

图数据处理方法、图数据的计算任务发布方法、装置、存储介质及计算机设备

本申请要求于 2018 年 05 月 16 日提交中国专利局、申请号为 201810467817.8、发明名称为“图数据处理方法和图数据的计算任务发布方法”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

本申请涉及图计算技术领域，特别是涉及一种图数据处理方法和装置、图数据的计算任务发布方法、装置、存储介质及计算机设备。

背景

随着计算机技术的发展，出现了以“图论”为基础的对现实世界的一种图结构的抽象表达，以及在这种数据结构上的计算模式，也就是图计算。

图计算是以“图论”为基础的对现实世界的一种“图”结构的抽象表达，以及在这种数据结构上的计算模式。图数据结构很好的表达了数据之间的关联性，因此，很多应用中出现的问题都可以抽象成图来表示，以图论的思想或者以图为基础建立模型来解决问题。例如，可以用图数据表示社交网络、商品购买关系、道路交通网络、通信网络等等信息，而基于图数据的计算需求也越来越多，越来越复杂。

技术内容

本申请实例提供了一种图数据处理方法、图数据的计算任务发布方法、装置、计算机可读存储介质和计算机设备。

本申请实例提供了一种图数据处理方法，由分布式计算节点集群中的计算节点所分布的计算机设备执行，包括：

获取从待处理图数据中划分出的子图数据；

执行对于所述子图数据的计算任务，得到相应全局数据和局部数据；

将所述全局数据写入区块链网络中；所述区块链网络中的全局数据，由所述分布式计算节点集群更新；

从所述区块链网络中获取最新的全局数据；

根据获取的最新的全局数据和所述局部数据，迭代执行对于所述子图数据的计算任务，直到满足迭代停止条件时获得计算结果。

本申请实例还提供了一种图数据处理装置，包括：

计算管理模块，用于获取从待处理图数据中划分出的子图数据；

所述计算管理模块还用于执行对于所述子图数据的计算任务，得到相应
的全局数据和局部数据；

通信模块，用于将所述全局数据写入区块链网络中；所述区块链网络中的
全局数据，由所述分布式计算节点集群更新；

所述通信模块还用于从所述区块链网络中获取最新的全局数据；

所述计算管理模块还用于根据获取的最新的全局数据和所述局部数据，迭
代执行对于所述子图数据的计算任务，直到满足迭代停止条件时获得计算结果。

本申请实例还提供了一种计算机可读存储介质，存储有计算机程序，所述
计算机程序被处理器执行时，使得所述处理器执行所述图数据处理方法的步骤。

本申请实例还提供了一种计算机设备，包括存储器和处理器，所述存储器
存储有计算机程序，所述计算机程序被所述处理器执行时，使得所述处理器执
行所述图数据处理方法的步骤。

本申请实例还提供了一种图数据的计算任务发布方法，由计算机设备执行，
所述方法包括：

获取与待处理图数据相对应的任务信息；

创建新区块；

将所述任务信息写入所述新区块；

当所述新区块通过区块链网络验证后，将所述新区块加入所述区块链网络；

在所述区块链网络中广播所述任务信息；广播的任务信息，用于指示计算
节点获取从待处理图数据中划分出的子图数据，并基于所述区块链网络迭代执
行对于所述子图数据的计算任务，获得计算结果。

本申请实例还提供了一种图数据的计算任务发布装置，所述装置包括：

任务管理模块，用于获取与待处理图数据相对应的任务信息；

通信模块，用于创建新区块；

所述通信模块还用于将所述任务信息写入所述新区块；

所述通信模块还用于当所述新区块通过区块链网络验证后，将所述新区块
加入所述区块链网络；

所述通信模块还用于在所述区块链网络中广播所述任务信息；广播的任务信息，用于指示计算节点获取从待处理图数据中划分出的子图数据，并基于所述区块链网络迭代执行对于所述子图数据的计算任务，获得计算结果。

本申请实例还提供了一种计算机可读存储介质，存储有计算机程序，所述计算机程序被处理器执行时，使得所述处理器执行所述图数据的计算任务发布方法的步骤。

本申请实例还提供了一种计算机设备，包括存储器和处理器，所述存储器存储有计算机程序，所述计算机程序被所述处理器执行时，使得所述处理器执行所述图数据的计算任务发布方法的步骤。

附图简要说明

- 图 1 为一些实例中图数据处理方法的应用环境图；
- 图 2 为一些实例中图数据处理方法的流程示意图；
- 图 3 为一些实例中获取区块链网络中广播的与待处理图数据相对应的任务信息的步骤的流程示意图；
- 图 4 为一些实例中重建丢失数据步骤的流程示意图；
- 图 5 为一些实例中图节点交换步骤的流程示意图；
- 图 6 为另一些实例中图数据处理方法的流程示意图；
- 图 7 为一些实例中图数据的计算任务发布方法的流程示意图；
- 图 8 为一些实例中任务编辑界面的界面示意图；
- 图 9 为一些实例中任务执行状态显示界面的结构示意图；
- 图 10 为一些实例中计算节点状态显示界面的结构示意图；
- 图 11 为另一些实例中图数据的计算任务发布方法的流程示意图；
- 图 12 为一些实例中图数据处理装置的结构框图；
- 图 13 为另一些实例中图数据处理装置的结构框图；
- 图 14 为一些实例中图数据处理装置的模块示意图；
- 图 15 为一些实例中计算机设备的结构框图；
- 图 16 为一些实例中图数据的计算任务发布装置的结构框图；
- 图 17 为一些实例中计算机设备的结构框图。

实施方式

为了使本申请的目的、技术方案及优点更加清楚明白，以下结合附图及实例，对本申请进行进一步详细说明。应当理解，此处所描述的具体实例仅仅用以解释本申请，并不用于限定本申请。

在一些实例中，图计算方法可以是基于中心化思想的图计算方法，在图计算过程中常常使用参数服务器对需要进行共享的数据进行存储和更新。虽然参数服务器可以将需要共享的数据分布式地存储在各个服务器节点上，以在一定程度上避开单点故障，但在数据更新时却需要依赖一个中心化的驱动程序来协调处理。这在图计算过程中往往会导致巨大的网络吞吐量，从而使得网络通信成为图计算系统的瓶颈。特别是当处理超大规模数据量时，数据更新延迟，使得传统的基于中心化的图计算方法存在计算效率低的问题。

基于此，有必要针对图数据计算效率低的技术问题，提供一种图数据处理方法、图数据的计算任务发布方法、装置、计算机可读存储介质和计算机设备。

图 1 为一些实例中图数据处理方法的应用环境图。参照图 1，该图数据处理方法应用于分布式的图计算系统。该分布式的图计算系统包括控制节点 110、计算节点 120 和数据仓库 130。控制节点 110 和计算节点 120 通过区块链网络连接，控制节点 110 和计算节点 120 分别通过网络与数据仓库 130 连接。控制节点 110 可以通过终端来实现，具体可以是台式终端或移动终端，移动终端比如手机、平板电脑或笔记本电脑等。计算节点 120 可以通过部署在一个或多个服务器上的程序来实现，比如图 1 所示的计算节点 a、b、c 等分别分布在高性能计算机 1 至高性能计算机 N 上。数据仓库 130 可以是集中式的存储设备，也可以是分布式的存储集群或设备。在一些实例中，该分布式的图计算系统包括分布式计算节点集群，该图数据处理方法可应用于分布式计算节点集群中的计算节点。

如图 2 所示，在一些实例中，提供了一种图数据处理方法。本实例主要以该方法应用于上述图 1 中的分布式计算节点集群中的计算节点来举例说明。参照图 2，该图数据处理方法具体包括如下步骤：

S202，获取从待处理图数据中划分出的子图数据。

其中，图数据是采用图结构组织起来的结构化数据，通过应用图形理论来存储实体之间的关系信息。在数学上，图结构可以使用二元组 $G = \langle V, E \rangle$ 来表示，其中， $V = \{v_i\}_{i=1}^N$ 表示图数据中的 N 个图节点的集合， $E = \{e_{ij}\} =$

$\{(v_i, v_j)_m\}_{m=1}^M$ 表示图节点之间的连接边。待处理图数据是计算节点待处理的图数据。子图数据是从待处理图数据划分出的部分图数据。

通常，图数据由图节点和图节点间的边构成。图节点是图数据中的顶点，可以用来表示图数据中的主体。比如，当用图数据存储社会网络中个体之间的信息时，可以用图数据中的不同图节点表示不同的个体；或者当用图数据表示商品购买关系时，每个用户和每个商品均是一个图节点。其中，图节点可以包含图节点的节点标识和节点属性等信息。图节点间的边是图数据中的边，可以用来表示图数据中不同主体之间的关系。比如，当用图数据存储社会网络中个体之间的信息时，用图数据中的不同图节点表示不同的个体，可以用图数据中的边表示个体之间的关系，如好友关系；当用图数据表示商品购买关系时，每个用户和每个商品均是一个图节点，用户购买商品表示成一条边等。

在一些实例中，各个计算节点可以是部署在一个或多个服务器上的程序，计算节点可通过区块链这一媒介构成一个虚拟的网络结构。计算节点可通过将数据、计算任务执行日志或计算节点状态等信息写入新区块，并将新区块加入区块链网络，以方便进行计算节点彼此间的通信协调，以完成子图数据所对应的计算任务，进而完成待处理图数据所对应的计算任务。计算节点状态是计算节点的状态信息，包括当前计算节点执行计算任务所消耗的 CPU（Central Processing Unit，中央处理器）内存资源、算法名称和类型、出错次数等。

在一些实例中，该分布式的图计算系统中包含控制节点，控制节点可通过与该分布式的图计算系统中的一个或多个计算节点进行通信，以接入此区块链网络。控制节点可通过将与待处理图数据相对应的任务信息记录在新区块中，并将新区块加入区块链网络，在区块链网络中广播该任务信息。这样，在该区块链网络中的所有计算节点均可接收到广播的任务信息。其中，任务信息是与计算任务相关的信息。

在一些实例中，计算节点可通过定时器或定时程序等定时查看区块链网络中的最新区块中记录的信息。当检测到任务信息后，判断计算节点自身的负载状态，当计算节点未超载时，计算节点可从区块链网络中加载该任务信息。

在一些实例中，任务信息可以包括任务可执行文件的获取路径、执行计算任务的算法参数、子图数据的获取路径、计算结果的输出路径，以及期望完成时间等。计算节点从区块链网络中加载该任务信息后，可获取待处理图数据所对应的图结构中的子图结构，并根据获取的子图结构从数据仓库中获取相应的子图数据。

在一些实例中，当计算节点从区块链网络中加载该任务信息后，可根据自身负载状态，从待处理图数据所对应的图结构中随机获取一定数量的图节点及相应的边，构成子图结构。并根据任务信息中记载的子图数据的获取路径，从数据仓库中获取该子图结构所对应的子图数据。

在一些实例中，分布式的图计算系统可将待处理图数据按照当前的计算节点的数量，将待处理图数据按预设条件，比如平均或随机，划分成相应数量的子图数据。

S204，执行对于子图数据的计算任务，得到相应全局数据和局部数据。

其中，计算任务是对待处理图数据进行处理时需要计算的工作量。全局数据是各计算节点对子图数据进行计算的过程中需要进行全局共享并全局更新的数据，全局数据的规模是有限的。局部数据是各计算节点对子图数据进行计算的过程中仅被少数计算节点使用和更新的数据。

具体地，计算节点在获取从待处理图数据中划分出的子图数据后，可对获取的子图数据中的图节点和边进行相应的计算，得到相应全局数据和局部数据。

在一些实例中，子图数据中的每个图节点可以对应一个单位计算任务。在此实例中，单位计算任务是该分布式的图计算系统中的最小计算分配单位，比如在当次迭代计算过程中，某个图节点对应的计算任务。

在一些实例中，计算节点在加载任务信息后，根据任务信息中记载的任务可执行文件的获取路径，以及执行计算任务的算法参数等信息，从数据仓库中拉取相应的任务可执行文件和执行计算任务的算法参数。进一步地，计算节点在获取子图数据后，根据执行计算任务的算法参数，执行拉取的任务可执行文件，以执行对于子图数据的计算任务。

在一些实例中，计算节点可对全局数据和局部数据随机或依历史经验赋予初始值。计算节点根据执行计算任务的算法参数，以及全局数据和局部数据的初始值，执行任务可执行文件，并更新全局数据和局部数据。

在一些实例中，对待处理图数据进行的计算是迭代计算，随着算法的迭代更新，每次迭代过程中都会产生新的计算任务。子图数据中的每个图节点可以通过算法驱动源源不断地产生需要完成计算的单位计算任务。

在一些实例中，计算节点执行对于子图数据的计算任务，得到相应全局数据和局部数据后，可将相应的数据按不同的任务、不同的图节点进行隔离存储。

S206，将全局数据写入区块链网络中；区块链网络中的全局数据，由分布

式计算节点集群更新。

其中，区块链网络是运行区块链技术的载体和组织方式。区块链技术，简称 BT (Blockchain technology)，也被称之为分布式账本技术，是一种互联网数据库技术，其特点是去中心化、公开透明，让每个人均可参与数据库记录。区块链技术是利用块链式数据结构来验证与存储数据、利用分布式节点共识算法来生成和更新数据、利用密码学的方式保证数据传输和访问的安全、利用由自动化脚本代码组成的智能合约来编程和操作数据的一种全新的分布式基础架构与计算方式。分布式计算节点集群是由分布式的计算节点所组成的一个集合。

具体地，计算节点执行对于子图数据的计算任务，得到相应全局数据和局部数据后，可通过区块链网络在分布式的计算节点中共享获得的全局数据。计算节点可通过将全局数据写入新区块，当新区块通过区块链网络的验证后，将新区块附加在区块链网络的尾端的方式以将全局数据写入区块链网络中。

在一些实例中，该分布式的图计算系统中的各计算节点可异步执行对于子图数据的计算任务，分别得到相应全局数据和局部数据。各计算节点异步执行计算任务，完成相应的计算任务后可根据各计算节点自身的情况将全局数据写入区块链网络中，区块链网络中的全局数据，由分布式计算节点集群共同更新。

在一些实例中，计算节点可对全局数据进行哈希加密，得到相应的字符串，再采用预设的私钥加密字符串生成签名，再将全局数据和签名均写入区块链网络中。当其他节点获取到全局数据和签名后，可从数据仓库中获取相应的公钥验证签名的合法性，当验证通过后再获取相应全局数据并缓存更新本地的全局数据。

在一些实例中，上述区块链网络可以由一个私有的区块链构成，在区块链网络中记录数据所需的密钥可以根据相关非对称加密算法标准，比如椭圆曲线加密算法或 RSA 加密算法（一种非对称加密算法）事先生成，存储在数据仓库中。计算节点可在需要使用时访问数据仓库以获取相关密钥。

在一些实例中，计算节点还可将从上一区块到当前时间的所有相关数据，比如当前的全局数据、计算任务执行日志或计算节点状态等记录在新区块并将新区块加入区块链网络中。

在一些实例中，计算节点可通过主动广播新区块中记录的相关数据，以与分布式的图计算系统中的所有计算节点共享广播的相关数据，比如记录在新区块中的全局数据。或者，也可通过请求的形式扩散新区块中记录的相关数据。

其中，以请求的形式扩散是指，计算节点可通过发送请求的方式获取新区块中记录的相关数据。

在具体应用场景中，待处理图数据所涉及的图计算问题中，待处理图数据所对应的图结构通常是较为稀疏的图，待处理图数据中的边数 M 远远小于图节点数 N 的两两组合： $M \ll A_n^2$ 。这样，通常需要全局共享的全局数据规模有限，大部分的数据都具有一定的局部性，仅被少数计算节点的计算所使用。采用该分布式的图计算系统，将全局数据通过区块链网络进行全局共享，局部数据则可以缓存至相应的计算节点中，通过请求的形式获取，避免了大量的不必要的网络通信开销。

S208，从区块链网络中获取最新的全局数据。

具体地，计算节点可通过区块链网络获取新区块中记录的最新的全局计算数据。在一些实例中，计算节点可通过定时查看新区块的方式，从区块链网络中缓存最新的全局计算数据。

在一些实例中，当区块链网络中广播新区块时，计算节点可获取该新区块中的全局数据。计算节点将获取的全局数据缓存至本地，计算节点可通过计时器驱动，将最老的全局数据进行替代更新，其更新策略与常见的 FIFO (First Input First Output, 先入先出队列) 和 LFU (least frequently used, 最不经常使用页置换算法) 策略类似，在此就不赘述。

在一些实例中，计算节点可定时检测区块链网络中的区块数量，当区块数量增加时，获取新区块中记录的全局数据，根据获取的全局数据更新本地的全局数据。

在一些实例中，计算节点可通过定时器或运行预设定时检测程序以定时检测区块链网络中的区块数据。当区块数量增加时，计算节点可通过请求的形式获取新区块中记录的全局数据，根据获取的全局数据更新本地的全局数据。这样可以快速便捷地更新全局数据，以提高后续对子图数据的处理执行效率。

在一些实例中，计算节点可根据全局数据和局部数据生成新一轮迭代计算的计算任务。当计算节点在生成新一轮迭代计算的计算任务时，以数据请求的形式从区块链网络中获取最新的全局数据，从相邻的计算节点处获取所需的局部数据，根据获取的最新的全球数据、相邻计算节点处的局部数据，以及本地的局部数据，生成新一轮迭代计算的计算任务。

S210，根据获取的最新的全球数据和局部数据，迭代执行对于子图数据的计算任务，直到满足迭代停止条件时获得计算结果。

其中，迭代停止条件是结束迭代计算的条件。迭代停止条件可以是达到预设的迭代次数，或者是迭代计算达到预设时长，亦或者是迭代计算得到的计算结果收敛到稳定值等。

具体地，计算节点可根据获取的最新的全局数据和局部数据，迭代执行对于子图数据的计算任务，若执行完对于子图数据的计算任务后，不满足迭代停止条件，则返回步骤 S204 继续计算，直到满足迭代停止条件时结束计算。

在一些实例中，子图数据中的图节点可根据算法驱动源源不断地产生需要完成计算的计算任务。计算节点可根据子图数据、已完成的计算任务的数量和内容、最新的全局数据和局部数据，确定子图数据中待处理的计算任务。比如，计算节点可根据子图数据，生成当前迭代计算的计算任务所依赖的前 s 个时点的已完成的计算任务，以及相应的最新的全局数据和当前的局部数据，在算法的驱动下生成下次迭代计算的计算任务。计算节点执行生成的计算任务，并更新全局数据和局部数据。以此循环迭代，直到满足迭代停止条件时获得计算结果。

以该图数据处理方法应用于模型训练来举例说明，可将图数据输入至机器学习模型，以确定图数据中的图节点所对应的中间预测结果，按照中间预测结果与图数据相应的标签的差异，调整机器学习模型的模型参数并继续训练，直至满足迭代停止条件时结束训练。其中，每次模型训练过程中，训练得到的模型参数就可包含全局数据和局部数据，当次模型训练过程中会使用前次模型训练得到的模型参数，当次模型训练得到的模型参数再用于下次模型训练，如此循环。

在一些实例中，在图计算问题中，对于数据的一致性要求依算法不同差异很大，在本实例中，由于区块链网络中的信息传递是洪泛式的，因此计算节点获取的全局数据通常会存在一定的数据延迟，而不能保证强一致性，也就是全局参数的误差会在一个可控范围内。这对于如随机梯度下降算法这类是不敏感的。但相应的，由于局部数据是存储于本地，或者通过点对点直接相连进行通信共享的，因此其一致性能够得到很好的保证，不存在数据延迟。

在一些实例中，当计算节点根据获取的最新的全局数据和局部数据，迭代执行对于子图数据的计算任务后，满足迭代停止条件时，可获得子图数据所对应的计算结果。各计算节点可将各子图的局部数据，以及计算结果存储至数据仓库中。数据仓库可对区块链网络中的全局数据、各子图的局部数据、以及各子图所对应的计算结果进行一致性整合。

上述图数据处理方法，通过将待处理图数据分割成子图数据进行分布式处理，可大大提高图数据的处理效率。再将各计算节点在进行图数据的分布式计算过程中获得的需要进行全局共享的全局数据写入区块链网络中，通过区块链网络进行全局共享。大大减少了数据共享所耗费的通信量。并且，在对待处理图数据进行迭代计算的过程中，可迅速直接地从区块链网络中获取最新的全局数据和缓存至本地的局部数据，无需一个中心化的驱动程序来协调更新数据，大大提高了对图数据进行处理的效率。

在一些实例中，步骤 S202 具体包括：获取区块链网络中广播的与待处理图数据相对应的任务信息；根据任务信息，读取相应的任务可执行文件和从待处理图数据中划分出的子图数据。步骤 S204 具体包括：执行任务可执行文件，以执行对于子图数据的计算任务，得到相应的全局数据和局部数据。

具体地，计算节点可通过区块链网络获取区块链网络中广播的与待处理图数据相对应的任务信息，并根据任务信息，从本地或数据仓库中读取相应的任务可执行文件，以及从待处理图数据中划分出的子图数据。执行任务可执行文件，以执行对于子图数据的计算任务。

在一些实例中，该分布式的图计算系统中包含控制节点，控制节点可通过将与待处理图数据相对应的任务信息记录在新区块中，并将新区块加入区块链网络，在区块链网络中广播该任务信息。

在一些实例中，区块链网络中的所有计算节点均可通过新区块的广播而接收到与待处理图数据相对应的任务信息。

在一些实例中，计算节点可通过定时器或定时程序等定时查看区块链网络中的最新区块中记录的信息。主动获取区块链网络中广播的与待处理图数据相对应的任务信息。

上述实例中，获取区块链网络中广播的与待处理图数据相对应的任务信息，再根据任务信息，读取相应的任务可执行文件和相应的子图数据，执行任务可执行文件，可以实现执行计算任务的步骤。其中，任务信息可以是一个很轻的文件，这样，通过区块链网络进行任务的发布可大大减少分布式的图计算系统中任务发布和获取的通信量，大大提高了任务发布的效率。

在一些实例中，获取区块链网络中广播的与待处理图数据相对应的任务信息的步骤具体包括以下步骤：

S302，定时检测区块链网络中广播的与待处理图数据相对应的任务信息。

具体地，计算节点可通过定时器或定时程序等定时检测区块链网络中广播

的与待处理图数据相对应的人物信息。在一些实例中，控制节点通过将任务信息记录在新区块中的方式以将任务信息通过区块链网络进行广播。

S304，当检测到任务信息时，确定当前计算节点的负载状态。

具体地，当计算节点检测到有新的任务信息后，可确定当前计算节点的负载状态。

S306，当负载状态满足预设负载条件时，拉取任务信息。

其中，预设负载条件是预先设置的条件，比如负载量小于预设阈值，或负载量在预设范围内等。计算节点可判断当前的负载状态是否满足预设负载条件，当计算节点的负载状态满足预设负载条件时，比如，计算节点处于空闲状态时，计算节点可拉取任务信息。当计算节点的负载状态不满足预设负载条件时，比如计算节点处于超载状态时，计算节点可忽略该任务信息。

上述实例中，通过定时检测区块链网络中广播的与待处理图数据相对应的任务信息，可实时获取最新的任务信息。当计算节点的负载状态满足预设负载条件时才拉取任务信息，可保障拉取任务信息的计算节点是有空闲资源来处理相应的计算任务的，避免了无效拉取，进一步提高了对图数据进行处理效率。

在一些实例中，步骤 S206 具体包括以下步骤：创建新区块；将全局数据写入创建的新区块；当新区块通过区块链网络验证后，将新区块加入区块链网络。

具体地，计算节点可定时创建新区块，将执行对于子图数据的计算任务后得到的全局计算数据写入创建的新区块中。当新区块通过区块链网络的验证后，将新区块加入区块链网络。

在一些实例中，计算节点可根据相应的共识算法生成新区块。共识算法，比如共识哈希算法。计算节点可将已完成的计算任务相应全局数据写入新区块。在一些实例中，计算节点可将全局数据、计算任务执行日志、图节点的交换数据或计算节点状态等信息写入新区块。

在一些实例中，当区块链是私有链时，可以缩减共识哈希算法的位数，以提高处理效率，达到提升系统吞吐量的目的。缩减共识哈希算法的位数，比如使用 128 位的 SHA（Secure Hash Algorithm，安全哈希算法）而不是 256 位；也可以调节共识哈希算法为 POS（Proof of Work，工作量证明），使用币龄来决定下一个新区块的生成者。

在一些实例中，对于共识哈希函数，也可以采用限制共识前缀 0 的数量来控制复杂度，与位数控制类似，约束哈希结果前缀 0 的数量越多，共识问题的

难度就越大。

在一些实例中，区块链网络可采用相应的共识算法来验证新区块，当新区块通过区块链网络的验证后，计算节点可将该新区块加入区块链网络。

在一些实例中，该分布式的图计算系统中可能有多个新区块同时生成，造成区块链网络的分叉，处理分叉的策略采用常见的多数认同原则，使得最终只有一个分叉成为主链。如果当前计算节点创建的新块没有成为主链，计算节点的状态会退回块产生新区块之前的状态。为避免记录在该新区块中的数据的丢失，各计算节点可将相应的数据缓存至本地一段时间。

上述实例中，将全局数据写入新区块，当新区块通过区块链网络的验证后，可在区块链网络中共享全局数据，大大减小了共享全局数据所需的通信量。

在一些实例中，将全局数据写入创建的新区块的步骤包括，将全局数据和相应的计算任务执行日志写入新区块。该图数据处理方法还包括重建丢失数据的步骤，该步骤具体包括：

S402，根据区块链网络中的计算任务执行日志，确定故障计算节点和相应的丢失数据。

其中，计算任务执行日志是执行计算任务时产生的日志数据。在一些实例中，当计算节点从相邻节点获取局部数据失败时，可从区块链网络中根据需要拉取和整理相应的计算任务执行日志，确定故障计算节点和相应的丢失数据。

S404，确定参与重建丢失数据的计算节点。

具体地，参与重建丢失数据的计算节点通常是故障节点的相邻计算节点。在一些实例中，当确定了故障计算节点时，该故障计算节点的相邻计算节点可依自身负载状态获取该故障计算节点的部分子图数据结构，根据获取的部分子图数据结构从数据仓库中获取相应的部分子图数据，以合并构成新的子图数据。

S406，从确定的计算节点中获取与故障计算节点相关的局部数据。

具体地，计算节点可从参与重建丢失数据的其他计算节点处，获取与故障计算节点相关的局部数据。在一些实例中，计算节点也可从本地缓存的局部数据中查找与故障计算节点相关的局部数据，将查找到的局部数据共享至参与重建丢失数据的计算节点，共享的局部数据用于其他计算节点重建丢失数据。其中，共享的方式可以是计算节点主动将本地的与故障计算节点相关的局部数据发送至相应的计算节点处，也可以是需要数据的计算节点发起数据获取请求，以获取相应的局部数据。

在一些实例中，参与重建丢失数据的计算节点也可以是一个空载的计算节

点。此时，参与重建丢失数据的计算节点可获取与故障计算节点相邻的计算节点共享的与故障计算节点相关的局部数据，根据共享的局部数据重建丢失数据。

S408，根据获取的局部数据重建丢失数据。

具体地，计算节点可根据最小误差原则对于丢失数据进行重建。计算节点可根据获取的局部数据，采用插值法重建丢失数据。其中，常用的差值法，比如滑动平均窗口法、回归方法、插值函数法等，或者将获取的局部数据的均值、中位数或众数作为丢失数据的重建值。

上述实例中，通过记录在区块链网络中的计算任务执行日志，可快速定位故障节点和相应的丢失数据。根据参与重建丢失数据的计算节点所共享的局部数据可快速重建丢失的数据，避免了计算节点的单点故障而影响整体的计算，使得整体分布式的图处理系统具有较高的可靠性。

在一些实例中，该图数据处理方法还包括图节点交换的步骤，该步骤具体包括：

S502，获取已形成相应区块链数据且已完成的计算任务所对应的第一量化值。

其中，区块链数据是记录在区块链网络中的数据，区块链网络中的新区块可以通过广播的方式共享记录的信息。量化值是对计算任务进行量化而得到的数值，比如一个单位计算任务对应一个单位的量化值。第一量化值是对计算节点中已经形成相应区块链数据的，且已完成的计算任务进行量化而得到的数值。第一量化值可以衡量计算节点的计算能力，与部分已完成的计算任务相对应。第一量化值，具体可以是在区块链网络中进行流通和交换的资源，可以称之为通货值或虚拟货币值等。

具体地，计算节点执行对于子图数据的计算任务后，可将部分已完成的计算任务所对应全局数据记录至区块链网络中形成相应区块链数据，这部分已完成的计算任务所对应的量化值即为第一量化值。

在一些实例中，计算节点还可将从上一区块到当前时间的所有相关数据，比如当前的全局数据、计算任务执行日志或计算节点状态记录在新区块并将新区块加入区块链网络，并获得相应的第一量化值。

S504，获取未形成相应区块链数据且已完成的计算任务所对应的第二量化值。

其中，第二量化值是对计算节点中未形成相应区块链数据的，且已完成的计算任务进行量化而得到的数值。第二量化值可以衡量计算节点的计算能力，

与计算节点中另一部分已完成的计算任务相对应。第二量化值，具体可以是计算节点当前所拥有的可兑现为第一量化值的资源，可以称之为资产值。当计算节点将已完成的计算任务所对应的数据记录在区块链网络中形成相应的区块链数据后，这部分区块链数据所对应的资产值就可转换成区块链网络中流通的相同数额的通货值。

具体地，计算节点执行对于子图数据的计算任务后，得到相应全局数据和局部数据。计算节点对已完成的计算任务进行量化，得到第二量化值。在一些实例中，计算节点执行对于子图数据的计算任务后，完成相应的计算任务。当计算节点完成相应的计算任务后，也就可以获取已完成的计算任务所对应的第二量化值。

在一些实例中，计算节点将全局数据写入新区块前，获取已完成的计算任务所对应的历史第二量化值。其中，历史第二量化值，对应将全局数据写入新区块前的已完成的计算任务。当计算节点将全局数据写入新区块，且该新区块通过区块链网络的验证后，从历史第二量化值中扣减生成与写入的全局数据相应的计算任务所对应的第一量化值，得到当前的第二量化值。

这样，通过将全局数据写入新区块以形成相应区块链数据的方式，可将相应的历史第二量化值转换成第一量化值。通过记账的方式，将未来可兑现的第二量化值，快捷方便地转化为相应的可流通的第一量化值。这样，第一量化值和当前的第二量化值的总和就可表示计算节点当前的计算能力。

S506，确定子图数据中未完成的计算任务所对应的第三量化值。

其中，第三量化值是对计算节点中的未完成的计算任务进行量化而得到的数值。第三量化值，具体可以是计算节点待计算的计算任务所对应的数值，可以称之为负债值，可以衡量计算节点的负载状态。当计算节点执行未完成的计算任务并完成后，可将相应的第三量化值转换成相同数额的第二量化值，进而再转换成相同数额的第一量化值。

具体地，计算节点可实时获取当前的未完成的计算任务，根据未完成的计算任务，确定相应的第三量化值。在一些实例中，第一量化值、第二量化值和第三量化值的单位相同，均与计算任务相对应。

在一些实例中，待处理图数据所对应的总任务是不断变化的，各计算节点中的子图数据所产生的子任务也是不断变化的。计算节点可根据子图数据迭代产生计算任务。随着算法的迭代更新，每次迭代过程中都会产生新的计算任务。子图数据中的每个图节点可以通过算法驱动源源不断地产生需要完成计算的单

位计算任务。

在一些实例中，计算节点可根据子图数据、已完成的计算任务的数量、已完成的计算任务的内容、区块链网络中共享的全局数据，以及局部数据，确定子图数据中未完成的计算任务所对应的第三量化值。

S508，当第二量化值和第三量化值不符合均衡条件时，在计算节点之间进行图节点和第一量化值的交换。

其中，均衡条件是预先设置的用于衡量计算节点当前的计算能力和负载状态之间是均衡关系的条件。均衡条件，比如子图数据当前的第三量化值的第二总和，和子图数据当前的第二量化值的第一总和的对比值，在指定范围内。其中，对比值是两个数值的差异值，具体可以通过数学计算来确定两个数值的差异值。数学计算，比如将两个数直接相除、取对数后相除、相减，或进行其他运算后再取对数后相除等等。对比值可以衡量一个数值相对于另一个数值的差异状态。

具体地，当第二量化值和第三量化值不符合均衡条件时，可在计算节点之间进行图节点和第一量化值的交换，以保持第二量化值和第三量化值符合均衡条件。

在一些实例中，均衡条件所对应的指定范围可以是预设的固定范围，也可以是由随时间变化的函数所确定的范围。

在一些实例中，计算节点可获取子图数据当前的第二量化值的第一总和，获取子图数据当前的第三量化值的第二总和。确定第二总和相对于第一总和的对比值，当对比值超出指定范围时，第二量化值和第三量化值不符合均衡条件。

在一些实例中，计算节点可根据以下公式计算第二总和相对于第一总和的对比值：

$$\frac{\log_a^{m+|\sum_{i=1}^n D_i^t|}}{\log_a^{m+|\sum_{i=1}^n C_i^t|}}$$

其中， a 和 m 均为常数； $a > 0$ ，且 $a \neq 1$ ； $m \geq 1$ ； $|\sum_{i=1}^n D_i^t|$ 表示子图数据当前的第三量化值的第二总和； $|\sum_{i=1}^n C_i^t|$ 表示子图数据当前的第二量化值的第一总和。

在一些实例中，当对比值为对第二总和和第一总和取对数后再相除所得的值，则指定范围的最小值 $\mu(t)$ 和最大值 $\lambda(t)$ 均是随时间 t 变化的线性减函数，且 $\mu(t) < \lambda(t)$ 。比如，可用如下公式表示对比值在指定范围内：

$$\mu(t) < \frac{\log_a^{m+|\sum_{i=1}^n D_i^t|}}{\log_a^{m+|\sum_{i=1}^n C_i^t|}} < \lambda(t)$$

在一个具体地实例中，常数 a 可以为10，常数 m 为1，则上述公式可简化为

$$\mu(t) < \frac{\lg(1 + |\sum_{i=1}^n D_i^t|)}{\lg(1 + |\sum_{i=1}^n C_i^t|)} < \lambda(t)$$

在一些实例中，当对比值小于指定范围的最小值时，将当前计算节点的第一量化值与非当前计算节点的相应图节点进行交换，以保持对比值在指定范围内；当对比值大于指定范围的最大值时，将当前计算节点的图节点与非当前计算节点的相应第一量化值进行交换，以保持对比值在指定范围内。

需要指出的是，在上述的实例中，对比值是第二总和相对于第一总和的对比值，当对比值小于指定范围的最小值时，将当前计算节点的第一量化值与非当前计算节点的相应图节点进行交换；当对比值大于指定范围的最大值时，将当前计算节点的图节点与非当前计算节点的相应第一量化值进行交换，以保持对比值在指定范围内。相应的，对比值也可以是第一总和相对于第二总和的对比值，而当对比值大于指定范围的最大值时，将当前计算节点的第一量化值与非当前计算节点的相应图节点进行交换；当对比值小于指定范围的最小值时，将当前计算节点的图节点与非当前计算节点的相应第一量化值进行交换，以保持对比值在指定范围内。

在一些实例中，在计算节点之间进行图节点和第一量化值的交换的步骤具体包括以下步骤：在待进行交换的计算节点间，确定提供待交换的图节点的第一方，以及换取待交换的图节点的第二方。确定第一方对待交换的图节点的第一预估量化值；确定第二方对待交换的图节点的第二预估量化值。根据第一预估量化值和第二预估量化值，确定用于与待交换的图节点交换的第一量化值。在待进行交换的计算节点之间，进行待交换的图节点与确定的第一量化值间的交换。

其中，预估量化值是预估的对待交换的图节点所拥有的未完成的计算任务进行量化的值。第一预估量化值是提供待交换的图节点的第一方对待交换的图节点所拥有的未完成的计算任务进行量化的值。第二预估量化值是换取待交换的图节点的第二方对待交换的图节点所拥有的未完成的计算任务进行量化的值。

具体地，当计算节点为提供待交换的图节点的第一方时，计算节点可根据待交换的图节点所对应的第二量化值与第一方所对应的第二量化值的关系，待

交换的图节点所对应的第三量化值与第一方所对应的第三量化值的关系，确定第一方对待交换的图节点的第一预估量化值。比如，计算节点可通过以下公式计算第一预估量化值：

$$y_1(v_i, k) = \alpha e^{-\beta \frac{D_i - \overline{D_k}}{C_i - \overline{C_k}}}$$

其中， v_i 表示待交换的图节点； k 表示提供待交换的图节点的第一方； $y_1(v_i, k)$ 表示第一方对待交换的图节点的第一预估量化值； D_i 表示待交换的图节点的第三量化值； C_i 表示待交换的图节点的第二量化值； $\overline{D_k}$ 表示提供待交换的图节点的第一方 k 中的图节点的平均第三量化值； $\overline{C_k}$ 表示提供待交换的图节点的第一方 k 中的图节点的平均第二量化值； α 和 β 分别为相应的参数； e 为自然常数。

在一些实例中，当计算节点为换取待交换的图节点的第二方时，计算节点在确定第二预估量化值时，可以考虑在换取待交换的图节点后，第二方所对应的计算节点需要增加的计算任务和减少的通信距离量。因此，计算节点可根据待交换的图节点的第三量化值，以及待交换的图节点与第二方中的图节点间的通信距离，来确定第二方对待交换的图节点的第二预估量化值。比如，计算节点可根据以下公式计算第二预估量化值：

$$y_2(v_i, l) = \alpha e^{-\gamma D_i + \sum_{j \in l} \text{dist}(i, j)}$$

其中， v_i 表示待交换的图节点； l 表示换取待交换的图节点的第二方； $y_2(v_i, l)$ 表示第二方对待交换的图节点的第二预估量化值； D_i 表示待交换的图节点的第三量化值； $\sum_{j \in l} \text{dist}(i, j)$ 表示换取待交换的图节点的第二方 l 中的图节点 j 与待交换的图节点 i 的通信距离的总和。

在一些实例中， $\text{dist}(i, j)$ 可表示图节点 i 和图节点 j 之间的通信距离。由于计算图数据中任意两图节点间的通信距离的计算量巨大，因此计算节点采用数学上的近似方法来计算图节点间的通信距离，比如，可采用局部近似法来计算图节点间的通信距离，具体可根据以下公式计算图节点间的通信距离：

$$\text{dist}(i, j) \stackrel{\text{def}}{=} \begin{cases} 1, & e_{i,j} \in E \\ 0, & e_{i,j} \notin E \end{cases}$$

其中， $\text{dist}(i, j)$ 表示换取待交换的图节点的第二方 l 中的图节点 j 与待交换的图节点 i 间的通信距离； $e_{i,j} \in E$ 表示图节点 j 与图节点 i 之间通过边连接； $e_{i,j} \notin E$ 表示图节点 j 与图节点 i 之间无边。

也就是，当图节点 i 和图节点 j 之间通过边连接时，可近似认为图节点 i 和图

节点 j 的距离很近。当第二方所对应的计算节点换取了待交换的图节点后,计算节点可减少相应的通信距离,因此,第二预估量化值越高。当图节点 i 和图节点 j 之间无边连接时,当第二方所对应的计算节点换取了待交换的图节点后,计算节点并未减少通信距离。

在一些实例中,计算节点可根据第一预估量化值和第二预估量化值,确定用于与待交换的图节点交换的第一量化值。比如,可计算第一预估量化值和第二预估量化值的平均值,将平均值作为与待交换的图节点交换的第一量化值。或者,计算节点也可根据一定的权重比值,对第一预估量化值和第二预估量化值进行加权求和后取平均值,将加权求和后的平均值作为与待交换的图节点交换的第一量化值等。进一步地,在待进行交换的计算节点间,第二方可通过确定的第一量化值换取第一方的待交换的图节点。

上述实例中,分别获取第一量化值、第二量化值和第三量化值,当第二量化值和第三量化值不符合均衡条件时,在计算节点之间进行图节点和第一量化值的交换。其中,第一量化值和第二量化值可以衡量计算节点的计算能力,第三量化值可以衡量计算节点的负载状态,这样,可将计算节点的计算能力和负载状态用和计算任务相应的量化值量化表示,准确又直观。通过在计算节点之间进行图节点和第一量化值的交换,来保持第二量化值和第三量化值满足均衡条件,这样,可以不需要依赖特定的服务器或节点分配任务,而是通过计算节点之间彼此协调分配图节点,并动态调整分配,实现自组织的负载均衡,避免了特定服务器的单点故障和网络拥塞问题,大大提高了任务的调度效率。并且,采用这样自组织的动态任务调度方法,能够适应更大规模集群的计算任务调度,动态增减计算节点的数量均不会对已有的计算任务造成影响,具有高可扩展性。

如图6所示,在一个具体的实例中,图数据处理方法包括以下步骤:

S602,定时检测区块链网络中广播的与待处理图数据相对应的任务信息。

S604,当检测到任务信息时,确定当前计算节点的负载状态。

S606,当负载状态满足预设负载条件时,拉取任务信息。

S608,根据任务信息,读取相应的任务可执行文件和从待处理图数据中划分出的子图数据。

S610,执行任务可执行文件,以执行对于子图数据的计算任务,得到相应的全局数据和局部数据。

S612,创建新区块。

S614,将全局数据和相应的计算任务执行日志写入新区块。

S616, 当新区块通过区块链网络验证后, 将新区块加入区块链网络; 区块链网络中的全局数据, 由分布式计算节点集群更新。

S618, 获取已形成相应区块链数据且已完成的计算任务所对应的第一量化值。

S620, 获取未形成相应区块链数据且已完成的计算任务所对应的第二量化值。

S622, 确定子图数据中未完成的计算任务所对应的第三量化值。

S624, 当第二量化值和第三量化值不符合均衡条件时, 在计算节点之间进行图节点和第一量化值的交换。

S626, 定时检测区块链网络中的区块数量。

S628, 当区块数量增加时, 获取新区块中记录的全局数据。

S630, 根据获取的全局数据更新本地的全局数据。

S632, 根据获取的最新的全局数据和局部数据, 迭代执行对于子图数据的计算任务, 直到满足迭代停止条件时获得计算结果。

S634, 根据区块链网络中的计算任务执行日志, 确定故障计算节点和相应的丢失数据。

S636, 确定参与重建丢失数据的计算节点。

S638, 从确定的计算节点中获取与故障计算节点相关的局部数据。

S640, 根据获取的局部数据重建丢失数据。

上述图数据处理方法, 通过将待处理图数据分割成子图数据进行分布式处理, 可大大提高图数据的处理效率。再将各计算节点在进行图数据的分布式计算过程中获得的需要进行全局共享的全局数据写入区块链网络中, 通过区块链网络进行全局共享。大大减少了数据共享所耗费的通信量。并且, 在对待处理图数据进行迭代计算的过程中, 可迅速直接地从区块链网络中获取最新的全局数据和缓存至本地的局部数据, 无需一个中心化的驱动程序来协调更新数据, 大大提高了对图数据进行处理的效率。

图 6 为一些实例中图数据处理方法的流程示意图。应该理解的是, 虽然图 6 的流程图中的各个步骤按照箭头的指示依次显示, 但是这些步骤并不是必然按照箭头指示的顺序依次执行。除非本文中有明确的说明, 这些步骤的执行并没有严格的顺序限制, 这些步骤可以以其它的顺序执行。而且, 图 6 中的至少一部分步骤可以包括多个子步骤或者多个阶段, 这些子步骤或者阶段并不必然是在同一时刻执行完成, 而是可以在不同的时刻执行, 这些子步骤或者阶段的

执行顺序也不必然是依次进行，而是可以与其它步骤或者其它步骤的子步骤或者阶段的至少一部分轮流或者交替地执行。

在一个具体实例中，控制节点可将任务信息写入新区块，在区块链网络中发布任务信息，计算节点探测到任务信息后判断自身负载状态，当负载状态满足预设负载条件时，拉取任务信息，并根据任务信息从数据仓库中获取子图数据和任务可执行文件。计算节点执行任务可执行文件，以对子图数据进行计算。计算节点在计算过程中，可将计算得到的全局数据、计算任务执行日志和计算节点状态等信息写入区块链网络。计算节点在计算过程中，可分别获取相应的第一量化值、第二量化值和第三量化值，当第二量化值和第三量化值不符合均衡条件时，在计算节点之间进行图节点和第一量化值的交换。这样，不断地对图数据进行循环迭代计算，直到满足迭代停止条件时获得计算结果。计算节点可将相应的计算结果记录到区块链中，或者存储至数据仓库中。

在具体应用场景中，比如 TSP 问题中（Traveling Salesman Problem，旅行推销员问题），可通过上述图数据处理方法高效地求得较优解。其中，TSP 问题具体指推销员如何在最短距离内走完所有城市（图节点）并回到出发的城市。这是一个 NP（Non-deterministic Polynomial，非确定性多项式）完全问题，一般会求其近似解。通过本方案中的图数据处理方法，可将该 TSP 问题拆分成子图中的最短遍历问题，即通过寻找局部最优来近似逼近全局最优解。其中全局数据即为当前待处理图数据中的的路径以及总长度，而局部数据则包括各个子图数据中较优的路径长度。计算的过程为不断优化路径选择，计算节点之间通过将全局数据写入区块链网络中进行共享，可大大减少在计算过程中所耗费的网络通信量。根据不断更新的全局数据和局部数据，以执行计算任务，并求得最优解。

如图 7 所示，在一些实例中，提供了一种图数据的计算任务发布方法。本实例主要以该方法应用于上述图 1 中的控制节点 110 来举例说明。参照图 7，该图数据的计算任务发布方法具体包括如下步骤：

S702，获取与待处理图数据相对应的任务信息。

在一些实例中，控制节点可获取本地存储的与待处理图数据相对应的任务信息，或者通过网络连接、接口连接等方式获取其他设备处所存储的任务信息。

在一些实例中，控制节点可接收任务添加指令，根据任务添加指令展示任务编辑界面。用户可通过任务编辑界面输入任务信息。控制节点可获取输入至任务编辑界面中的与待处理图数据相对应的任务信息。

图 8 示出了一些实例中任务编辑界面的界面示意图。如图 8 所示，控制节点可展示任务编辑界面，具体地，可在任务编辑界面的上方展示任务数量、运行中任务数、成功任务数、失败任务数等信息。在任务编辑界面的中间部位以表格的形式展示各任务的相关信息，比如任务名称、运行时间、运行命令、进度，以及可对已存在的任务进行状态变更的处理控件。在任务编辑界面的下方展示添加任务的文本框，当用户点击“添加任务”的按钮时，可在任务编辑界面的下方对应的文本框中输入相应的任务信息，比如算法参数、任务可执行文件的获取路径、输入数据路径、结果输出路径，以及期望完成的时间等。

S704，创建新区块。

具体地，控制节点在有新的计算任务发布时可创建新区块，以将任务信息写入新区块的方式发布任务信息。

在一些实例中，控制节点在对已经存在或执行的计算任务进行修改参数或输出结果路径等信息时，也可创建新区块，将修改的内容写入新区块，以在区块链网络中更新任务信息。

在一些实例中，控制节点对已经存在或执行的计算任务进行终止任务时，可在区块链中将其标记为不可用状态。也是通过添加新区块的方式在区块链网络中进行广播。

S706，将任务信息写入新区块。

具体地，控制节点可将获取的任务信息写入新区块。

S708，当新区块通过区块链网络验证后，将新区块加入区块链网络。

在一些实例中，区块链网络可采用相应的共识算法来验证新区块，当新区块通过区块链网络的验证后，计算节点可将该新区块加入区块链网络。

S710，在区块链网络中广播任务信息；广播的任务信息，用于指示计算节点获取从待处理图数据中划分出的子图数据，并基于区块链网络迭代执行对于子图数据的计算任务，获得计算结果。

具体地，当控制节点用来记录任务信息的新区块通过区块链网络的验证后，可通过区块链网络广播新区块中的任务信息。广播的任务信息，用于指示计算节点获取从待处理图数据中划分出的子图数据，并基于区块链网络迭代执行对于子图数据的计算任务，获得计算结果。

上述图数据的计算任务发布方法，通过将获取的任务信息写入区块链网络中的新区块，以在区块链网络中广播发布该任务信息，这样在区块链网络中的其他节点就可拉取相应的任务信息，并获取从待处理图数据中划分出的子图数

据，基于区块链网络迭代执行对于子图数据的计算任务，获得计算结果。这样可以在耗费极低的网络通信量的情况下，使得处于区块链网络中的计算节点可以迅速有效地接收到发布的任务信息，大大提高了图数据的计算任务的发布效率，进而提高了对图数据进行计算的处理效率。

在一些实例中，还可通过控制节点展示区块链网络中计算节点的任务执行状态。具体地，控制节点可接收任务执行状态展示指令；根据任务执行状态展示指令在区块链网络中访问相应的计算节点所对应的计算任务执行日志，以获取任务执行状态信息，并展示任务执行状态显示界面；在任务执行状态显示界面中展示区块链网络中已存在的计算任务所对应的任务执行状态信息。

如图9所示，图9示出了一些实例中任务执行状态显示界面的结构示意图。如图9所示，在任务执行状态显示界面中可展示计算节点数量、任务数量、持续运行时间、累计运行时间、成功任务数、失败任务数、运行中任务数、CPU使用率、网络吞吐量、内存使用率以及节点可用率等信息。其中，还可通过图表的形式辅助展示上述相关的信息。对于特定的计算节点，控制节点可访问该计算节点在区块链网络上的计算任务执行日志，并在任务执行状态显示界面中进行展示。

如图10所示，图10示出了一些实例中计算节点状态显示界面的结构示意图。如图10所示，在计算节点状态显示界面中可展示计算节点数量、计算节点可用率、计算节点故障数、CPU使用率、内存使用率、网络吞吐量等信息。还可在计算节点状态显示界面中可展示各节点的状态信息，比如计算节点名称、运行时间、当前状态、负载率、查看日志等。在计算节点状态显示界面的下方展示添加节点的文本框，当用户点击“添加节点”的按钮时，可在计算节点状态显示界面的下方对应的文本框中输入相应的计算节点信息以添加计算节点。其中计算节点信息比如，比如全局数据、节点参数、日志选项和计算任务等。

在一些实例中，可通过控制节点设置系统的运行参数，包括区块链网络的参数，比如区块大小、更新速度、验证方式和加密算法等。

如图11所示，在一个具体的实例中，图数据的计算任务发布方法包括以下步骤：

S1102，接收任务添加指令。

S1104，根据任务添加指令展示任务编辑界面。

S1106，获取输入至任务编辑界面中的与待处理图数据相对应的任务信息。

S1108，创建新区块。

S1110, 将任务信息写入新区块。

S1112, 当新区块通过区块链网络验证后, 将新区块加入区块链网络。

S1114, 在区块链网络中广播任务信息; 广播的任务信息, 用于指示计算节点获取从待处理图数据中划分出的子图数据, 并基于区块链网络迭代执行对于子图数据的计算任务, 获得计算结果。

上述图数据的计算任务发布方法, 通过将获取的任务信息写入区块链网络中的新区块, 以在区块链网络中广播发布该任务信息, 这样在区块链网络中的其他节点就可拉取相应的任务信息, 并获取从待处理图数据中划分出的子图数据, 基于区块链网络迭代执行对于子图数据的计算任务, 获得计算结果。这样可以在耗费极低的网络通信量的情况下, 使得处于区块链网络中的计算节点可以迅速有效地接收到发布的任务信息, 大大提高了图数据的计算任务的发布效率, 进而提高了对图数据进行计算的处理效率。

如图 12 所示, 在一些实例中, 提供了一种图数据处理装置 1200, 包括: 计算管理模块 1201 和通信模块 1202。

计算管理模块 1201, 用于获取从待处理图数据中划分出的子图数据;

计算管理模块 1201 还用于执行对于子图数据的计算任务, 得到相应全局数据和局部数据;

通信模块 1202, 用于将全局数据写入区块链网络中; 区块链网络中的全局数据, 由分布式计算节点集群更新;

通信模块 1202 还用于从区块链网络中获取最新的全局数据;

计算管理模块 1201 还用于根据获取的最新的全局数据和局部数据, 迭代执行对于子图数据的计算任务, 直到满足迭代停止条件时获得计算结果。

在一些实例中, 通信模块 1202 还用于获取区块链网络中广播的与待处理图数据相对应的任务信息; 计算管理模块 1201 还用于根据任务信息, 读取相应的任务可执行文件和从待处理图数据中划分出的子图数据; 执行任务可执行文件, 以执行对于子图数据的计算任务, 得到相应全局数据和局部数据。

在一些实例中, 通信模块 1202 还用于定时检测区块链网络中广播的与待处理图数据相对应的任务信息; 当检测到任务信息时, 确定当前计算节点的负载状态; 当负载状态满足预设负载条件时, 拉取任务信息。

在一些实例中, 通信模块 1202 包括新块生成子模块 12021, 新块生成子模块 12021 用于创建新区块; 将全局数据写入创建的新区块; 当新区块通过区块链网络验证后, 将新区块加入区块链网络。

在一些实例中，新块生成子模块 12021 还用于将全局数据和相应的计算任务执行日志写入新区块；计算管理模块 1201 还包括数据恢复子模块 12011，数据恢复子模块 12011，用于根据区块链网络中的计算任务执行日志，确定故障计算节点和相应的丢失数据；确定参与重建丢失数据的计算节点；从确定的计算节点中获取与故障计算节点相关的局部数据；根据获取的局部数据重建丢失数据。

在一些实例中，数据恢复子模块 12011 还用于查找本地与故障计算节点相关的局部数据；将查找到的局部数据共享至参与重建丢失数据的计算节点；共享的局部数据用于重建丢失数据。

在一些实例中，通信模块 1202 还用于定时检测区块链网络中的区块数量；当区块数量增加时，获取新区块中记录的全局数据；根据获取的全局数据更新本地的全局数据。

如图 13 所示，在一些实例中，图数据处理装置 1200 还包括调度管理模块 1203，调度管理模块 1203 用于获取已形成相应区块链数据且已完成的计算任务所对应的第一量化值；获取未形成相应区块链数据且已完成的计算任务所对应的第二量化值；确定子图数据中未完成的计算任务所对应的第三量化值；当第二量化值和第三量化值不符合均衡条件时，在计算节点之间进行图节点和第一量化值的交换。

上述图数据处理装置，通过将待处理图数据分割成子图数据进行分布式处理，可大大提高图数据的处理效率。再将各计算节点在进行图数据的分布式计算过程中获得的需要进行全局共享的全局数据写入区块链网络中，通过区块链网络进行全局共享。大大减少了数据共享所耗费的通信量。并且，在对待处理图数据进行迭代计算的过程中，可迅速直接地从区块链网络中获取最新的全局数据和缓存至本地的局部数据，无需一个中心化的驱动程序来协调更新数据，大大提高了对图数据进行处理的效率。

在具体应用场景中，如图 14 所示，图 14 示出了一些实例中图数据处理装置可包括通信模块、数据缓存模块、调度管理模块和计算管理模块。其中，通信模块包括新块生成子模块、交易记账子模块、日志管理子模块和状态报告子模块。数据缓存模块包括全局数据缓存子模块、局部数据缓存子模块、更新计时器子模块和数据读取/写回子模块。调度管理模块包括多任务协调子模块、自组织网络子模块、过程控制子模块和密钥管理子模块。计算管理模块包括计算执行子模块、内部变量存储子模块、数据恢复子模块和结果校验子模块。

在一些实例中，作为图 14 所示的实例的一个简化的实现，可以将数据缓存模块中的全局数据缓存子模块和局部数据缓存子模块放入计算管理模块，允许内部变量存储器直接访问数据仓库，省略数据缓存模块，数据缓存的工作完全由计算管理模块完成，并由执行的算法定义驱动数据缓存的策略，从而实现更紧凑的结构。

下面以图 14 为例，详细说明各模块之间是如何相互协调工作的。

其中，通信模块负责所有和区块链网络通信交互的功能，既包括维持区块链网络的正常运行，也包括维持其他模块与区块链网络的有效信息传递。其中，与区块链网络主要进行全局数据、计算任务的算法配置和计算节点状态数据等信息的交互。通信模块中的新块生成子模块用于将当前的全局数据、计算节点状态数据（包括第二量化值和第三量化值，以及图节点索引等数据）和从上一区块产生后发生的图节点交换信息写入新节点。交易记账子模块用于处理图节点的交换情况，即图节点的买卖和第一量化值的管理。日志管理子模块用于整理当前计算节点产生的计算任务执行日志，并从区块链上根据需要拉取和整理所需要的计算任务执行日志，这在故障计算节点的数据恢复过程中十分重要。状态报告子模块从计算管理模块中获得计算节点的状态信息，包括当前执行计算任务所消耗的 CPU 内存资源、算法名称和类型、出错次数等。

在一些实例中，通信模块还可以包括校验模块，以保证新块生成和记账的可靠性。

数据缓存模块包括全局数据缓存、局部数据缓存、更新计时器、数据读写回四个子模块，数据更新策略与常见的 FIFO 和 LFU 策略类似，通过计时器驱动，将最老的缓存替代。

调度管理模块是整个分布式的图计算系统的核心，它驱动着整个系统有序运行。其中，自组织网络子模块负责自组织策略的执行，包括确定用于交换图节点的第一量化值、交易决策、资产核算和管理等。多任务协调子模块控制计算执行多个不同任务的数据，这里可以采用常见的队列调度和优先级调度方法进行控制协调。密钥管理子模块从数据仓库中取得计算节点的授权信息及相应的密钥。过程控制子模块监控计算过程的完整有效性，特别是当计算过程需要回退时的相关上下文处理。

计算管理模块中的计算执行子模块按照调度管理模块的指令，从数据仓库中获取所需的任务可执行文件以完成相关计算。对应的变量存储在内部变量存储子模块中，需要根据不同计算任务和图节点进行数据的隔离存储。数据恢复

子模块用于当区块链网络中出现故障计算节点时，尝试对存储在其上的数据进行有效地恢复。

相比于中心化的图数据处理装置，本申请实例中的图数据处理装置具有以下优势：

高可靠性，由于所有的计算节点可以自发地进行计算，因此单个计算节点的掉线不会影响整体的计算，也不会导致任务的失败，使得整个系统具有较高的可靠性。

灵活性高，正如上一点所述，新加入的节点同样能够很快地自适应地加入这一计算网络，使得计算网络的规模可以灵活扩大

可适应异构计算，由于负载均衡的自主化，异构的设备加入此网络同样可以按照类似的方式协调，不会因为计算能力和方式的差异产生瓶颈。

图 15 示出了一些实例中计算机设备的内部结构图。该计算机设备具体可以是图 1 中的计算节点 120 所分布的计算机设备。如图 15 所示，该计算机设备包括该计算机设备包括通过系统总线连接的处理器、存储器和网络接口。其中，存储器包括非易失性存储介质和内存储器。该计算机设备的非易失性存储介质存储有操作系统，还可存储有计算机程序，该计算机程序被处理器执行时，可使得处理器实现图数据处理方法。该内存储器中也可储存有计算机程序，该计算机程序被处理器执行时，可使得处理器执行图数据处理方法。

本领域技术人员可以理解，图 15 中示出的结构，仅仅是与本申请方案相关的部分结构的框图，并不构成对本申请方案所应用于其上的计算机设备的限定，具体的计算机设备可以包括比图中所示更多或更少的部件，或者组合某些部件，或者具有不同的部件布置。

在一些实例中，本申请提供的图数据处理装置可以实现为一种计算机程序的形式，计算机程序可在如图 15 所示的计算机设备上运行。计算机设备的存储器中可存储组成该图数据处理装置的各个程序模块，比如，图 12 所示的计算管理模块和通信模块。各个程序模块构成的计算机程序使得处理器执行本说明书中描述的本申请各个实例的图数据处理方法中的步骤。

例如，图 15 所示的计算机设备可以通过如图 12 所示的图数据处理装置中的计算管理模块执行步骤 S202、S204 和 S210。计算机设备可通过通信模块执行步骤 S206 和 S208。

如图 16 所示，在一些实例中，提供了一种图数据的计算任务发布装置 1600，包括：任务管理模块 1601 和通信模块 1602。

任务管理模块 1601, 用于获取与待处理图数据相对应的任务信息;

通信模块 1602, 用于创建新区块;

通信模块 1602 还用于将任务信息写入新区块;

通信模块 1602 还用于当新区块通过区块链网络验证后, 将新区块加入区块链网络;

通信模块 1602 还用于在区块链网络中广播任务信息; 广播的任务信息, 用于指示计算节点获取从待处理图数据中划分出的子图数据, 并基于区块链网络迭代执行对于子图数据的计算任务, 获得计算结果。

在一些实例中, 任务管理模块 1601 还用于接收任务添加指令; 根据任务添加指令展示任务编辑界面; 获取输入至任务编辑界面中的与待处理图数据相对应的任务信息。

上述图数据的计算任务发布装置, 通过将获取的任务信息写入区块链网络中的新区块, 以在区块链网络中广播发布该任务信息, 这样在区块链网络中的其他节点就可拉取相应的任务信息, 并获取从待处理图数据中划分出的子图数据, 基于区块链网络迭代执行对于子图数据的计算任务, 获得计算结果。这样可以在耗费极低的网络通信量的情况下, 使得处于区块链网络中的计算节点可以迅速有效地接收到发布的任务信息, 大大提高了图数据的计算任务的发布效率, 进而提高了对图数据进行计算的处理效率。

图 17 示出了一些实例中计算机设备的内部结构图。该计算机设备具体可以是图 1 中的控制节点 110。如图 17 所示, 该计算机设备包括该计算机设备包括通过系统总线连接的处理器、存储器、网络接口、输入装置和显示屏。其中, 存储器包括非易失性存储介质和内存储器。该计算机设备的非易失性存储介质存储有操作系统, 还可存储有计算机程序, 该计算机程序被处理器执行时, 可使得处理器实现图数据的计算任务发布方法。该内存储器中也可储存有计算机程序, 该计算机程序被处理器执行时, 可使得处理器执行图数据的计算任务发布方法。计算机设备的显示屏可以是液晶显示屏或者电子墨水显示屏, 计算机设备的输入装置可以是显示屏上覆盖的触摸层, 也可以是计算机设备外壳上设置的按键、轨迹球或触控板, 还可以是外接的键盘、触控板或鼠标等。

本领域技术人员可以理解, 图 17 中示出的结构, 仅仅是与本申请方案相关的部分结构的框图, 并不构成对本申请方案所应用于其上的计算机设备的限定, 具体的计算机设备可以包括比图中所示更多或更少的部件, 或者组合某些部件, 或者具有不同的部件布置。

在一些实例中，本申请提供的图数据的计算任务发布装置可以实现为一种计算机程序的形式，计算机程序可在如图 17 所示的计算机设备上运行。计算机设备的存储器中可存储组成该图数据的计算任务发布装置的各个程序模块，比如，图 16 所示的任务管理模块和通信模块。各个程序模块构成的计算机程序使得处理器执行本说明书中描述的本申请各个实例的图数据的计算任务发布方法中的步骤。

例如，图 17 所示的计算机设备可以通过如图 16 所示的图数据的计算任务发布装置中的任务管理模块执行步骤 S702。计算机设备可通过通信模块执行步骤 S704、S706 和 S708。

在一些实例中，本方案提供了一种分布式的图计算系统，包括计算节点和控制节点：控制节点用于获取与待处理图数据相对应的任务信息；创建新区块；将任务信息写入新区块；当新区块通过区块链网络验证后，将新区块加入区块链网络；在区块链网络中广播任务信息；广播的任务信息，用于指示计算节点获取从待处理图数据中划分出的子图数据，并基于区块链网络迭代执行对于子图数据的计算任务，获得计算结果。计算节点用于获取从待处理图数据中划分出的子图数据；执行对于子图数据的计算任务，得到相应全局数据和局部数据；将全局数据写入区块链网络中；区块链网络中的全局数据，由分布式计算节点集群更新；从区块链网络中获取最新的全局数据；根据获取的最新的全球数据和局部数据，迭代执行对于子图数据的计算任务，直到满足迭代停止条件时获得计算结果。

上述分布式的图计算系统，控制节点通过将获取的任务信息写入区块链网络中的新区块，以在区块链网络中广播发布该任务信息，可以在耗费极低的网络通信量的情况下，使得处于区块链网络中的计算节点可以迅速有效地接收到发布的任务信息，大大提高了图数据的计算任务的发布效率。计算节点获取任务信息后，可将待处理图数据分割成子图数据进行分布式处理，可大大提高图数据的处理效率。再将各计算节点在进行图数据的分布式计算过程中获得的需要进行全局共享的全局数据写入区块链网络中，通过区块链网络进行全局共享。大大减少了数据共享所耗费的通信量。并且，在对待处理图数据进行迭代计算的过程中，可迅速直接地从区块链网络中获取最新的全局数据和缓存至本地的局部数据，无需一个中心化的驱动程序来协调更新数据，大大提高了对图数据进行处理效率。

在一些实例中，提供了一种计算机设备，包括存储器和处理器，存储器中

储存有计算机程序，计算机程序被处理器执行时，使得处理器执行以下步骤：获取从待处理图数据中划分出的子图数据；执行对于子图数据的计算任务，得到相应全局数据和局部数据；将全局数据写入区块链网络中；区块链网络中的全局数据，由分布式计算节点集群更新；从区块链网络中获取最新的全局数据；根据获取的最新的全球数据和局部数据，迭代执行对于子图数据的计算任务，直到满足迭代停止条件时获得计算结果。

在一些实例中，计算机程序使得处理器在执行获取从待处理图数据中划分出的子图数据的步骤时具体执行以下步骤：获取区块链网络中广播的与待处理图数据相对应的任务信息；根据任务信息，读取相应的任务可执行文件和从待处理图数据中划分出的子图数据；计算机程序使得处理器在执行对于子图数据的计算任务，得到相应全局数据和局部数据的步骤时具体执行以下步骤：执行任务可执行文件，以执行对于子图数据的计算任务，得到相应全局数据和局部数据。

在一些实例中，计算机程序使得处理器在执行获取区块链网络中广播的与待处理图数据相对应的任务信息的步骤时具体执行以下步骤：定时检测区块链网络中广播的与待处理图数据相对应的任务信息；当检测到任务信息时，确定当前计算节点的负载状态；当负载状态满足预设负载条件时，拉取任务信息。

在一些实例中，计算机程序使得处理器在执行将全局数据写入区块链网络中的步骤时具体执行以下步骤：创建新区块；将全局数据写入创建的新区块；当新区块通过区块链网络验证后，将新区块加入区块链网络。

在一些实例中，计算机程序使得处理器在执行将全局数据写入创建的新区块的步骤时具体执行以下步骤：将全局数据和相应的计算任务执行日志写入新区块；计算机程序使得处理器还执行以下步骤：根据区块链网络中的计算任务执行日志，确定故障计算节点和相应的丢失数据；确定参与重建丢失数据的计算节点；从确定的计算节点中获取与故障计算节点相关的局部数据；根据获取的局部数据重建丢失数据。

在一些实例中，计算机程序使得处理器还执行以下步骤：查找本地与故障计算节点相关的局部数据；将查找到的局部数据共享至参与重建丢失数据的计算节点；共享的局部数据用于重建丢失数据。

在一些实例中，计算机程序使得处理器在执行从区块链网络中获取最新的全局数据的步骤时具体执行以下步骤：定时检测区块链网络中的区块数量；当区块数量增加时，获取新区块中记录的全局数据；根据获取的全局数据更新本

地的全局数据。

在一些实例中，计算机程序使得处理器还执行以下步骤：获取已形成相应区块链数据且已完成的计算任务所对应的第一量化值；获取未形成相应区块链数据且已完成的计算任务所对应的第二量化值；确定子图数据中未完成的计算任务所对应的第三量化值；当第二量化值和第三量化值不符合均衡条件时，在计算节点之间进行图节点和第一量化值的交换。

上述计算机设备，通过将待处理图数据分割成子图数据进行分布式处理，可大大提高图数据的处理效率。再将各计算节点在进行图数据的分布式计算过程中获得的需要进行全局共享的全局数据写入区块链网络中，通过区块链网络进行全局共享。大大减少了数据共享所耗费的通信量。并且，在对待处理图数据进行迭代计算的过程中，可迅速直接地从区块链网络中获取最新的全局数据和缓存至本地的局部数据，无需一个中心化的驱动程序来协调更新数据，大大提高了对图数据进行处理效率。

在一些实例中，提供了一种计算机设备，包括存储器和处理器，存储器中储存有计算机程序，计算机程序被处理器执行时，使得处理器执行以下步骤：获取与待处理图数据相对应的任务信息；创建新区块；将任务信息写入新区块；当新区块通过区块链网络验证后，将新区块加入区块链网络；在区块链网络中广播任务信息；广播的任务信息，用于指示计算节点获取从待处理图数据中划分出的子图数据，并基于区块链网络迭代执行对于子图数据的计算任务，获得计算结果。

在一些实例中，计算机程序使得处理器在执行获取与待处理图数据相对应的任务信息的步骤时具体执行以下步骤：接收任务添加指令；根据任务添加指令展示任务编辑界面；获取输入至任务编辑界面中的与待处理图数据相对应的任务信息。

上述计算机设备，通过将获取的任务信息写入区块链网络中的新区块，以在区块链网络中广播发布该任务信息，这样在区块链网络中的其他节点就可拉取相应的任务信息，并获取从待处理图数据中划分出的子图数据，基于区块链网络迭代执行对于子图数据的计算任务，获得计算结果。这样可以在耗费极低的网络通信量的情况下，使得处于区块链网络中的计算节点可以迅速有效地接收到发布的任务信息，大大提高了图数据的计算任务的发布效率，进而提高了对图数据进行计算的处理效率。

一种计算机可读存储介质，存储有计算机程序，该计算机程序被处理器执

行时实现以下步骤：获取从待处理图数据中划分出的子图数据；执行对于子图数据的计算任务,得到相应全局数据和局部数据；将全局数据写入区块链网络中；区块链网络中的全局数据,由分布式计算节点集群更新；从区块链网络中获取最新的全局数据；根据获取的最新的全局数据和局部数据,迭代执行对于子图数据的计算任务,直到满足迭代停止条件时获得计算结果。

在一些实例中,计算机程序使得处理器在执行获取从待处理图数据中划分出的子图数据的步骤时具体执行以下步骤：获取区块链网络中广播的与待处理图数据相对应的任务信息；根据任务信息,读取相应的任务可执行文件和从待处理图数据中划分出的子图数据；计算机程序使得处理器在执行对于子图数据的计算任务,得到相应全局数据和局部数据的步骤时具体执行以下步骤：执行任务可执行文件,以执行对于子图数据的计算任务,得到相应全局数据和局部数据。

在一些实例中,计算机程序使得处理器在执行获取区块链网络中广播的与待处理图数据相对应的任务信息的步骤时具体执行以下步骤：定时检测区块链网络中广播的与待处理图数据相对应的任务信息；当检测到任务信息时,确定当前计算节点的负载状态；当负载状态满足预设负载条件时,拉取任务信息。

在一些实例中,计算机程序使得处理器在执行将全局数据写入区块链网络中的步骤时具体执行以下步骤：创建新区块；将全局数据写入创建的新区块；当新区块通过区块链网络验证后,将新区块加入区块链网络。

在一些实例中,计算机程序使得处理器在执行将全局数据写入创建的新区块的步骤时具体执行以下步骤：将全局数据和相应的计算任务执行日志写入新区块；计算机程序使得处理器还执行以下步骤：根据区块链网络中的计算任务执行日志,确定故障计算节点和相应的丢失数据；确定参与重建丢失数据的计算节点；从确定的计算节点中获取与故障计算节点相关的局部数据；根据获取的局部数据重建丢失数据。

在一些实例中,计算机程序使得处理器还执行以下步骤：查找本地与故障计算节点相关的局部数据；将查找到的局部数据共享至参与重建丢失数据的计算节点；共享的局部数据用于重建丢失数据。

在一些实例中,计算机程序使得处理器在执行从区块链网络中获取最新的全局数据的步骤时具体执行以下步骤：定时检测区块链网络中的区块数量；当区块数量增加时,获取新区块中记录的全局数据；根据获取的全局数据更新本地的全局数据。

在一些实例中，计算机程序使得处理器还执行以下步骤：获取已形成相应区块链数据且已完成的计算任务所对应的第一量化值；获取未形成相应区块链数据且已完成的计算任务所对应的第二量化值；确定子图数据中未完成的计算任务所对应的第三量化值；当第二量化值和第三量化值不符合均衡条件时，在计算节点之间进行图节点和第一量化值的交换。

上述计算机可读存储介质，通过将待处理图数据分割成子图数据进行分布式处理，可大大提高图数据的处理效率。再将各计算节点在进行图数据的分布式计算过程中获得的需要进行全局共享的全局数据写入区块链网络中，通过区块链网络进行全局共享。大大减少了数据共享所耗费的通信量。并且，在对待处理图数据进行迭代计算的过程中，可迅速直接地从区块链网络中获取最新的全局数据和缓存至本地的局部数据，无需一个中心化的驱动程序来协调更新数据，大大提高了对图数据进行处理效率。

一种计算机可读存储介质，存储有计算机程序，该计算机程序被处理器执行时实现以下步骤：获取与待处理图数据相对应的任务信息；创建新区块；将任务信息写入新区块；当新区块通过区块链网络验证后，将新区块加入区块链网络；在区块链网络中广播任务信息；广播的任务信息，用于指示计算节点获取从待处理图数据中划分出的子图数据，并基于区块链网络迭代执行对于子图数据的计算任务，获得计算结果。

在一些实例中，计算机程序使得处理器在执行获取与待处理图数据相对应的任务信息的步骤时具体执行以下步骤：接收任务添加指令；根据任务添加指令展示任务编辑界面；获取输入至任务编辑界面中的与待处理图数据相对应的任务信息。

上述计算机可读存储介质，通过将获取的任务信息写入区块链网络中的新区块，以在区块链网络中广播发布该任务信息，这样在区块链网络中的其他节点就可拉取相应的任务信息，并获取从待处理图数据中划分出的子图数据，基于区块链网络迭代执行对于子图数据的计算任务，获得计算结果。这样可以在耗费极低的网络通信量的情况下，使得处于区块链网络中的计算节点可以迅速有效地接收到发布的任务信息，大大提高了图数据的计算任务的发布效率，进而提高了对图数据进行计算的处理效率。

本领域普通技术人员可以理解实现上述实例方法中的全部或部分流程，是可以通过计算机程序来指令相关的硬件来完成，所述的程序可存储于一非易失性计算机可读存储介质中，该程序在执行时，可包括如上述各方法的实例的

流程。其中，本申请所提供的各实例中所使用的对存储器、存储、数据库或其它介质的任何引用，均可包括非易失性和/或易失性存储器。非易失性存储器可包括只读存储器 (ROM)、可编程 ROM (PROM)、电可编程 ROM (EPROM)、电可擦除可编程 ROM (EEPROM) 或闪存。易失性存储器可包括随机存取存储器 (RAM) 或者外部高速缓冲存储器。作为说明而非局限，RAM 以多种形式可得，诸如静态 RAM (SRAM)、动态 RAM (DRAM)、同步 DRAM (SDRAM)、双数据率 SDRAM (DDRSDRAM)、增强型 SDRAM (ESDRAM)、同步链路 (Synchlink) DRAM (SLDRAM)、存储器总线 (Rambus) 直接 RAM (RDRAM)、直接存储器总线动态 RAM (DRDRAM)、以及存储器总线动态 RAM (RDRAM) 等。

以上实例的各技术特征可以进行任意的组合，为使描述简洁，未对上述实例中的各个技术特征所有可能的组合都进行描述，然而，只要这些技术特征的组合不存在矛盾，都应当认为是本说明书记载的范围。

以上所述实例仅表达了本申请的几种实施方式，其描述较为具体和详细，但并不能因此而理解为对本申请专利范围的限制。应当指出的是，对于本领域的普通技术人员来说，在不脱离本申请构思的前提下，还可以做出若干变形和改进，这些都属于本申请的保护范围。因此，本申请专利的保护范围应以所附权利要求为准。