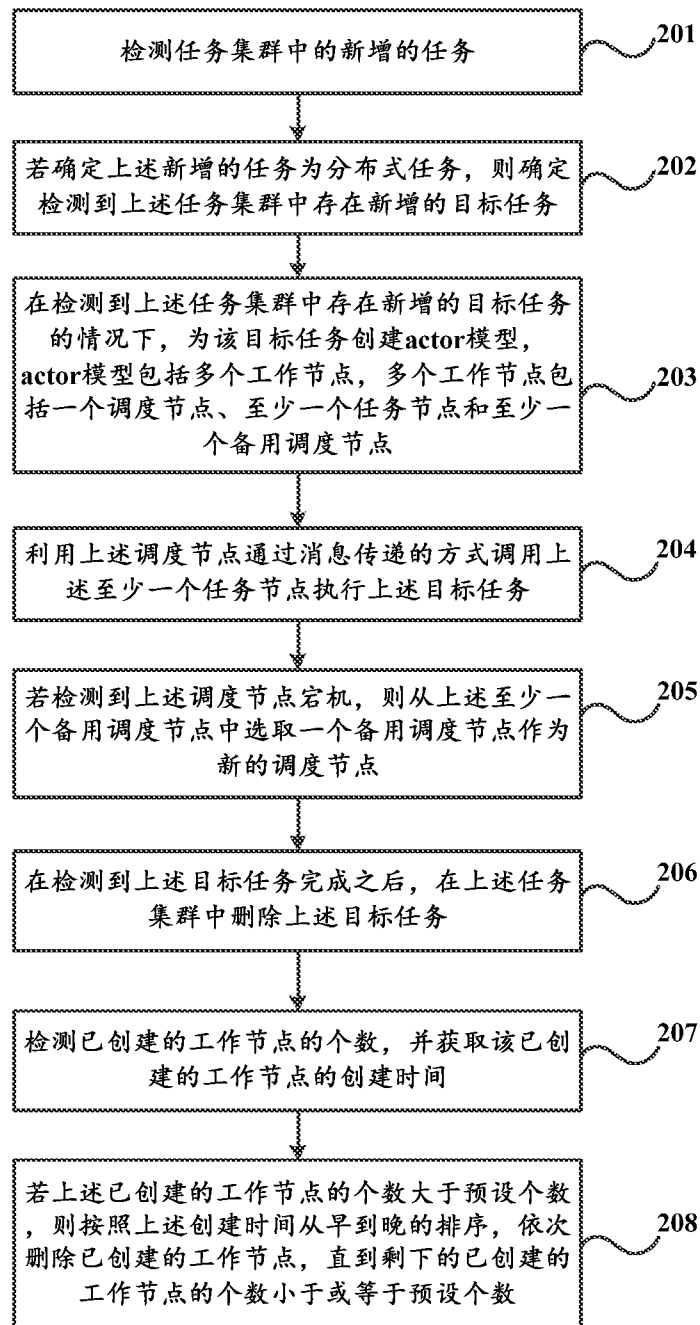


图 2

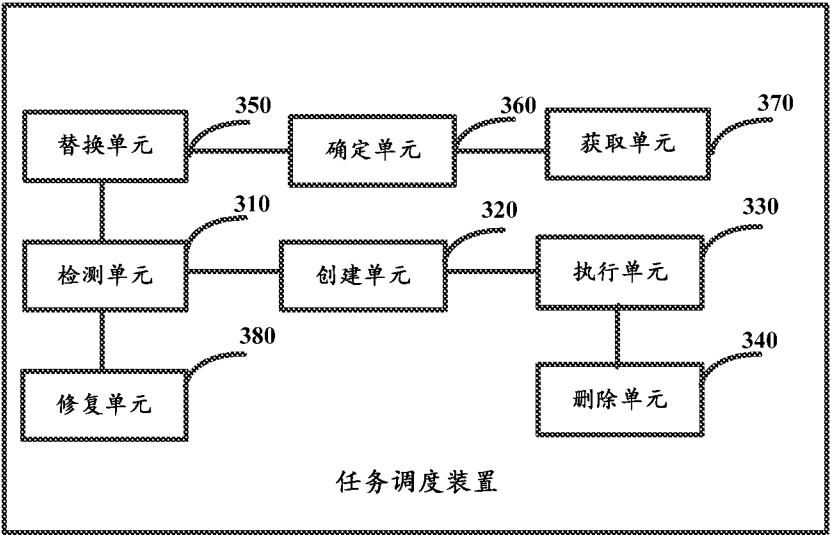


图 3

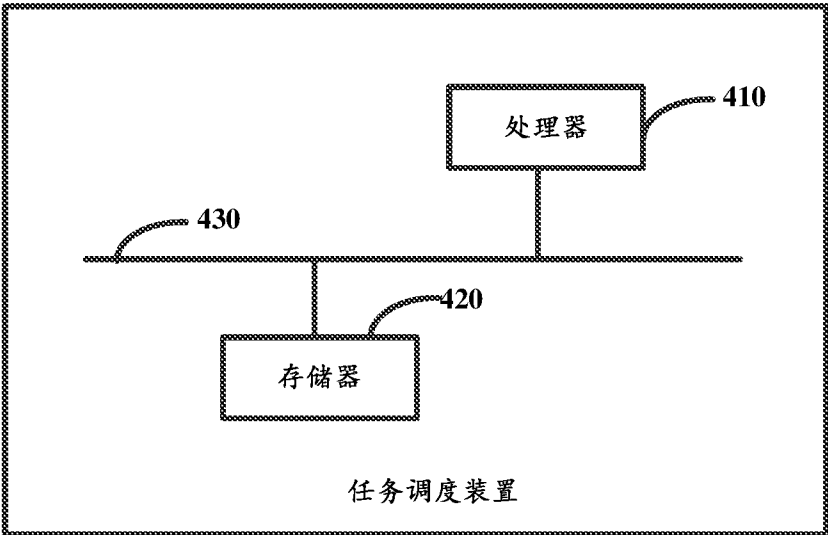


图 4

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2019/102741

A. CLASSIFICATION OF SUBJECT MATTER

G06F 9/52(2006.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNABS, CNTXT, CNKI, VEN, USTXT, EPTXT, WOTXT: 角色, 任务, 调度, 节点, 备用, 检测, actor, task, dispatch, schedule, node, reserve, detect

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	陈昊 等 (CHEN, Hao et al.). "基于Actor模型的高性能分布式XMPP服务器 (High-performance distributed XMPP server based on actor model)" 计算机系统应用 (Computer Systems & Applications), Vol. 24, No. 10, 15 October 2015 (2015-10-15), ISSN: 1003-3254, sections 2-5	1-4, 6-7, 9-14, 16-17, 19-20
A	CN 106406983 A (ALIBABA GROUP HOLDING LIMITED) 15 February 2017 (2017-02-15) description, paragraphs 9-25	1-20
A	US 2018077520 A1 (WISILICA INC) 15 March 2018 (2018-03-15) entire document	1-20



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

06 February 2020

Date of mailing of the international search report

21 February 2020

Name and mailing address of the ISA/CN

China National Intellectual Property Administration (ISA/
CN)
No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing
100088
China

Authorized officer

Facsimile No. (86-10)62019451

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2019/102741

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)		Publication date (day/month/year)
CN	106406983	A	15 February 2017	None		
US	2018077520	A1	15 March 2018	US	10313861 B2	04 June 2019

国际检索报告

国际申请号

PCT/CN2019/102741

A. 主题的分类

G06F 9/52 (2006.01) i

按照国际专利分类 (IPC) 或者同时按照国家分类和 IPC 两种分类

B. 检索领域

检索的最低限度文献 (标明分类系统和分类号)

G06F

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库 (数据库的名称, 和使用的检索词 (如使用))

CNABS, CNTXT, CNKI, VEN, USTXT, EPTXT, WOTXT: 角色, 任务, 调度, 节点, 备用, 检测, actor, task, dispatch, schedule, node, reserve, detect

C. 相关文件

类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求
X	陈吴 等. “基于Actor模型的高性能分布式XMP服务器” 计算机系统应用, 第24卷, 第10期, 2015年 10月 15日 (2015 - 10 - 15), ISSN: 1003-3254, 第2-5节	1-4、6-7、9-14、 16-17、19-20
A	CN 106406983 A (阿里巴巴集团控股有限公司) 2017年 2月 15日 (2017 - 02 - 15) 说明书第9-25段	1-20
A	US 2018077520 A1 (WISILICA INC) 2018年 3月 15日 (2018 - 03 - 15) 全文	1-20

☐ 其余文件在C栏的续页中列出。☒ 见同族专利附件。

* 引用文件的具体类型:

“A” 认为不特别相关的表示了现有技术一般状态的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的)

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期

2020年 2月 6日

国际检索报告邮寄日期

2020年 2月 21日

ISA/CN的名称和邮寄地址

中国国家知识产权局 (ISA/CN)

中国北京市海淀区蓟门桥西土城路6号 100088

传真号 (86-10)62019451

授权官员

刘细金

电话号码 86-(20)-28950395

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2019/102741

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	106406983	A	2017年 2月 15日	无			
US	2018077520	A1	2018年 3月 15日	US	10313861	B2	2019年 6月 4日