(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 110609749 B
(45)授权公告日 2023.07.14

(21)申请号 201910841793.2

(22)申请日 2019.09.06

(65)同一申请的已公布的文献号
申请公布号 CN 110609749 A

(43)申请公布日 2019.12.24

(73)专利权人 创新先进技术有限公司
地址 开曼群岛大开曼岛乔治镇医院路27号开曼企业中心邮编KY1-9008

(72)发明人 王益 严伟 唐源 章海涛 文春阳 李明昊 齐俊 刘勇峰

(74)专利代理机构 北京晋德允升知识产权代理有限公司 11623
专利代理师 王戈

(51)Int.Cl.
*G06F 9/50*(2006.01)

(56)对比文件
CN 105022662 A,2015.11.04
CN 107153573 A,2017.09.12
CN 110032444 A,2019.07.19
CN 107343045 A,2017.11.10
US 10031785 B2,2018.07.24

审查员 周真

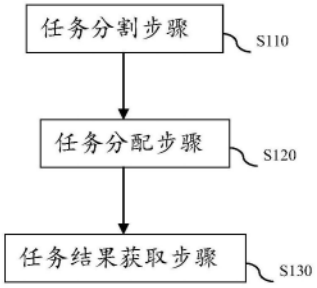
权利要求书3页 说明书13页 附图4页

(54)发明名称

一种分布式任务运行方法、系统及设备

(57)摘要

本申请公开了一种分布式任务运行方法、系统及设备。本说明书一实施例的方法流程包括：获取任务分片，将任务分片分配给有效计算节点进行处理，获取任务结果。在分配任务分片的过程中：每个有效计算节点同一时间仅能分配一个任务分片，有效计算节点被分配任务分片后即开始运行任务分片，当有效计算节点完成其被分配的任务分片时，有效计算节点可被分配新的任务分片；当有效计算节点出错时，将出错的有效计算节点当前所分配到的任务分片重新分配；当有效计算节点被关闭或挪用时，将被关闭或挪用的有效计算节点当前所分配到的任务分片重新分配；当新的有效计算节点被拉起时，将未分配的任务分片分配给新的有效计算节点。




CN 110609749 B

1.一种分布式任务运行方法,所述方法应用于深度学习框架,所述方法包括:

任务分割步骤,分割待处理任务,获取多个任务分片,其中,每个任务分片为一个独立的任务;所述待处理任务包括深度学习作业的任务;

任务分配步骤,将所述任务分片分配给有效计算节点进行处理,所述计算节点用于调用所述任务分片对应的运算方法对所述任务分片对应的训练数据进行数据计算,其中:

将所述多个任务分片中的至少部分任务分片分配给当前计算资源所支撑的有效计算节点,每个所述有效计算节点同一时间仅能分配一个任务分片,所述有效计算节点被分配任务分片后即开始运行所述任务分片,被分配任务分片的各个所述有效计算节点并行运行,当所述有效计算节点完成其被分配的任务分片时,所述有效计算节点可被分配新的任务分片;

若当前分布式任务的可用计算资源减少,具体包括:当所述有效计算节点出错时,将出错的有效计算节点当前所分配到的任务分片重新分配;当所述有效计算节点被关闭或挪用时,将被关闭或挪用的有效计算节点当前所分配到的任务分片重新分配;

若当前分布式任务的可用计算资源增加,则拉起新的有效计算节点,当所述新的有效计算节点被拉起时,将未分配的任务分片分配给所述新的有效计算节点;

任务结果获取步骤,当任一有效计算节点完成一个任务分片时,获取对应的任务分片处理结果,综合任务分片对应的任务分片处理结果以获取所述待处理任务对应的任务处理结果。

2.根据权利要求1所述的方法,在所述任务分割步骤中:

分割所述待处理任务,其中,令每个所述任务分片满足预设数据量。

3.根据权利要求2所述的方法:

根据历史信息来确定所述预设数据量;

或者,

由用户自定义设置所述预设数据量。

4.根据权利要求1所述的方法,所述方法还包括:

根据所述任务分割步骤的处理结果拉起有效计算节点,其中,确认基于可用计算资源能够拉起的有效计算节点数是否满足任务分片数,如果满足,拉起对应任务分片数的有效计算节点,如果不满足,利用所有可用计算资源拉起有效计算节点。

5.根据权利要求1所述的方法,所述方法还包括:

在存在未使用的可用计算资源且存在未分配任务分片时,拉起新的有效计算节点,其中:

监控可用计算资源变化,当发现新的可用计算资源且当前存在未分配任务分片时,拉起新的有效计算节点;

和/或,

当所述有效计算节点出错导致存在需要重新分配的任务分片且当前存在可用计算资源时,拉起新的有效计算节点。

6.根据权利要求4所述的方法,所述方法还包括:

利用可用计算资源拉起有效计算节点,其中,拉起的有效节点总数不超过用户指定数量。

7.根据权利要求1所述的方法,所述方法还包括:

在所述有效计算节点运行所述任务分片的过程中,保存所述有效计算节点的阶段性运行结果。

8.根据权利要求7所述的方法,所述方法还包括:

在所述有效计算节点运行所述任务分片的过程中,检测所述阶段性运行结果。

9.根据权利要求1所述的方法,所述方法还包括:

当存在所述待处理任务时,利用可用计算资源拉起主节点以及所述有效计算节点,所述主节点用于执行所述任务分割步骤以及所述任务分配步骤。

10.根据权利要求9所述的方法,所述主节点还用于拉起所述有效计算节点。

11.根据权利要求9所述的方法,所述主节点还用于:

执行所述任务结果获取步骤;

和/或,

获取并保存所述有效计算节点的阶段性运行结果;

和/或,

创建展示页面,所述展示页面用于展示当前的任务处理进度状态。

12.根据权利要求9所述的方法,当存在所述待处理任务时,利用可用计算资源拉起主节点,其中:

由客户端采集用户输入的所述待处理任务;

当所述客户端采集到所述待处理任务时,由所述客户端发送请求令所述可用计算资源对应的计算集群装置拉起所述主节点,并且,由所述客户端向所述计算集群装置发送所述待处理任务。

13.根据权利要求12所述的方法,由所述客户端发送请求令所述可用计算资源对应的计算集群装置拉起所述主节点,并且,由所述客户端向所述计算集群装置发送所述待处理任务,包括:

由所述客户端发送所述待处理任务到所述可用计算资源对应的计算集群装置;

由所述计算集群装置验证所述待处理任务,验证通过后由所述计算集群装置拉起所述主节点。

14.根据权利要求13所述的方法:

当所述客户端采集到所述待处理任务后,建立所述客户端到所述计算集群装置间的数据链接;

由所述计算集群装置验证所述待处理任务,验证通过后断开所述客户端到所述计算集群装置间的数据链接。

15.根据权利要求1所述的方法,在所述任务结果获取步骤中,综合任务分片对应的任务分片处理结果以获取所述待处理任务对应的任务处理结果,其中,每当获取到预设数量的任务分片处理结果后汇总产生任务中间结果。

16.根据权利要求1所述的方法:

所述待处理任务包括深度学习作业所需要的训练数据及模型计算信息;

所述任务分片包括对应的训练数据以及运算方法。

17.根据权利要求16所述的方法,所述方法还包括:

基于用户指定的轮次，对所述待处理任务进行多轮分布式计算。

18.一种分布式任务运行装置，所述装置应用于深度学习框架，所述装置包括：

任务分割单元，其用于分割待处理任务，获取多个任务分片，其中，每个任务分片为一个独立的任务；所述待处理任务包括深度学习作业的任务；

任务分配单元，将所述任务分片分配给计算节点进行处理，所述计算节点用于调用所述任务分片对应的运算方法对所述任务分片对应的训练数据进行数据计算，其中：

将所述多个任务分片中的至少部分任务分片分配给当前计算资源所支撑的有效计算节点，每个所述有效计算节点同一时间仅能分配一个任务分片，所述有效计算节点被分配任务分片后即开始运行所述任务分片，被分配任务分片的各个所述有效计算节点并行运行，当所述有效计算节点完成其被分配的任务分片时，所述有效计算节点可被分配新的任务分片；

若当前分布式任务的可用计算资源减少，具体包括：当所述有效计算节点出错时，将出错的有效计算节点当前所分配到的任务分片重新分配；当所述有效计算节点被关闭或挪用时，将被关闭或挪用的有效计算节点当前所分配到的任务分片重新分配；

若当前分布式任务的可用计算资源增加，则拉起新的有效计算节点，当所述新的有效计算节点被拉起时，将未分配的任务分片分配给所述新的有效计算节点；

任务结果获取单元，当任一有效计算节点完成一个任务分片时，获取对应的任务分片处理结果，综合任务分片对应的任务分片处理结果以获取所述待处理任务对应的任务处理结果。

19.一种分布式任务运行系统，所述系统包括：

任务获取模块，其用于获取待处理任务；

分布式系统构建模块，其用于在所述任务获取模块获取到所述待处理任务后，利用可用计算资源建立如权利要求18所述的装置，并向所述装置发送所述待处理任务。

20.一种用于在访问方设备端信息处理的设备，该设备包括用于存储计算机程序指令的存储器和用于执行程序指令的处理器，其中，当该计算机程序指令被该处理器执行时，触发该设备执行权利要求1至17中任一项所述的方法。

# 一种分布式任务运行方法、系统及设备

## 技术领域

[0001] 本说明书涉及计算机技术领域，尤其涉及一种分布式任务运行方法、系统及设备。

## 背景技术

[0002] 分布式计算是一门计算机科学，它研究如何把一个需要非常巨大的计算能力才能解决的问题分成许多小的部分，然后把这些部分分配给许多计算系统进行处理，最后把这些计算结果综合起来得到最终的结果。由于分布式计算在大数据运行处理方面的优势，当前在使用深度学习来训练的时候，用户提交的一个作业通常会选择分布式来执行，用多台机器来协同完成。

[0003] 在一个有多台机器组成的计算集群中，“错误”的发生比想象中更常见，集群中偶发的机器故障或网络故障等，不同任务之间由于优先级不同而发生的“抢占”等，都会导致一个作业的某些任务出错。然而，由于分布式计算的原理是综合所有的分支计算系统的计算结果来获得最终的计算结果，因此，在分布式计算的执行过程中，一个作业的某些任务出错会对整个作业的进程产生严重影响，从而大大影响用于分布式计算的计算资源利用效率。

## 发明内容

[0004] 有鉴于此，本说明书实施例提供了一种分布式任务运行方法、系统及设备，用于解决现有技术中分布式计算执行过程中计算资源利用效率低下的问题。

[0005] 本说明书实施例采用下述技术方案：

[0006] 本说明书实施例提供一种分布式任务运行方法，所述方法包括：

[0007] 任务分割步骤，分割待处理任务，获取多个任务分片，其中，每个任务分片为一个独立的任务；

[0008] 任务分配步骤，将所述任务分片分配给有效计算节点进行处理，其中：

[0009] 每个所述有效计算节点同一时间仅能分配一个任务分片，所述有效计算节点被分配任务分片后即开始运行所述任务分片，当所述有效计算节点完成其被分配的任务分片时，所述有效计算节点可被分配新的任务分片；

[0010] 当所述有效计算节点出错时，将出错的有效计算节点当前所分配到的任务分片重新分配；

[0011] 当所述有效计算节点被关闭或挪用时，将被关闭或挪用的有效计算节点当前所分配到的任务分片重新分配；

[0012] 当新的有效计算节点被拉起时，将未分配的任务分片分配给所述新的有效计算节点；

[0013] 任务结果获取步骤，当任一有效计算节点完成一个任务分片时，获取对应的任务分片处理结果，综合任务分片对应的任务分片处理结果以获取所述待处理任务对应的任务处理结果。

[0014] 在本说明书一实施例中，在所述任务分割步骤中：

[0015] 分割所述待处理任务，其中，令每个所述任务分片满足预设数据量。

[0016] 在本说明书一实施例中：

[0017] 根据历史信息来确定所述预设数据量；

[0018] 或者，

[0019] 由用户自定义设置所述预设数据量。

[0020] 在本说明书一实施例中，所述方法还包括：

[0021] 根据所述任务分割步骤的处理结果拉起有效计算节点，其中，确认基于可用计算资源能够拉起的有效计算节点数是否满足任务分片数，如果满足，拉起对应任务分片数的有效计算节点，如果不满足，利用所有可用计算资源拉起有效计算节点。

[0022] 在本说明书一实施例中，所述方法还包括：

[0023] 在存在未使用的可用计算资源且存在未分配任务分片时，拉起新的有效计算节点，其中：

[0024] 监控可用计算资源变化，当发现新的可用计算资源且当前存在未分配任务分片时，拉起新的有效计算节点；

[0025] 和/或，

[0026] 当所述有效计算节点出错导致存在需要重新分配的任务分片且当前存在可用计算资源时，拉起新的有效计算节点。

[0027] 在本说明书一实施例中，所述方法还包括：

[0028] 利用可用计算资源拉起有效计算节点，其中，拉起的有效节点总数不超过用户指定数量。

[0029] 在本说明书一实施例中，所述方法还包括：

[0030] 在所述有效计算节点运行所述任务分片的过程中，保存所述有效计算节点的阶段性运行结果。

[0031] 在本说明书一实施例中，所述方法还包括：

[0032] 在所述有效计算节点运行所述任务分片的过程中，检测所述阶段性运行结果。

[0033] 在本说明书一实施例中，所述方法还包括：

[0034] 当存在所述待处理任务时，利用可用计算资源拉起主节点以及所述有效计算节点，所述主节点用于执行所述任务分割步骤以及所述任务分配步骤。

[0035] 在本说明书一实施例中，所述主节点还用于拉起所述有效计算节点。

[0036] 在本说明书一实施例中，所述主节点还用于：

[0037] 执行所述任务结果获取步骤；

[0038] 和/或，

[0039] 获取并保存所述有效计算节点的阶段性运行结果；

[0040] 和/或，

[0041] 创建展示页面，所述展示页面用于展示当前的任务处理进度状态。

[0042] 在本说明书一实施例中，当存在所述待处理任务时，利用可用计算资源拉起主节点，其中：

[0043] 由客户端采集用户输入的所述待处理任务；

[0044] 当所述客户端采集到所述待处理任务时，由所述客户端发送请求令所述可用计算资源对应的计算集群装置拉起所述主节点，并且，由所述客户端向所述计算集群装置发送所述待处理任务。

[0045] 在本说明书一实施例中，由所述客户端发送请求令所述可用计算资源对应的计算集群装置拉起所述主节点，并且，由所述客户端向所述计算集群装置发送所述待处理任务，包括：

[0046] 由所述客户端发送所述待处理任务到所述可用计算资源对应的计算集群装置；

[0047] 由所述计算集群装置验证所述待处理任务，验证通过后由所述计算集群装置拉起所述主节点。

[0048] 在本说明书一实施例中：

[0049] 当所述客户端采集到所述待处理任务后，建立所述客户端到所述计算集群装置间的数据链接；

[0050] 由所述计算集群装置验证所述待处理任务，验证通过后断开所述客户端到所述计算集群装置间的数据链接。

[0051] 在本说明书一实施例中，在所述任务结果获取步骤中，综合任务分片对应的任务分片处理结果以获取所述待处理任务对应的任务处理结果，其中，每当获取到预设数量的任务分片处理结果后汇总产生任务中间结果。

[0052] 在本说明书一实施例中：

[0053] 所述待处理任务包括深度学习作业所需要的训练数据及模型计算信息；

[0054] 所述任务分片包括对应的训练数据以及运算方法；

[0055] 所述计算节点用于调用所述任务分片对应的运算方法对所述任务分片对应的训练数据进行数据计算。

[0056] 在本说明书一实施例中，所述方法还包括：

[0057] 基于用户指定的轮次，对所述待处理任务进行多轮分布式计算。

[0058] 本说明书实施例还提出了一种分布式任务运行装置，所述装置包括：

[0059] 任务分割单元，其用于分割待处理任务，获取多个任务分片，其中，每个任务分片为一个独立的任务；

[0060] 任务分配单元，将所述任务分片分配给计算节点进行处理，其中：

[0061] 每个所述有效计算节点同一时间仅能分配一个任务分片，所述有效计算节点被分配任务分片后即开始运行所述任务分片，当所述有效计算节点完成其被分配的任务分片时，所述有效计算节点可被分配新的任务分片；

[0062] 当所述有效计算节点出错时，将出错的有效计算节点当前所分配到的任务分片重新分配；

[0063] 当所述有效计算节点被关闭或挪用时，将被关闭或挪用的有效计算节点当前所分配到的任务分片重新分配；

[0064] 当新的有效计算节点被拉起时，将未分配的任务分片分配给所述新的有效计算节点；

[0065] 任务结果获取单元，当任一有效计算节点完成一个任务分片时，获取对应的任务分片处理结果，综合任务分片对应的任务分片处理结果以获取所述待处理任务对应的任务

处理结果。

[0066] 本说明书实施例还提出了一种分布式任务运行系统，所述系统包括：

[0067] 任务获取模块，其用于获取待处理任务；

[0068] 分布式系统构建模块，其用于在所述任务获取模块获取到所述待处理任务后，利用可用计算资源建立如本说明书一实施例中所述的装置，并向所述装置发送所述待处理任务。

[0069] 本说明书实施例还提出了一种用于在访问方设备端信息处理的设备，该设备包括用于存储计算机程序指令的存储器和用于执行程序指令的处理器，其中，当该计算机程序指令被该处理器执行时，触发该设备执行本说明书实施例所述系统所述的方法。

[0070] 本说明书实施例采用的上述至少一个技术方案能够达到以下有益效果：根据本说明书实施例的方法，可以在避免任务分片的遗漏以及过量分配的前提下，有效利用现有的可用计算资源，确保计算节点的运行效率，避免有效计算节点出现等待而导致计算资源被浪费的情况；根据本说明书实施例的方法，还可以在单个计算节点出错时进行任务分片的重新分配，避免由单个计算节点出错而导致整个任务的重新执行，从而减少计算资源的浪费。

## 附图说明

[0071] 此处所说明的附图用来提供对本申请的进一步理解，构成本申请的一部分，本申请的示意性实施例及其说明用于解释本申请，并不构成对本申请的不当限定。在附图中：

[0072] 图1、图4为本说明书实施例中应用程序的运行方法的流程图；

[0073] 图2、图3为本说明书实施例中应用程序的部分运行方法的流程图；

[0074] 图5为本说明书实施例中模块架构以及数据流示意图；

[0075] 图6为本说明书实施例中分布式任务运行装置的结构框图；

[0076] 图7为本说明书实施例中分布式任务运行系统的结构框图。

## 具体实施方式

[0077] 为使本申请的目的、技术方案和优点更加清楚，下面将结合本申请具体实施例及相应的附图对本申请技术方案进行清楚、完整地描述。显然，所描述的实施例仅是本申请一部分实施例，而不是全部的实施例。基于本申请中的实施例，本领域普通技术人员在没有做出创造性劳动前提下所获得的所有其他实施例，都属于本申请保护的范围。

[0078] 分布式计算执行过程中存在的计算资源利用效率低下的问题，本说明书实施例提出了一种分布式任务运行方法。为了提出本说明书实施例的方法，发明人首先以具体的分布式计算应用场景为对象进行分析。

[0079] 在实际应用场景中，会为分布式计算任务的每一个任务分片(子任务)分配一个计算节点(该计算节点可以是物理意义上的一台计算机，也可以是一个虚拟的计算系统)。理论上，多个计算节点会并行运行多个任务分片，但实际上，各个任务分片可能并没有全都被分配到“有效的”计算节点上。例如，虽然在逻辑上每一个任务分片分配到了一个计算节点，但是在实际中，可用的计算资源并不支持那么多的计算节点，很多计算节点并没有配置对应的计算资源，是无效的；或者，虽然在最初的任务分配环节为每一个任务分片分配到了一

个计算节点并为每一个计算节点分配了对应的计算资源，但是，由于硬件/软件错误等问题，某些计算节点是无法正常运行的，那么这些计算节点在实际上就是无效的。

[0080] 然而，在通常的分布式计算模式下，由于最终的任务处理结果是综合所有任务分片的处理结果而得出的，因此，会要求各个任务分片同步并行执行。例如，Tensorflow是目前一种主流的深度学习框架，它本身支持分布式计算模式。用户在使用Tensorflow的分布式计算模式时，只有所有用于执行分布式计算任务的计算节点都可用时，整个作业才能开始计算。如果由于某些原因，例如计算集群资源不足，导致某些计算节点无法运行时，其它已经拉起(已经创建好、配置好的)的计算节点都会处于等待状态，整个作业也没法开始。

[0081] 基于上述应用场景分析，如果要提高分布式计算执行过程中的计算资源利用效率，一种可行的方法就是为每一个任务分片分配“有效的”计算节点，从而避免由于无效计算节点的存在而带来的有效计算节点空置等待的问题的发生。然而，在一般的应用场景中，计算资源的缺乏往往是常态，再加上无法完全避免计算节点出现错误，因此，很难做到为每一个任务分片都分配有效的计算节点。针对上述问题，在本说明书一实施例中，对并行分布式计算的计算逻辑进行调整，并不采用所有任务分片完全并行执行的逻辑，而是基于现有的计算资源，让当前能够并行执行的一部分任务分片并行执行，让受限于计算资源或计算节点错误等原因不能立即执行的任务分片排在后面，当前一批任务执行完毕后，才使用释放出来的计算资源执行为执行的任务分片。

[0082] 具体的，在本说明书一实施例中，并不是为每个任务分片分配一个计算节点，而是基于当前计算资源可以支撑的有效计算节点(可被分配计算资源并正常运行的计算节点)，为每个有效计算节点分配一个任务分片，有效计算节点被分配到任务分片后即开始运行该任务分片，不必等待其他计算节点。多个有效计算节点并行运行，如果在最初一轮分配中，每个有效计算节点分配到一个任务分片后，仍存在未分配的任务分片，则等待有效计算节点完成其被分配到的任务分片、释放对应的计算资源后再给该有效计算节点分配未分配的任务分片。

[0083] 根据本说明书实施例的方法，构造了一种并行与串行相结合的分布式计算逻辑，从而有效避免了有效的计算节点出现空置等待的情况。

[0084] 进一步的，在实际应用场景中，很难做到完全避免计算节点出现错误。例如，在Tensorflow的分布式计算模式下，如果计算过程中某个计算节点发生上述的“错误”，则会重启整个作业。整个作业的重启就意味着之前计算节点的计算全部无效，这势必造成计算资源的浪费，降低计算资源的利用效率。针对上述问题，在本说明书一实施例中，在进行任务分片的分配时，如果某计算节点出现错误，将视该计算节点为无效计算节点，不为其分配任务分片。进一步的，在本说明书一实施例中，在有效计算节点运行分配到的任务分片时，如果该有效计算节点出现错误，则其为无效的计算节点，并且，将当前分配到该计算节点的任务分配视为未分配的任务分片，重新对该任务分片进行分配。

[0085] 进一步的，在实际应用场景中，可用计算资源的量并不是固定不变的。在某个分布式任务的执行过程中，可能存在优先级更高的任务被启动需要挪用当前分布式任务的计算资源的情况，或者由硬件/软件错误导致某些可用计算资源变得不可用的情况(当前分布式任务的可用计算资源减少)；也可能存在其他分布式任务完成后释放出新的可用计算资源的情况(当前分布式任务的可用计算资源增加)。对应可用计算资源减少，即当前的有效计

算节点被关闭或挪用;对应可用计算资源增加,即可以拉起新的有效计算节点。

[0086] 针对上述情况,在本说明书一实施例中,当可用计算资源减少,当前的有效计算节点被关闭或挪用,则视该计算节点为无效,将当前分配到该计算节点的任务分配视为未分配的任务分片,重新对该任务分片进行分配;当可用计算资源增加,可以拉起新的有效计算节点,则将未分配的任务分片分配给新拉起的有效计算节点。

[0087] 以下结合附图,详细说明本说明书各实施例提供的技术方案。

[0088] 在本说明书一实施例中,如图1所示,分布式任务运行方法包括以下步骤:

[0089] S110,任务分割步骤,分割待处理任务,获取多个任务分片,其中,每个任务分片为一个独立的任务;

[0090] S120,任务分配步骤,将任务分片分配给有效计算节点进行处理,其中:

[0091] 每个有效计算节点同一时间仅能分配一个任务分片,有效计算节点被分配任务分片后即开始运行任务分片,当有效计算节点完成其被分配的任务分片时,该有效计算节点可被分配新的任务分片;

[0092] 当有效计算节点出错时,将出错的有效计算节点当前所分配到的任务分片重新分配;

[0093] 当有效计算节点被关闭或挪用时,将被关闭或挪用的有效计算节点当前所分配到的任务分片重新分配;

[0094] 当新的有效计算节点被拉起时,将未分配的任务分片分配给新的有效计算节点;

[0095] S130,任务结果获取步骤,当任一有效计算节点完成一个任务分片时,获取对应的任务分片处理结果,综合任务分片对应的任务分片处理结果以获取所述待处理任务对应的任务处理结果。

[0096] 根据本说明书实施例的方法,可以在避免任务分片的遗漏以及过量分配的前提下,有效利用现有的可用计算资源,确保计算节点的运行效率,避免有效计算节点出现等待而导致计算资源被浪费的情况;根据本说明书实施例的方法,还可以在单个计算节点出错时进行任务分片的重新分配,避免由单个计算节点出错而导致整个任务的重新执行,从而减少计算资源的浪费。

[0097] 进一步的,在本说明书一实施例中,在任务结果获取步骤中,综合所有的任务分片对应的任务分片处理结果以获取所述待处理任务对应的任务处理结果。进一步的,在上述应用场景中,先获取的任务分片处理结果需要保存直到所有的任务分片对应的任务分片处理结果才能统一计算,为降低任务分片处理结果统一计算的数据处理压力,每当获取到预设数量的任务分片处理结果后汇总产生任务中间结果。在后续需要之前的任务分片处理结果的计算操作中,直接调用对应的任务中间结果。

[0098] 进一步的,在本说明书一实施例中,为了实现最佳的数据处理效率,为每个计算节点分配最匹配的待处理数据量,在分割待处理任务的过程中,令每个任务分片满足预设数据量。

[0099] 具体的,在本说明书一实施例中,根据历史信息来确定预设数据量。具体的,在本说明书一实施例中,根据历史任务处理纪录确定单个有效计算节点的计算量,根据单个有效计算节点的计算量确定预设的任务分片数据量。

[0100] 进一步的,考虑到待处理任务的多样性,根据历史任务处理纪录所确定单个有效

计算节点的计算量并不能有效匹配当前所要运行的待处理任务，因此，在本说明书一实施例中，由用户自定义设置预设的任务分片数据量。

[0101] 具体的，以一应用场景为例，用户输入的整体数据是10K，然后用户自定义每个任务分片处理1K，那一共就会有10个任务分片(10K/1K)。

[0102] 进一步的，在本说明书一实施例中，由用户指定一个最多计算节点数(max_nodes)，在拉起有效计算节点的过程中，拉起的有效节点总数不超过用户指定数量(max_nodes)。例如用户指定说最多用100个节点来跑。然后主节点根据当前可用计算资源拉起0～100个计算节点来计算。

[0103] 进一步的，在实际应用场景中，存在可用计算资源过量的情况，在这种情况下，如果利用所有的可用计算资源拉起有效计算节点，就会出现某些有效计算节点无法分配到任务分片的情况，从而产生计算资源浪费。针对上述情况，在本说明书一实施例中，根据所述任务分割步骤的处理结果拉起有效计算节点，其中，确认基于可用计算资源能够拉起的有效计算节点数是否满足任务分片数，如果满足，拉起对应任务分片数的有效计算节点，如果不满足，利用所有可用计算资源拉起有效计算节点。

[0104] 具体的，在本说明书一实施例中，如图2所示，方法包括：

[0105] S210，任务分割步骤，分割待处理任务，获取多个任务分片，其中，每个任务分片为一个独立的任务；

[0106] S220，拉起有效计算节点，其中，确认基于可用计算资源能够拉起的有效计算节点数是否满足任务分片数，如果满足，拉起对应任务分片数的有效计算节点，如果不满足，利用所有可用计算资源拉起有效计算节点。

[0107] 进一步的，在本说明书一实施例中，无论基于可用计算资源能够拉起的有效计算节点数是否满足任务分片数，利用可用计算资源拉起的有效计算节点的总数不能超过用户指定数量。

[0108] 进一步的，在本说明书一实施例中，当可用计算资源增加时，如果仍存在未分配的任务分片，则利用增加的可用计算资源拉起新的有效计算节点，将未分配的任务分片分配给新拉起的有效计算节点。

[0109] 具体的，在本说明书一实施例中，方法还包括：

[0110] 在存在未使用的计算资源且存在未分配任务分片时，拉起新的有效计算节点，其中，监控可用计算资源变化，当发现新的可用计算资源且当前存在未分配任务分片时，拉起新的有效计算节点。

[0111] 进一步的，在本说明书一实施例中，在初期可用计算资源过量的情况下，在各个计算节点并行运行时，是存在未使用的可用计算资源的。在这种情况下，如果某个或某几个有效计算节点发生错误，则可以使用未使用的可用计算资源拉起新的有效计算节点，使用新的有效计算节点运行出错的有效计算节点所运行的任务分片。

[0112] 具体的，在本说明书一实施例中，方法还包括：

[0113] 在存在未使用的计算资源且存在未分配任务分片时，拉起新的有效计算节点，其中，当有效计算节点出错导致存在需要重新分配的任务分片且当前存在可用计算资源时，拉起新的有效计算节点。

[0114] 进一步的，在本说明书一实施例中，无论基于何种原因需要拉起新的有效计算节

点，利用可用计算资源拉起的有效计算节点的总数不能超过用户指定数量。

[0115] 进一步的，为了便于后续进行数据分析，在本说明书一实施例中，方法还包括：在有效计算节点运行所述任务分片的过程中，保存有效计算节点的阶段性运行结果。

[0116] 进一步的，为了监控有效计算节点是否出现计算错误，在本说明书一实施例中，方法还包括：在有效计算节点运行任务分片的过程中，检测有效计算节点的阶段性运行结果。

[0117] 进一步的，在本说明书一实施例中，有效计算节点依托可用计算资源进行任务分片的运行。进一步的，还需构造对应的执行单元来执行任务分割步骤以及任务分配步骤。具体的，在本说明书一实施例中，使用客户端来执行任务分割步骤以及任务分配步骤。具体的，使用客户端获取用户的待处理任务，使用客户端执行任务分割步骤获取任务分片，使用客户端利用可用计算资源拉起有效计算节点并向有效计算节点分配任务分片，最终使用客户端获取各个任务分片对应的任务分片处理结果，综合所有的任务分片对应的任务分片处理结果以获取待处理任务对应的任务处理结果。

[0118] 然而，在上述运行模式下，客户端就需要承载较大的运算处理任务(任务分割以及任务分配)。为降低客户端数据处理压力，在本说明书一实施例中，将任务分割以及任务分配交与可用计算资源处理。具体的，在本说明书一实施例中，方法还包括：当存在待处理任务时，利用可用计算资源拉起主节点以及有效计算节点，主节点用于执行任务分割步骤以及任务分配步骤。

[0119] 具体的，在本说明书一实施例中，使用客户端获取用户的待处理任务，使用客户端利用可用计算资源拉起主节点。具体的，在本说明书一实施例中，使用客户端向可用计算资源对应的计算集群装置发送请求，令计算集群装置利用可用计算资源拉起主节点。

[0120] 进一步的，在本说明书一实施例中，在本说明书一实施例中，使用客户端利用可用计算资源拉起有效计算节点，其中，使用客户端向可用计算资源对应的计算集群装置发送请求，令计算集群装置利用可用计算资源拉起有效计算节点。

[0121] 然而，考虑到在并行分布式计算过程中，有效计算节点是存在变动的，而这个变动是与任务分片的分配直接相关的，为了降低数据传输复杂性，在本说明书一实施例中，不由客户端拉起有效计算节点，利用主节点拉起有效计算节点，即，主节点还用于拉起有效计算节点。

[0122] 具体的，在本说明书一实施例中，当存在所述待处理任务时，利用可用计算资源拉起主节点，其中：

[0123] 由客户端采集用户输入的待处理任务；

[0124] 当客户端采集到所述待处理任务时，由客户端发送请求令可用计算资源对应的计算集群装置拉起主节点，并且，由客户端向计算集群装置发送待处理任务。

[0125] 具体的，在本说明书一实施例中，如图3所示：

[0126] S310，由客户端采集用户输入的待处理任务；

[0127] 当客户端采集到待处理任务时，S320，由客户端发送请求令可用计算资源对应的计算集群装置拉起主节点；

[0128] S321，由客户端向主节点发送待处理任务；

[0129] S330，主节点执行任务分割步骤；

[0130] S340，主节点拉起有效计算节点；

[0131] S350,主节点向有效计算节点分配任务分片。

[0132] 进一步的,在本说明书一实施例中,由客户端发送请求令可用计算资源对应的计算集群装置拉起主节点,并且,由客户端向计算集群装置发送待处理任务,包括:

[0133] 由客户端发送待处理任务到可用计算资源对应的计算集群装置;

[0134] 由计算集群装置验证待处理任务,验证通过后由计算集群装置拉起主节点。

[0135] 进一步的,在本说明书一实施例中,主节点还用于执行任务结果获取步骤。

[0136] 具体的,在本说明书一实施例中,如图4所示:

[0137] S410,由客户端100采集用户输入的待处理任务;

[0138] 当客户端100采集到待处理任务时,S420,由客户端发送待处理任务到可用计算资源对应的计算集群装置200;

[0139] S421,计算集群装置200验证待处理任务;

[0140] 验证失败时,S422,计算集群装置200向客户端100返回错误提示;

[0141] 验证成功时,S423,计算集群装置200拉起主节点300;

[0142] S430,主节点300执行任务分割步骤;

[0143] S440,主节点300拉起有效计算节点400;

[0144] S450,主节点300向有效计算节点400分配任务分片;

[0145] S460,有效计算节点400运行任务分片;

[0146] S470,有效计算节点400向主节点300返回任务分片运行结果;

[0147] S480,主节点300汇总任务分片运行结果获取任务运行结果。

[0148] 进一步的,考虑到分布式任务的运行时间一般较常,并且,在任务运行期间,客户端是不需要参与运行的,因此,在本说明书一实施例中,为了降低通信压力,当客户端采集到待处理任务后,建立客户端到计算集群装置间的数据链接;由计算集群装置验证待处理任务,验证通过后断开所述客户端到计算集群装置间的数据链接。

[0149] 进一步的,在本说明书一实施例中,主节点还用于获取并保存有效计算节点的阶段性运行结果。

[0150] 进一步的,在本说明书一实施例中,主节点还用于创建展示页面,该展示页面用于展示当前的任务处理进度状态。

[0151] 进一步的,针对深度学习的应用场景,在本说明书一实施例中:

[0152] 待处理任务包括深度学习作业所需要的训练数据及模型计算信息;

[0153] 任务分片包括对应的训练数据以及运算方法;

[0154] 计算节点用于调用所述任务分片对应的运算方法对所述任务分片对应的训练数据进行数据计算。

[0155] 进一步的,针对深度学习的应用场景,由于深度学习任务是会对同一个数据集做多轮的运算,因此,在本说明书一实施例中,方法还包括,基于用户指定的轮次,对待处理任务进行多轮分布式计算。

[0156] 以一应用场景为例,10K的数据量,每个任务切片1K数据,这样每轮会产生10个任务切片。如果一个深度学习任务一共指定了5轮(这个轮数也是用户自定义的),那最终的任务总数是50。

[0157] 具体的,在本说明书一实施例中,如图5所示:

[0158] 客户端510获取用户输入的待处理任务，之后拉起主节点520，并发送待处理任务，待处理任务被保存在共享存储500中；

[0159] 主节点520开启界面展示服务(tensorboard service)(展示任务运行情况)、运算结果周期性保存服务(checkpoint service)(采集并保存阶段性任务运行结果)、运算结果检测服务(evaluation service)(检测阶段性任务运行结果)；

[0160] 主节点520读取共享存储500中的待处理任务，执行任务分割；

[0161] 主节点520拉起有效计算节点531、532、533...；

[0162] 主节点520向有效计算节点531、532、533...分配任务分片；

[0163] 有效计算节点531、532、533...读取共享存储500中对应被分配的任务分片的数据；

[0164] 有效计算节点531、532、533...运行任务分片；

[0165] 有效计算节点531、532、533...向主节点520反馈阶段性任务运行结果；

[0166] 主节点520检测阶段性任务运行结果并将阶段性任务运行结果保存到共享存储500；

[0167] 有效计算节点531、532、533...向主节点520反馈任务分片运行结果；

[0168] 主节点520汇总任务分片运行结果获取任务运行结果；

[0169] 主节点520将任务运行结果保存到共享存储500。

[0170] 基于本说明书实施例的方法，本说明书实施例还提出了一种分布式任务运行装置。具体的，在本说明书一实施例中，如图6所示，分布式任务运行装置包括：

[0171] 任务分割单元610，其用于分割待处理任务，获取多个任务分片，其中，每个任务分片为一个独立的任务；

[0172] 任务分配单元620，其用于将所述任务分片分配给计算节点进行处理，其中：

[0173] 每个所述有效计算节点同一时间仅能分配一个任务分片，所述有效计算节点被分配任务分片后即开始运行所述任务分片，当所述有效计算节点完成其被分配的任务分片时，所述有效计算节点可被分配新的任务分片；

[0174] 当所述有效计算节点出错时，将出错的有效计算节点当前所分配到的任务分片重新分配；

[0175] 当所述有效计算节点被关闭或挪用时，将被关闭或挪用的有效计算节点当前所分配到的任务分片重新分配；

[0176] 当新的有效计算节点被拉起时，将未分配的任务分片分配给所述新的有效计算节点；

[0177] 任务结果获取单元630，其用于当任一有效计算节点完成一个任务分片时，获取对应的任务分片处理结果，综合任务分片对应的任务分片处理结果以获取所述待处理任务对应的任务处理结果。

[0178] 基于本说明书实施例的方法，本说明书实施例还提出了一种分布式任务运行系统具体的，在本说明书一实施例中，如图7所示，分布式任务运行系统包括：

[0179] 任务获取模块710，其用于获取待处理任务；

[0180] 分布式系统构建模块720，其用于在任务获取模块710获取到待处理任务后，利用可用计算资源建立如本说明书实施例所述的分布式任务运行装置，并向分布式任务运行装

置发送待处理任务。

[0181] 进一步的，基于本发明的方法，本发明还提出了一种用于在访问方设备端信息处理的设备，该设备包括用于存储计算机程序指令的存储器和用于执行程序指令的处理器，其中，当该计算机程序指令被该处理器执行时，触发该设备执行本发明所述的方法。

[0182] 在20世纪90年代，对于一个技术的改进可以很明显地区分是硬件上的改进(例如，对二极管、晶体管、开关等电路结构的改进)还是软件上的改进(对于方法流程的改进)。然而，随着技术的发展，当今的很多方法流程的改进已经可以视为硬件电路结构的直接改进。设计人员几乎都通过将改进的方法流程编程到硬件电路中来得到相应的硬件电路结构。因此，不能说一个方法流程的改进就不能用硬件实体模块来实现。例如，可编程逻辑器件(Programmable Logic Device,PLD)(例如现场可编程门阵列(Field Programmable Gate Array,FPGA))就是这样一种集成电路，其逻辑功能由访问方对器件编程来确定。由设计人员自行编程来把一个数字系统“集成”在一片PLD上，而不需要请芯片制造厂商来设计和制作专用的集成电路芯片。而且，如今，取代手工地制作集成电路芯片，这种编程也多半改用“逻辑编译器(logic compiler)”软件来实现，它与程序开发撰写时所用的软件编译器相类似，而要编译之前的原始代码也得用特定的编程语言来撰写，此称之为硬件描述语言(Hardware Description Language,HDL)，而HDL也并非仅有一种，而是有许多种，如ABEL(Advanced Boolean Expression Language)、AHDL(Altera Hardware Description Language)、Confluence、CUPL(Cornell University Programming Language)、HDCal、JHDL(Java Hardware Description Language)、Lava、Lola、MyHDL、PALASM、RHDL(Ruby Hardware Description Language)等，目前最普遍使用的是VHDL(Very-High-Speed Integrated Circuit Hardware Description Language)与Verilog。本领域技术人员也应该清楚，只需要将方法流程用上述几种硬件描述语言稍作逻辑编程并编程到集成电路中，就可以很容易得到实现该逻辑方法流程的硬件电路。

[0183] 控制器可以按任何适当的方式实现，例如，控制器可以采取例如微处理器或处理器以及存储可由该(微)处理器执行的计算机可读程序代码(例如软件或固件)的计算机可读介质、逻辑门、开关、专用集成电路(Application Specific Integrated Circuit,ASIC)、可编程逻辑控制器和嵌入微控制器的形式，控制器的例子包括但不限于以下微控制器:ARC 625D、Atmel AT91SAM、Microchip PIC18F26K20以及Silicone Labs C8051F320，存储器控制器还可以被实现为存储器的控制逻辑的一部分。本领域技术人员也知道，除了以纯计算机可读程序代码方式实现控制器以外，完全可以通过将方法步骤进行逻辑编程来使得控制器以逻辑门、开关、专用集成电路、可编程逻辑控制器和嵌入微控制器等的形式来实现相同功能。因此这种控制器可以被认为是一种硬件部件，而对其内包括的用于实现各种功能的装置也可以视为硬件部件内的结构。或者甚至，可以将用于实现各种功能的装置视为既可以是实现方法的软件模块又可以是硬件部件内的结构。

[0184] 上述实施例阐明的系统、装置、模块或单元，具体可以由计算机芯片或实体实现，或者由具有某种功能的产品来实现。一种典型的实现设备为计算机。具体的，计算机例如可以为个人计算机、膝上型计算机、蜂窝电话、相机电话、智能电话、个人数字助理、媒体播放器、导航设备、电子邮件设备、游戏控制台、平板计算机、可穿戴设备或者这些设备中的任何设备的组合。

[0185] 为了描述的方便，描述以上装置时以功能分为各种单元分别描述。当然，在实施本申请时可以把各单元的功能在同一个或多个软件和/或硬件中实现。

[0186] 本领域内的技术人员应明白，本发明的实施例可提供为方法、系统、或计算机程序产品。因此，本发明可采用完全硬件实施例、完全软件实施例、或结合软件和硬件方面的实施例的形式。而且，本发明可采用在一个或多个其中包含有计算机可用程序代码的计算机可用存储介质(包括但不限于磁盘存储器、CD-ROM、光学存储器等)上实施的计算机程序产品的形式。

[0187] 本发明是参照根据本发明实施例的方法、设备(系统)、和计算机程序产品的流程图和/或方框图来描述的。应理解可由计算机程序指令实现流程图和/或方框图中的每一流程和/或方框、以及流程图和/或方框图中的流程和/或方框的结合。可提供这些计算机程序指令到通用计算机、专用计算机、嵌入式处理机或其他可编程数据处理设备的处理器以产生一个机器，使得通过计算机或其他可编程数据处理设备的处理器执行的指令产生用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的装置。

[0188] 这些计算机程序指令也可存储在能引导计算机或其他可编程数据处理设备以特定方式工作的计算机可读存储器中，使得存储在该计算机可读存储器中的指令产生包括指令装置的制造品，该指令装置实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能。

[0189] 这些计算机程序指令也可装载到计算机或其他可编程数据处理设备上，使得在计算机或其他可编程设备上执行一系列操作步骤以产生计算机实现的处理，从而在计算机或其他可编程设备上执行的指令提供用于实现在流程图一个流程或多个流程和/或方框图一个方框或多个方框中指定的功能的步骤。

[0190] 在一个典型的配置中，计算设备包括一个或多个处理器(CPU)、输入/输出接口、网络接口和内存。

[0191] 内存可能包括计算机可读介质中的非永久性存储器，随机存取存储器(RAM)和/或非易失性内存等形式，如只读存储器(ROM)或闪存(flash RAM)。内存是计算机可读介质的示例。

[0192] 计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。计算机的存储介质的例子包括，但不限于相变内存(PRAM)、静态随机存取存储器(SRAM)、动态随机存取存储器(DRAM)、其他类型的随机存取存储器(RAM)、只读存储器(ROM)、电可擦除可编程只读存储器(EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器(CD-ROM)、数字多功能光盘(DVD)或其他光学存储、磁盒式磁带，磁带磁磁盘存储或其他磁性存储设备或任何其他非传输介质，可用于存储可以被计算设备访问的信息。按照本文中的界定，计算机可读介质不包括暂存电脑可读媒体(transitory media)，如调制的数据信号和载波。

[0193] 还需要说明的是，术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含，从而使得包括一系列要素的过程、方法、商品或者设备不仅包括那些要素，而且还包括没有明确列出的其他要素，或者是还包括为这种过程、方法、商品或者设备所固有的要素。在没有更多限制的情况下，由语句“包括一个……”限定的要素，并不排除在包括所述要素的过程、方法、商品或者设备中还存在另外的相同要素。

[0194] 本申请可以在由计算机执行的计算机可执行指令的一般上下文中描述，例如程序模块。一般地，程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、对象、组件、数据结构等等。也可以在分布式计算环境中实践本申请，在这些分布式计算环境中，由通过通信网络而被连接的远程处理设备来执行任务。在分布式计算环境中，程序模块可以位于包括存储设备在内的本地和远程计算机存储介质中。

[0195] 本说明书中的各个实施例均采用递进的方式描述，各个实施例之间相同相似的部分互相参见即可，每个实施例重点说明的都是与其他实施例的不同之处。尤其，对于系统实施例而言，由于其基本相似于方法实施例，所以描述的比较简单，相关之处参见方法实施例的部分说明即可。

[0196] 以上所述仅为本申请的实施例而已，并不用于限制本申请。对于本领域技术人员来说，本申请可以有各种更改和变化。凡在本申请的精神和原理之内所作的任何修改、等同替换、改进等，均应包含在本申请的权利要求范围之内。

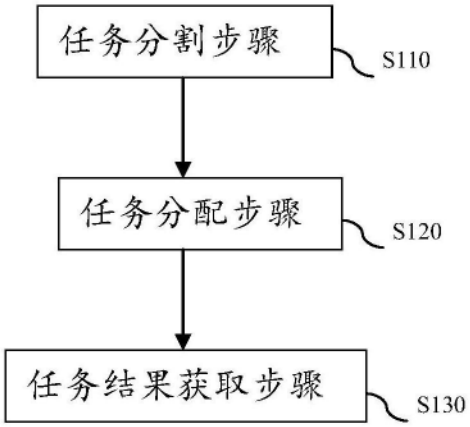

任务分割步骤
S110
任务分配步骤
S120
任务结果获取步骤
S130

图1

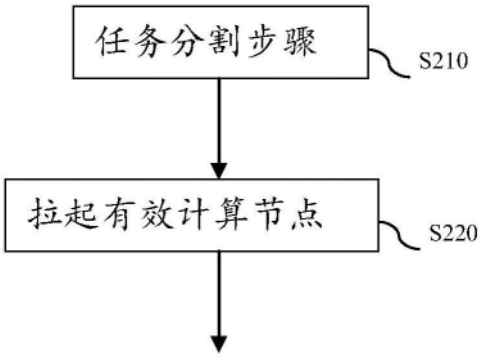

任务分割步骤
S210
拉起有效计算节点
S220

图2

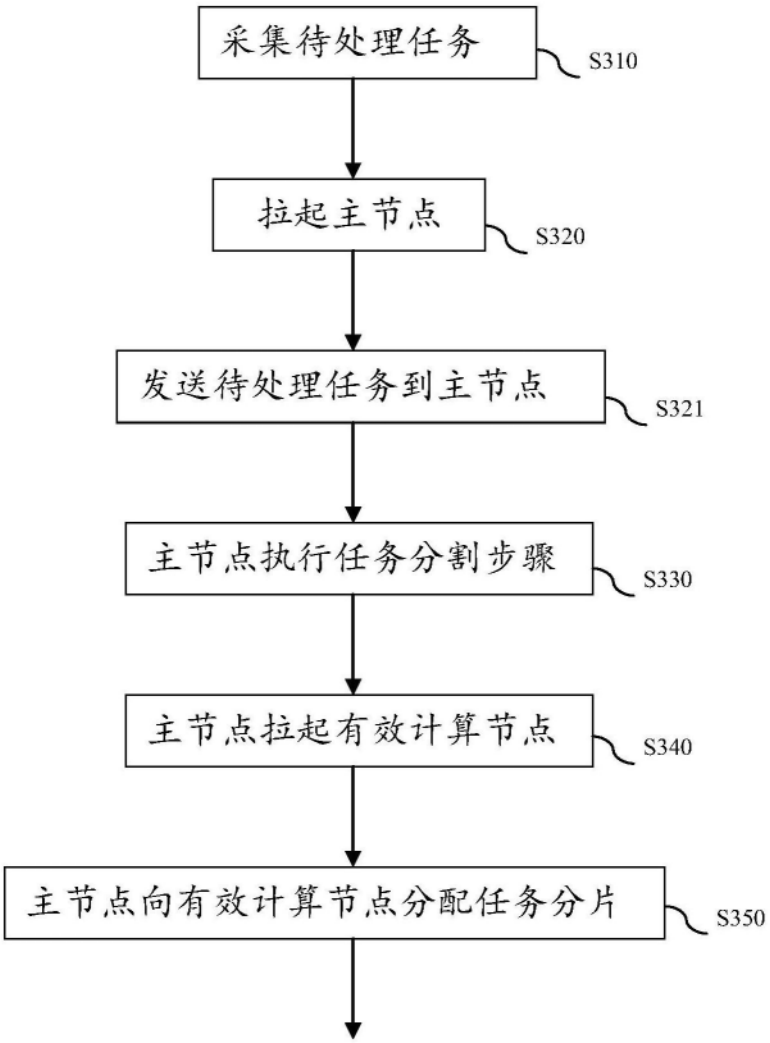

采集待处理任务
S310
拉起主节点
S320
发送待处理任务到主节点
S321
主节点执行任务分割步骤
S330
主节点拉起有效计算节点
S340
主节点向有效计算节点分配任务分片
S350

图3

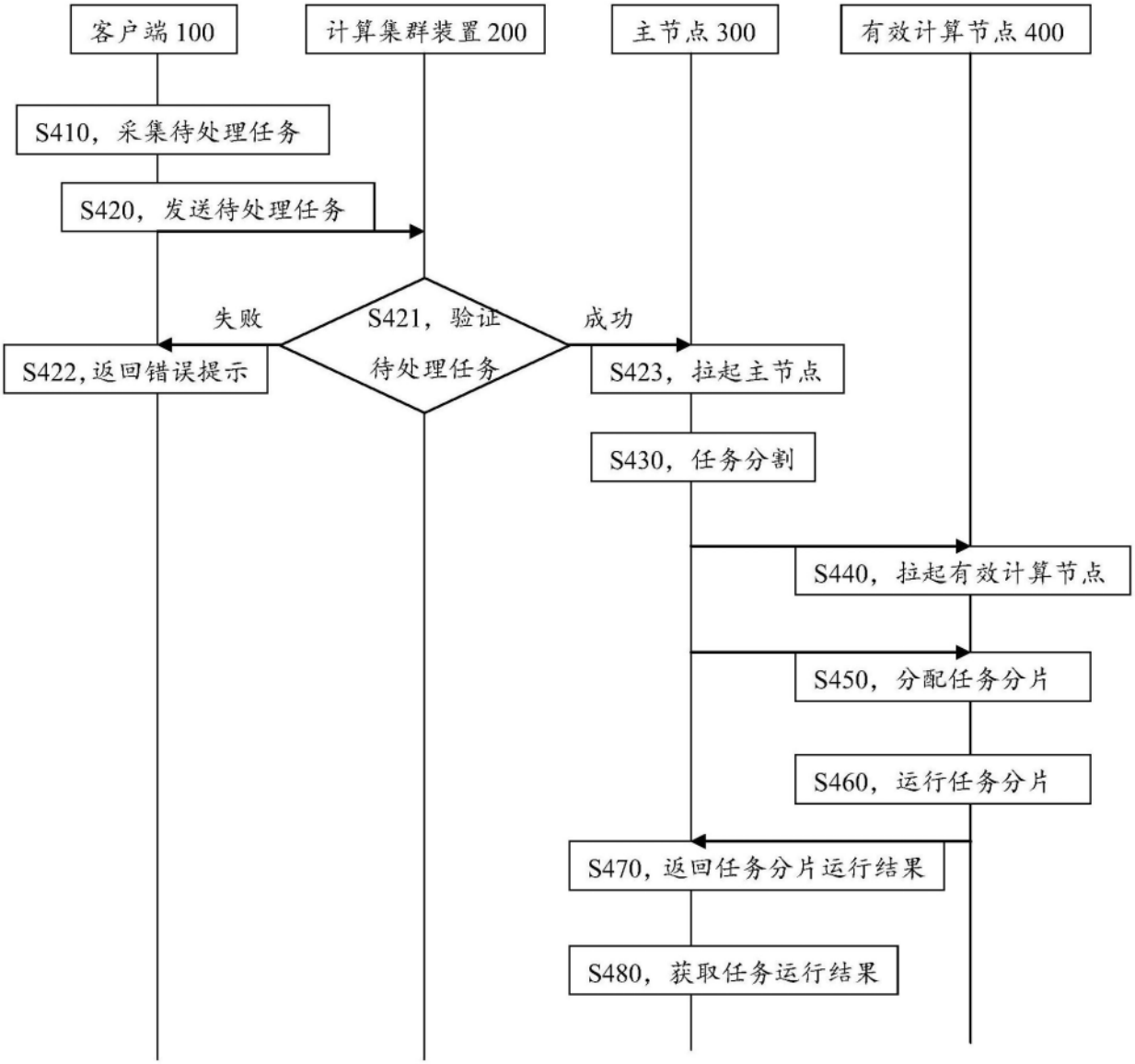

客户端 100
计算集群装置 200
主节点 300
有效计算节点 400
S410，采集待处理任务
S420，发送待处理任务
S421，验证待处理任务
失败
成功
S422,返回错误提示
S423，拉起主节点
S430，任务分割
S440，拉起有效计算节点
S450，分配任务分片
S460，运行任务分片
S470，返回任务分片运行结果
S480，获取任务运行结果

图4

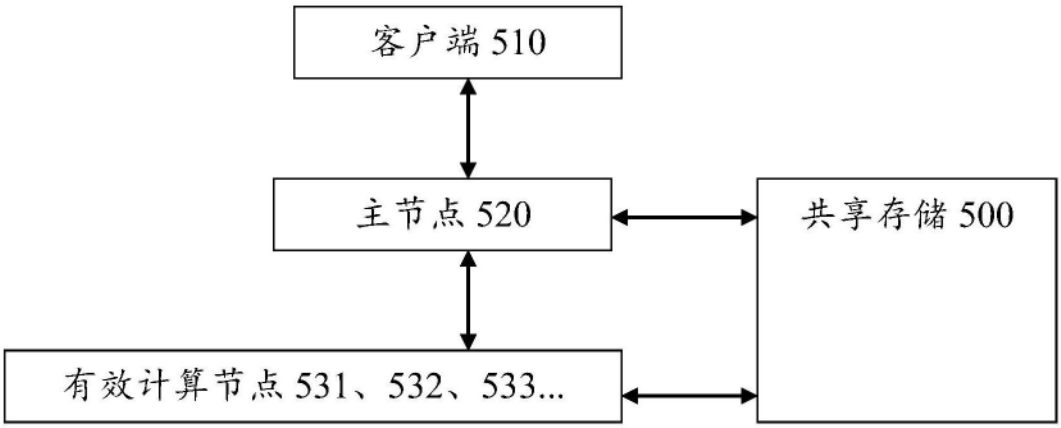

客户端 510
主节点 520
共享存储 500
有效计算节点 531、532、533...

图5

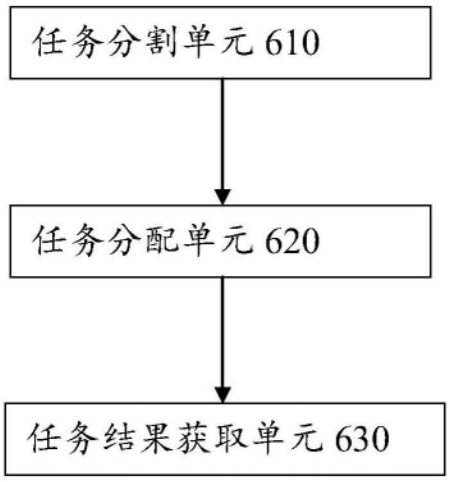

任务分割单元 610
任务分配单元 620
任务结果获取单元 630

图6

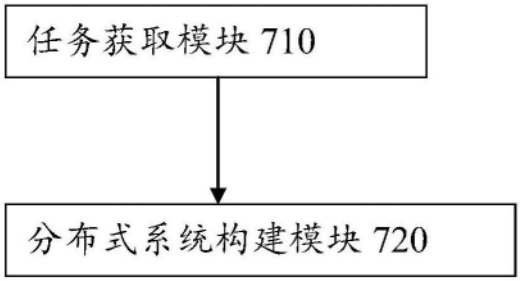

任务获取模块 710
分布式系统构建模块 720

图7