



(12) 发明专利申请

(10) 申请公布号 CN 117579629 A

(43) 申请公布日 2024. 02. 20

(21) 申请号 202311427995.5

(22) 申请日 2023.10.31

(71) 申请人 杭州富算科技有限公司
地址 310051 浙江省杭州市滨江区西兴街
道缤纷街615号4楼401室

(72) 发明人 尤志强 卞阳 王兆凯 赵华宇
张伟奇

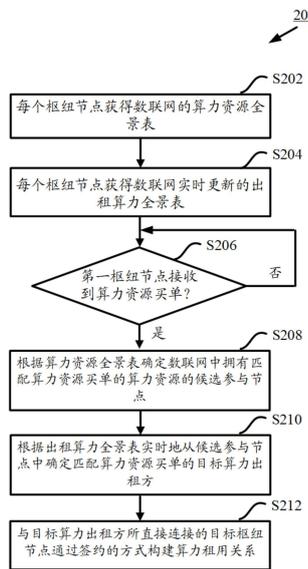
(74) 专利代理机构 北京慧加伦知识产权代理有
限公司 16035
专利代理师 李永敏

(51) Int. Cl.
H04L 67/104 (2022.01)
H04L 67/12 (2022.01)
H04L 67/1008 (2022.01)
H04L 67/60 (2022.01)

权利要求书3页 说明书10页 附图2页

(54) 发明名称
用于数联网的算力撮合方法

(57) 摘要
本公开的实施例提供一种用于数联网的算力撮合方法。数联网包括多个子网。每个子网包括枢纽节点和与枢纽节点直接连接的多个参与节点。该多个子网中的枢纽节点相互直接连接。算力撮合方法包括：每个枢纽节点获得数联网的算力资源全景表；每个枢纽节点获得数联网实时更新的出租算力全景表；响应于第一子网中的第一枢纽节点接收到第一子网中作为算力租用方的参与节点所提交的算力资源买单，第一枢纽节点：根据算力资源全景表确定数联网中拥有匹配算力资源买单的算力资源的候选参与节点；根据出租算力全景表实时地从候选参与节点中确定匹配算力资源买单的目标算力出租方；与目标算力出租方所直接连接的目标枢纽节点通过签约的方式构建算力租用关系。



1. 一种用于数联网的算力撮合方法,其特征在于,所述数联网包括多个子网,每个子网包括枢纽节点和与所述枢纽节点直接连接的多个参与节点,所述多个子网中的枢纽节点相互直接连接,所述算力撮合方法包括:

每个枢纽节点获得所述数联网的算力资源全景表,所述算力资源全景表包括所述数联网中的所有参与节点的算力资源信息;

每个枢纽节点获得所述数联网实时更新的出租算力全景表,所述出租算力全景表包括所述数联网中的所有作为算力出租方的参与节点所发布的算力出租信息;

响应于第一子网中的第一枢纽节点接收到所述第一子网中作为算力租用方的参与节点所提交的算力资源买单,所述第一枢纽节点执行以下操作:

根据所述算力资源全景表确定所述数联网中拥有匹配所述算力资源买单的算力资源的候选参与节点;

根据所述出租算力全景表实时地从所述候选参与节点中确定匹配所述算力资源买单的目标算力出租方;以及

与所述目标算力出租方所直接连接的目标枢纽节点通过签约的方式构建算力租用关系。

2. 根据权利要求1所述的算力撮合方法,其特征在于,所述算力撮合方法还包括:

每个算力出租方以容器的形式隔离待出租的闲赋算力资源;

响应于所述目标算力出租方的数量大于1,将所有目标算力出租方中的容器组成容器集群;

从各个目标算力出租方所直接连接的目标枢纽节点中选举一个主枢纽节点;

在执行所述算力租用方的指令的过程中,所述主枢纽节点实时监测所述容器集群的负载情况;

响应于所述负载减少,所述主枢纽节点缩减所述容器集群的规模;以及

响应于所述负载增加,所述主枢纽节点扩展所述容器集群的规模。

3. 根据权利要求2所述的算力撮合方法,其特征在于,所述算力撮合方法还包括:

响应于所述目标算力出租方的数量等于1,所述目标算力出租方所直接连接的目标枢纽节点收集和管理所述目标算力出租方中的容器的日志数据,以及在所述容器发生故障或不可用的情况下重启所述容器;以及

响应于所述目标算力出租方的数量大于1,所述主枢纽节点收集和管理所述容器集群中的各个容器的日志数据,以及在任一容器发生故障或不可用的情况下重启所述容器。

4. 根据权利要求1所述的算力撮合方法,其特征在于,每个枢纽节点获得所述数联网实时更新的出租算力全景表包括:

由每个子网中的所述枢纽节点收集该子网中的所述多个参与节点所发布的算力出租信息;

所述数联网中的所有枢纽节点相互转发所收集的算力出租信息以使得每个枢纽节点各自形成所述数联网的出租算力全景表;

响应于第一参与节点的算力出租信息已更新,所述第一参与节点立即向其直接连接的指定枢纽节点发布更新后的算力出租信息,所述指定枢纽节点使用所述第一参与节点的更新后的算力出租信息来实时更新所述出租算力全景表,所述指定枢纽节点向所述数联网中

的其它枢纽节点转发所述第一参与节点的更新后的算力出租信息,以便所述其它枢纽节点实时更新所述出租算力全景表。

5. 根据权利要求1至4中任一项所述的算力撮合方法,其特征在于,所述算力撮合方法还包括每个参与节点执行以下操作:

根据该参与节点的历史算力资源开销来预测该参与节点自身在未来一段时间的资源需求;

根据该参与节点的算力资源和所预测的资源需求来确定该参与节点在未来一段时间的闲赋算力资源;

该参与节点以容器的形式隔离所述闲赋算力资源;

响应于所预测的资源需求增加,释放所述容器中未使用的资源;以及

响应于所预测的资源需求降低,增加所述容器中的资源。

6. 根据权利要求1至4中任一项所述的算力撮合方法,其特征在于,所述算力撮合方法还包括每个参与节点执行以下操作:

对该参与节点的算力资源的使用情况进行实时监测以确定该参与节点实时的闲赋算力资源,所述使用情况包括算力资源的分配情况、使用率和闲置时长;

该参与节点以容器的形式隔离所述闲赋算力资源;

响应于该参与节点的算力资源的实时使用量增加,立即释放所述容器中未使用的资源;以及

响应于该参与节点的算力资源的实时使用量降低,立即增加所述容器中的资源。

7. 根据权利要求1至4中任一项所述的算力撮合方法,其特征在于,所述算力撮合方法还包括:所述目标枢纽节点在完成与所述第一枢纽节点的签约时立即冻结所述目标算力出租方被出租的算力资源。

8. 根据权利要求1至4中任一项所述的算力撮合方法,其特征在于,所述算力撮合方法还包括:

第二枢纽节点计算与所述第二枢纽节点直接连接的所有第二参与节点的算力贡献度;

所述第二枢纽节点根据每个第二参与节点的算力贡献度来计算该第二参与节点的贡献度占比;以及

所述第二枢纽节点按照每个第二参与节点的贡献度占比来向该第二参与节点分配奖励以提高该第二参与节点在算力撮合过程中的参与度;

其中,所述第二参与节点中的第j节点的算力贡献度被计算为:

$$\text{score}_j = (1 + W_{rc}) \times \left[\sum_i^n (W_{rai} \times R_{ai}) + W_{rt} \times \sum_i^m (W_{rqi} \times R_{qi}) \right]$$

其中, score_j 表示所述第j节点的算力贡献度, R_{ai} 表示所述第j节点的n个算力资源中的第i算力资源的数量, W_{rai} 表示根据所述第i算力资源的类别所设置的权重, R_{qi} 表示所述第j节点的m个质量参数中的第i质量参数, W_{rqi} 表示针对所述第i质量参数设置的权重, W_{rt} 表示根据所述第j节点出租算力的时间确定的权重, W_{rc} 表示根据所述第j节点所参与的任务的完成率确定的权重;

其中,所述第j节点的贡献度占比被计算为:

$$\text{ratio}_j = \frac{e^{\text{score}_j}}{\sum_j^K e^{\text{score}_j}}$$

其中, ratio_j 表示所述第 j 节点的贡献度占比, K 表示所述第二参与节点的节点数量, score_j 表示所述第 j 节点的算力贡献度。

9. 根据权利要求1至4中任一项所述的算力撮合方法, 其特征在于, 每个枢纽节点获得所述数联网的算力资源全景表包括:

由每个子网中的所述枢纽节点收集该子网中的所述多个参与节点的算力资源信息;

所述数联网中的所有枢纽节点相互转发所收集的算力资源信息以使得每个枢纽节点各自形成所述数联网的算力资源全景表。

10. 根据权利要求1至4中任一项所述的算力撮合方法, 其特征在于, 根据所述出租算力全景表实时地从所述候选参与节点中确定匹配所述算力资源买单的目标算力出租方包括:

响应于所述第一枢纽节点接收到多个算力资源买单, 通过联合求解下式来确定匹配每个算力资源买单的目标算力出租方:

$$\begin{cases} \min \sum (c_{ij} \times x_{ij}) \\ \sum (w_{ij} \times x_{ij}) \geq d_i \\ \sum (w_{ij} \times x_{ij}) \leq R_j \end{cases}$$

其中, x_{ij} 表示第 j 候选参与节点是否被第 i 任务选中, $x_{ij} = 1$ 表示选中, $x_{ij} = 0$ 表示未选中, c_{ij} 表示所述第 j 候选参与节点的算力资源分配给所述第 i 任务的成本, d_i 表示所述第 i 任务的总算力资源需求, w_{ij} 表示所述第 j 候选参与节点分配给所述第 i 任务的算力资源量, R_j 表示所述第 j 候选参与节点的总出租算力资源, 所述第 i 任务是第 i 算力资源买单对应的任务。

用于数联网的算力撮合方法

技术领域

[0001] 本公开的实施例涉及数联网技术领域,具体地,涉及用于数联网的算力撮合方法。

背景技术

[0002] 数据共享和数据价值的流通,对于当前推进数字化经济转型发展越来越重要。随着对数据的互通、共享、交换的需求持续升温,继个人计算机(PC)互联网、移动互联网、产业互联网、物联网之后,“数联网”也应运而生。本文所提到的新型“数联网”,是实现数据要素跨行业、跨区域、跨机构流通的基础设施,与传统的狭义数联网(仅仅用于数交所或类似场景中的数据买卖交易的互联网)显著不同。狭义数联网仅仅是“数数相连的互联网”,仅涉及数据在数联网中透明传输。应用端可以直接获取数据,产生各种基于数据的业务应用。

[0003] 在新型“数联网”中,数据使用权的转让应该更加合理有序。在此类数联网中,非常关键的底层能力是算力。算力在分布上是松散的。数联网作为开放性的网络体系,可以支持海量主体通过被允许的方式接入数联网。接入数联网的主体构成数联网中的一个数据节点。该节点具备一定的算力资源。当某些节点本地的计算任务结束之后,这些节点的算力资源会存在一定的闲置,导致算力资源的浪费。如果将闲置的算力资源出租给其它节点,则可充分利用数联网的算力资源,使得参与到数联网的节点获得除数据流通之外的价值。

发明内容

[0004] 本文中描述的实施例提供了一种用于数联网的算力撮合方法。

[0005] 根据本公开的第一方面,提供了一种用于数联网的算力撮合方法。数联网包括多个子网。每个子网包括枢纽节点和与枢纽节点直接连接的多个参与节点。该多个子网中的枢纽节点相互直接连接。该算力撮合方法包括:每个枢纽节点获得数联网的算力资源全景表,算力资源全景表包括数联网中的所有参与节点的算力资源信息;每个枢纽节点获得数联网实时更新的出租算力全景表,出租算力全景表包括数联网中的所有作为算力出租方的参与节点所发布的算力出租信息;响应于第一子网中的第一枢纽节点接收到第一子网中作为算力租用方的参与节点所提交的算力资源买单,第一枢纽节点执行以下操作:根据算力资源全景表确定数联网中拥有匹配算力资源买单的算力资源的候选参与节点;根据出租算力全景表实时地从候选参与节点中确定匹配算力资源买单的目标算力出租方;以及与目标算力出租方所直接连接的目标枢纽节点通过签约的方式构建算力租用关系。

[0006] 在本公开的一些实施例中,算力撮合方法还包括:每个算力出租方以容器的形式隔离待出租的闲赋算力资源;响应于目标算力出租方的数量大于1,将所有目标算力出租方中的容器组成容器集群;从各个目标算力出租方所直接连接的目标枢纽节点中选举一个主枢纽节点;在执行算力租用方的指令的过程中,主枢纽节点实时监测容器集群的负载情况;响应于负载减少,主枢纽节点缩减容器集群的规模;以及响应于负载增加,主枢纽节点扩展容器集群的规模。

[0007] 在本公开的一些实施例中,算力撮合方法还包括:响应于目标算力出租方的数量

等于1,目标算力出租方所直接连接的目标枢纽节点收集和管理目标算力出租方中的容器的日志数据,以及在容器发生故障或不可用的情况下重启容器;以及响应于目标算力出租方的数量大于1,主枢纽节点收集和管理容器集群中的各个容器的日志数据,以及在任一容器发生故障或不可用的情况下重启容器。

[0008] 在本公开的一些实施例中,每个枢纽节点获得数联网实时更新的出租算力全景表包括:由每个子网中的枢纽节点收集该子网中的多个参与节点所发布的算力出租信息;数联网中的所有枢纽节点相互转发所收集的算力出租信息以使得每个枢纽节点各自形成数联网的出租算力全景表;响应于第一参与节点的算力出租信息已更新,第一参与节点立即向其直接连接的指定枢纽节点发布更新后的算力出租信息,指定枢纽节点使用第一参与节点的更新后的算力出租信息来实时更新出租算力全景表,指定枢纽节点向数联网中的其它枢纽节点转发第一参与节点的更新后的算力出租信息,以便其它枢纽节点实时更新出租算力全景表。

[0009] 在本公开的一些实施例中,算力撮合方法还包括每个参与节点执行以下操作:根据该参与节点的历史算力资源开销来预测该参与节点自身在未来一段时间的资源需求;根据该参与节点的算力资源和所预测的资源需求来确定该参与节点在未来一段时间的闲赋算力资源;该参与节点以容器的形式隔离闲赋算力资源;响应于所预测的资源需求增加,释放容器中未使用的资源;以及响应于所预测的资源需求降低,增加容器中的资源。

[0010] 在本公开的一些实施例中,算力撮合方法还包括每个参与节点执行以下操作:对该参与节点的算力资源的使用情况进行实时监测以确定该参与节点实时的闲赋算力资源,使用情况包括算力资源的分配情况、使用率和闲置时长;该参与节点以容器的形式隔离闲赋算力资源;响应于该参与节点的算力资源的实时使用量增加,立即释放容器中未使用的资源;以及响应于该参与节点的算力资源的实时使用量降低,立即增加容器中的资源。

[0011] 在本公开的一些实施例中,算力撮合方法还包括:目标枢纽节点在完成与第一枢纽节点的签约时立即冻结目标算力出租方被出租的算力资源。

[0012] 在本公开的一些实施例中,算力撮合方法还包括:第二枢纽节点计算与第二枢纽节点直接连接的所有第二参与节点的算力贡献度;第二枢纽节点根据每个第二参与节点的算力贡献度来计算该第二参与节点的贡献度占比;以及第二枢纽节点按照每个第二参与节点的贡献度占比来向该第二参与节点分配奖励以提高该第二参与节点在算力撮合过程中的参与度。

[0013] 其中,第二参与节点中的第j节点的算力贡献度被计算为:

$$[0014] \quad \text{score}_j = (1 + W_{rc}) \times \left[\sum_i^n (W_{rai} \times R_{ai}) + W_{rt} \times \sum_i^m (W_{rqi} \times R_{qi}) \right]$$

[0015] 其中, score_j 表示第j节点的算力贡献度, R_{ai} 表示第j节点的n个算力资源中的第i算力资源的数量, W_{rai} 表示根据第i算力资源的类别所设置的权重, R_{qi} 表示第j节点的m个质量参数中的第i质量参数, W_{rqi} 表示针对第i质量参数设置的权重, W_{rt} 表示根据第j节点出租算力的时间确定的权重, W_{rc} 表示根据第j节点所参与的任务的完成率确定的权重。

[0016] 其中,第j节点的贡献度占比被计算为:

$$[0017] \quad \text{ratio}_j = \frac{e^{\text{score}_j}}{\sum_j^K e^{\text{score}_j}}$$

[0018] 其中, ratio_j 表示第 j 节点的贡献度占比, K 表示第二参与节点的节点数量, score_j 表示第 j 节点的算力贡献度。

[0019] 在本公开的一些实施例中, 每个枢纽节点获得数联网的算力资源全景表包括: 由每个子网中的枢纽节点收集该子网中的多个参与节点的算力资源信息; 数联网中的所有枢纽节点相互转发所收集的算力资源信息以使得每个枢纽节点各自形成数联网的算力资源全景表。

[0020] 在本公开的一些实施例中, 根据出租算力全景表实时地从候选参与节点中确定匹配算力资源买单的目标算力出租方包括: 响应于第一枢纽节点接收到多个算力资源买单, 通过联合求解下式来确定匹配每个算力资源买单的目标算力出租方:

$$[0021] \quad \begin{cases} \min \sum (c_{ij} \times x_{ij}) \\ \sum (w_{ij} \times x_{ij}) \geq d_i \\ \sum (w_{ij} \times x_{ij}) \leq R_j \end{cases}$$

[0022] 其中, x_{ij} 表示第 j 候选参与节点是否被第 i 任务选中, $x_{ij} = 1$ 表示选中, $x_{ij} = 0$ 表示未选中, c_{ij} 表示第 j 候选参与节点的算力资源分配给第 i 任务的成本, d_i 表示第 i 任务的总算力资源需求, w_{ij} 表示第 j 候选参与节点分配给第 i 任务的算力资源量, R_j 表示第 j 候选参与节点的总出租算力资源, 第 i 任务是第 i 算力资源买单对应的任务。

[0023] 在本公开的一些实施例中, 算力资源买单包括: 关于期望的算力资源的价格、数量、性能要求、任务描述。

[0024] 在本公开的一些实施例中, 算力出租信息包括算力资源卖单。算力资源卖单包括: 关于待出租的算力资源的价格、数量、性能、可执行任务的描述。

[0025] 根据本公开的第二方面, 提供了一种存储有计算机程序的计算机可读存储介质, 其中, 计算机程序在由处理器执行时实现根据本公开的第一方面所述的算力撮合方法的步骤。

附图说明

[0026] 为了更清楚地说明本公开的实施例的技术方案, 下面将对实施例的附图进行简要说明, 应当知道, 以下描述的附图仅仅涉及本公开的一些实施例, 而非对本公开的限制, 其中:

[0027] 图1是数联网的示意性拓扑图;

[0028] 图2是根据本公开的实施例的用于数联网的算力撮合方法的示意性流程图。

[0029] 需要注意的是, 附图中的元素是示意性的, 没有按比例绘制。

具体实施方式

[0030] 为了使本公开的实施例的目的、技术方案和优点更加清楚, 下面将结合附图, 对本公开的实施例的技术方案进行清楚、完整的描述。显然, 所描述的实施例是本公开的一部分实施例, 而不是全部的实施例。基于所描述的本公开的实施例, 本领域技术人员在无需创造性劳动的前提下所获得的所有其它实施例, 也都属于本公开保护的范围。

[0031] 除非另外定义, 否则在此使用的所有术语 (包括技术和科学术语) 具有与本公开主

题所属领域的技术人员所通常理解相同含义。进一步将理解的是,诸如在通常使用的词典中定义的那些的术语应解释为具有与说明书上下文和相关技术中它们的含义一致的含义,并且将不以理想化或过于正式的形式来解释,除非在此另外明确定义。另外,诸如“第一”和“第二”的术语仅用于将一个部件(或部件的一部分)与另一个部件(或部件的另一部分)区分开。

[0032] 图1示出数联网的示意性拓扑图。数联网可包括多个子网10。每个子网10包括枢纽节点11和与枢纽节点直接连接的多个参与节点12。该多个子网10中的枢纽节点11相互直接连接。枢纽节点11与枢纽节点11之间可以通过专网进行互联。枢纽节点11承担对参与节点12进行信息聚合、寻址导航等功能。参与节点12可以是各类政务主体、行业主体、公司主体、机构主体等。直接连接到同一个枢纽节点11的参与节点12通过该枢纽节点11进行通信。直接连接到不同枢纽节点11的参与节点12通过它们各自直接连接的枢纽节点11进行通信。也就是说,参与节点12只与其直接连接的枢纽节点11直接通信,枢纽节点11之间可直接通信,而参与节点12之间需经由相应的枢纽节点11进行通信。

[0033] 在实践中,数联网中可能存在海量的子网10。单个子网10中可能存在海量的参与节点12。因此,数联网中的参与节点12的数量可能是非常庞大的。当某些参与节点12本地的计算任务结束之后,这些参与节点12的算力资源会存在一定的闲置,导致算力资源的浪费。如果这些参与节点12将闲置的算力资源出租给需要额外算力的参与节点12,那么需要额外算力的参与节点12就不必购买额外的硬件设备,只需要租用数联网中闲置的算力资源就可完成计算任务,这样数联网的整体算力资源可被很好地整合和利用。

[0034] 算力租用方对额外算力的需求通常是实时的,因此需要快速找到可使用的闲置算力资源来帮助算力租用方完成计算任务。而闲置算力资源的分布会随着参与节点12本地计算任务的变化而实时变化。因此,在数联网中对海量的参与节点12的算力进行撮合时需要考虑时效性的问题。

[0035] 本公开提出了一种用于数联网的算力撮合方法。图2示出根据本公开的实施例的用于数联网的算力撮合方法200的示意性流程图。

[0036] 在图2的框S202处,每个枢纽节点11获得数联网的算力资源全景表。算力资源全景表包括数联网中的所有参与节点12的算力资源信息。算力资源信息可包括:计算资源和存储资源等,如处理器、内存、硬盘存储、图形处理器、分布式计算资源、服务器集群、内网带宽等配置信息以及型号等。数联网中的参与节点12一般是异构的,也就是不同参与节点12所提供的算力能力是存在差异的。有的参与节点12能够提供分布式计算资源,有的参与节点12只能提供单机计算资源。不同的计算资源提供水平会影响到后续算力组合之后的计算效率以及费用结算。

[0037] 在本公开的一些实施例中,由每个子网10中的枢纽节点11收集该子网10中的多个参与节点12的算力资源信息。在一个示例中,各参与节点12在注册加入到数联网时可以向其直接连接的枢纽节点11主动上报算力资源信息。在另一个示例中,各参与节点12生成本地资源配置信息表,然后被动地等待枢纽节点11来采集该资源配置信息表。枢纽节点11收集齐同一子网10中的所有参与节点12的算力资源信息后,可向其它枢纽节点转发所收集的算力资源信息。在一个示例中,数联网中的所有枢纽节点11相互转发所收集的算力资源信息以使得每个枢纽节点11各自形成数联网的算力资源全景表。在另一个示例中,可从数联

网中的所有枢纽节点11中选举出一个主枢纽节点。其它枢纽节点向主枢纽节点转发所收集的算力资源信息。主枢纽节点根据每个枢纽节点11所收集的算力资源信息来形成数联网的算力资源全景表,然后再将算力资源全景表转发给其它枢纽节点。

[0038] 参与节点12的算力资源取决于其硬件配置,因此每个参与节点12的算力资源可以认为是相对静态的(相对于完成一次算力撮合所需的时间来说,可以认为是固定不变的),这样,算力资源全景表可以认为是相对静态的。

[0039] 在参与节点12的算力资源发生改变时,参与节点12可向其直接连接的枢纽节点11主动上报新的算力资源信息,以便枢纽节点11更新算力资源全景表并将这一变化同步给其它枢纽节点。

[0040] 在框S204处,每个枢纽节点11获得数联网实时更新的出租算力全景表。出租算力全景表包括数联网中的所有作为算力出租方的参与节点12所发布的算力出租信息。在本公开的一些实施例中,算力出租信息包括算力资源卖单。算力资源卖单可包括:关于待出租的算力资源的价格、数量、性能、可执行任务的描述。

[0041] 在本公开的一些实施例中,由每个子网10中的枢纽节点11收集该子网10中的多个参与节点12所发布的算力出租信息。枢纽节点11收集算力出租信息的方式可与收集算力资源信息的方式相同。在一个示例中,数联网中的所有枢纽节点11相互转发所收集的算力出租信息以使得每个枢纽节点11各自形成数联网的出租算力全景表。在另一个示例中,可从数联网中的所有枢纽节点11中选举出一个主枢纽节点。其它枢纽节点向主枢纽节点转发所收集的算力出租信息。主枢纽节点根据每个枢纽节点11所收集的算力出租信息来形成数联网的出租算力全景表,然后再将出租算力全景表转发给其它枢纽节点。

[0042] 如果第一参与节点(第一参与节点可以是数联网中的任一参与节点12)的算力出租信息已更新,则第一参与节点立即向其直接连接的指定枢纽节点发布更新后的算力出租信息,指定枢纽节点使用第一参与节点的更新后的算力出租信息来实时更新出租算力全景表,指定枢纽节点向数联网中的其它枢纽节点转发第一参与节点的更新后的算力出租信息,以便其它枢纽节点实时更新出租算力全景表。

[0043] 在本公开的一些实施例中,各个参与节点12可通过动态分析预估的方式来确定闲赋算力资源(在上下文中也可替换地称为“闲置算力资源”)。每个参与节点12可根据该参与节点12的历史算力资源开销来预测该参与节点12自身在未来一段时间的资源需求。参与节点12根据自身的算力资源和所预测的资源需求来确定该参与节点12在未来一段时间的闲赋算力资源。在一个示例中,闲赋算力资源可等于该参与节点12的算力资源减去所预测的资源需求。在另一个示例中,为保证本地的计算任务能够被及时执行,闲赋算力资源可等于该参与节点12的算力资源减去所预测的资源需求与预设冗余值之和。这样,本地的计算任务所需的算力资源小幅度变化时,也有足够的算力资源支持该计算任务。

[0044] 参与节点12可以以容器的形式隔离闲赋算力资源。容器技术作为一种轻量级虚拟化技术,在云计算环境中被广泛应用。容器技术可以实现应用程序的快速部署、资源隔离和弹性伸缩,极大地提高了应用的可移植性和灵活性。参与节点12可执行容器隔离性评估,以评估容器之间的隔离程度,包括文件系统隔离、进程隔离和网络隔离等。在评估容器隔离性的过程中,可以通过测试容器之间的相互影响和干扰程度,例如在一个容器中进行网络扫描,观察其它容器是否受到影响。参与节点12可以通过记录容器的系统日志和行为日志,以

及应用安全审计和监控工具,对容器进行实时监测和分析,监测容器的运行状态、检测容器漏洞和异常行为、及时发现和响应安全事件等。参与节点12可评估容器镜像的安全性,包括镜像来源的可信度、镜像签名的验证、镜像的漏洞和恶意软件检测等。参与节点12可以使用镜像扫描工具对容器镜像进行漏洞扫描和恶意软件检测,以确保容器镜像的安全性。另外参与节点12还可以对容器网络的安全性进行评估,包括容器之间的通信安全和容器与外部网络的连接安全。参与节点12可以通过网络流量监测和防火墙规则验证等方法,检测和阻止潜在的网络攻击和入侵。因此通过采用容器技术,参与节点12能够安全且高效的隔离闲赋算力资源,以便其它参与节点12使用该闲赋算力资源。

[0045] 如果参与节点12所预测的资源需求增加,则参与节点12可释放容器中未使用的资源。如果参与节点12所预测的资源需求降低,则参与节点12可增加容器中的资源。这样,参与节点12可弹性的配置闲赋算力资源,保证本地的计算任务能够被及时执行且本地的闲赋算力资源不会被闲置。

[0046] 在本公开的另一些实施例中,各个参与节点12可通过事实性监测的方式来确定闲赋算力资源。每个参与节点12对该参与节点12的算力资源的使用情况进行实时监测以确定该参与节点12实时的闲赋算力资源。使用情况可包括算力资源的分配情况、使用率和闲置时长。在一个示例中,可通过使用监控工具对中央处理器(CPU)和内存等资源进行实时监测,记录服务器的各种资源使用情况,并可以提供灵活数据模型和查询语言,可以监测和分析CPU和内存等资源使用情况。此外,也可以在监控工具的基础上,开发高效的资源管理系统来追踪和管理算力资源,包括跟踪算力资源的分配情况、使用率和闲置时长,通过分析资源管理系统的确定哪些资源以及多少资源处于闲置状态。

[0047] 参与节点12可以以容器的形式隔离闲赋算力资源。如果参与节点12的算力资源的实时使用量增加,则参与节点12可立即释放容器中未使用的资源。如果参与节点12的算力资源的实时使用量降低,则参与节点12可立即增加容器中的资源。这样,参与节点12可弹性地配置闲赋算力资源,保证本地的计算任务能够被及时执行且本地的闲赋算力资源不会被闲置。

[0048] 在本公开的一些实施例中,作为算力租用方的参与节点12可向其直接连接的枢纽节点11提交算力资源买单。算力资源买单可包括:关于期望的算力资源的价格、数量、性能要求、任务描述。

[0049] 如果第一子网中的第一枢纽节点接收到第一子网中作为算力租用方的参与节点12所提交的算力资源买单(在框S206处为“是”),第一枢纽节点可执行框S208至框S212处的操作。在上下文中,第一子网指的是数联网中的任意一个子网10。第一枢纽节点指的是任意一个子网10中的枢纽节点11。

[0050] 在框S208处,第一枢纽节点根据算力资源全景表确定数联网中拥有匹配算力资源买单的算力资源的候选参与节点。如上所述,算力资源全景表是相对静态的,变化不大。因此,可先借助算力资源全景表进行初步筛选,将拥有匹配算力资源买单的算力资源的参与节点12确定为候选参与节点,将不具有匹配算力资源买单的算力资源的参与节点12排除出去。如果一个参与节点12的总算力资源都不足以匹配算力资源买单,则该参与节点12的闲赋算力资源更难以匹配该算力资源买单。对于拥有海量参与节点12的数联网来说,这一步能够预先排除掉一些不可能的参与节点12,从而降低后续操作的运算量。

[0051] 在框S210处,第一枢纽节点根据出租算力全景表实时地从候选参与节点中确定匹配算力资源买单的目标算力出租方。第一枢纽节点可根据竞价和资源要求,综合考虑算力资源买单的价格和可用资源进行匹配。匹配算法可能会优先考虑价格,然后再考虑可用资源,也可以按照资源性能优先原则进行匹配。算力租用方可在算力资源买单中设置匹配优先级。

[0052] 基于竞价和资源匹配的撮合系统需要灵活的算法和规则来平衡竞价和资源的优先级,综合考虑市场需求、资源管理、撮合算法和交易执行等因素,以实现公平、高效和优化的交易匹配。

[0053] 在本公开的一些实施例中,第一枢纽节点可能会接收到来自多个参与节点的多个算力资源买单。每个算力资源买单对应1个任务。可通过联合求解下式来确定匹配每个算力资源买单的目标算力出租方:

$$[0054] \quad \begin{cases} \min \sum (c_{ij} \times x_{ij}) \\ \sum (w_{ij} \times x_{ij}) \geq d_i \\ \sum (w_{ij} \times x_{ij}) \leq R_j \end{cases}$$

[0055] 其中, x_{ij} 表示第j候选参与节点是否被第i任务选中(即,第j候选参与节点是否作为第i任务的目标算力出租方), $x_{ij}=1$ 表示选中, $x_{ij}=0$ 表示未选中, c_{ij} 表示第j候选参与节点的算力资源分配给第i任务的成本, d_i 表示第i任务的总算力资源需求, w_{ij} 表示第j候选参与节点分配给第i任务的算力资源量, R_j 表示第j候选参与节点的总出租算力资源,第i任务是第i算力资源买单对应的任务。 $1 \leq i \leq p, 1 \leq j \leq q$, p 表示第一枢纽节点接收到的算力资源买单的总数, q 表示候选参与节点的总数。

[0056] $\min \sum (c_{ij} \times x_{ij})$ 用于最小化算力出租的总成本。 $\sum (w_{ij} \times x_{ij}) \geq d_i$ 用于保证每个任务的算力资源需求得到满足。 $\sum (w_{ij} \times x_{ij}) \leq R_j$ 用于保证每个候选参与节点的总出租算力资源不被超出。

[0057] 在框S212处,第一枢纽节点与目标算力出租方所直接连接的目标枢纽节点通过签约的方式构建算力租用关系。在本公开的一些实施例中,目标枢纽节点在完成与第一枢纽节点的签约时立即冻结目标算力出租方被出租的算力资源,以免该算力资源被分配给其它算力租用方。

[0058] 在签约之后,算力出租方执行当前交易并根据交易结果进行算力资源分配。交易的执行可包括转移资金、转移算力资产使用权以及记录交易的完成。可进行基于区块链的账本管理。第一枢纽节点和目标枢纽节点可维护基于区块链技术的订单账本,对交易的成交信息进行记账(订单账本无法篡改),对临时算力使用权进行转移直至交易完成。算力资源分配过程可能涉及算力库存调整、资金转移或其它资源的重新分配。

[0059] 在本公开的一些实施例中,每个算力出租方以容器的形式隔离待出租的闲赋算力资源。如果目标算力出租方的数量大于1,则将所有目标算力出租方中的容器组成容器集群(容器集群可以由多个主机组成,每个主机上运行多个容器,可使用容器编排工具将镜像部署到容器集群中),并且从各个目标算力出租方所直接连接的目标枢纽节点中选举一个主枢纽节点。在执行算力租用方的指令的过程中(交易过程中),主枢纽节点实时监测容器集群的负载情况。如果容器集群的负载减少,主枢纽节点缩减容器集群的规模,停止不再需要

的容器实例,释放算力资源以降低成本。如果容器集群的负载增加,主枢纽节点扩展容器集群的规模。这可以通过增加容器实例的数量来实现容器编排工具能够自动在可用的主机上启动新的容器实例。这样可以实现算力资源的弹性整合。

[0060] 在本公开的一些实施例中,在容器发生故障或不可用的情况下重启容器,确保应用程序的可用性。如果目标算力出租方的数量等于1,目标算力出租方所直接连接的目标枢纽节点收集和管理目标算力出租方中的容器的日志数据,以及在容器发生故障或不可用的情况下重启容器。如果目标算力出租方的数量大于1,主枢纽节点收集和管理容器集群中的各个容器的日志数据,以及在任一容器发生故障或不可用的情况下重启容器。集中收集和管理容器日志,能够便于进行故障排除和监控分析。日志数据可以帮助识别性能问题、异常行为和潜在故障等。

[0061] 通过使用容器技术,结合容器编排工具和自动化管理系统,可以实现基于容器的弹性整合算力。这种方法可以快速、灵活地扩展和缩减算力资源,提高应用程序的可用性和性能,并减少算力资源的浪费。同时,由于容器的轻量级和可移植性,容器化的应用程序可以在不同的环境中部署和运行,实现跨平台的弹性整合算力。

[0062] 在本公开的一些实施例中,枢纽节点11还可以进行算力资源监控。算力资源监控是指对计算资源的使用情况进行实时监测和记录,以便进行性能分析、资源优化和故障排除等操作。在弹性整合算力的背景下,算力资源监控可以帮助实时了解容器集群或服务器集群的负载情况。

[0063] 算力资源监控指标可包括:CPU使用率、内存消耗、网络流量、存储空间使用、容器或进程状态。对CPU使用率的监测可了解计算资源的负载情况。对计算资源的内存使用情况(内存消耗)的监测可确保足够的可用内存供应用程序使用。对计算资源的网络流量(包括入口流量和出口流量)的监测能够优化网络性能和带宽需求。对计算资源的存储空间使用情况的监测可包括硬盘或存储卷的空间占用和可用空间。对容器或进程的状态(包括运行状态、启动时间、停止时间等)的监测能够便于及时发现异常和故障。

[0064] 为了实现算力资源监控,可以使用各种监控工具和平台。这些工具可以收集、存储和展示监控数据。

[0065] 基于区块链的存证是利用区块链技术的不可篡改性和去中心化的特点,将数据的存储和验证与区块链上的交易记录结合起来。基于区块链的存证可以确保数据的完整性、可信性和不可变性,使得数据的来源和内容可以被有效地验证和追溯。

[0066] 在存证过程中,数据的摘要或哈希值通常被计算出来,并存储在区块链上的交易中。任何对数据进行更改或篡改的尝试都会改变数据的摘要,从而在区块链上留下痕迹,使得数据的完整性得以验证。此外,存证的时间戳也可以通过区块链的区块时间来确认。

[0067] 为了提高参与节点12在算力撮合过程中的参与度,本公开的一些实施例还提出了算力贡献度的概念。算力贡献度是指在算力共享网络中,每个参与节点12对参与计算任务的计算能力所做的贡献程度。算力贡献度根据参与节点12所提供的计算资源的数量和质量来衡量。

[0068] 在本公开的一些实施例中,第二枢纽节点计算与第二枢纽节点直接连接的所有第二参与节点的算力贡献度。在本文中,第二枢纽节点指的是数联网中的任意一个枢纽节点11。第二参与节点指的是与该任意一个枢纽节点11直接连接的参与节点12。第二枢纽节点

根据每个第二参与节点的算力贡献度来计算该第二参与节点的贡献度占比。然后,第二枢纽节点按照每个第二参与节点的贡献度占比来向该第二参与节点分配奖励以提高该第二参与节点在算力撮合过程中的参与度。

[0069] 在一个示例中,第二参与节点中的第j节点的算力贡献度被计算为:

$$[0070] \quad \text{score}_j = (1 + W_{rc}) \times \left[\sum_i^n (W_{rai} \times R_{ai}) + W_{rt} \times \sum_i^m (W_{rqi} \times R_{qi}) \right]$$

[0071] 其中, score_j 表示第j节点的算力贡献度, R_{ai} 表示第j节点的n个算力资源中的第i算力资源的数量, W_{rai} 表示根据第i算力资源的类别所设置的权重(可根据实际应用具体设置), R_{qi} 表示第j节点的m个质量参数中的第i质量参数, W_{rqi} 表示针对第i质量参数设置的权重, W_{rt} 表示根据第j节点出租算力的时间确定的权重, W_{rc} 表示根据第j节点所参与的任务的完成率确定的权重。

[0072] 算力资源主要指CPU、GPU、存储、带宽等。 W_{rai} 的取值可根据实际应用具体设置。质量参数可包括性能指标和可靠性指标。性能指标可以包含响应时间、吞吐量、任务处理能力、并发处理能力等。可靠性指标可以包含可用性、容错性、可靠性、故障处理能力。长时间参与并提供计算资源的参与者更具贡献能力。因此参与节点12出租算力的时间越长则 W_{rt} 越大。在一个示例中,以近5次任务参与的耗时总和占总任务耗时比例来计算参与节点12出租算力的时间。节点所参与的任务的完成率可以指的是近5次的任务完成率。近期历史任务中某参与节点12的完成率越高,则可以得到算力贡献度加成。默认第一次参与任务时 W_{rc} 的值为0,如果完成率低于0.5,则 W_{rc} 为-0.5;否则 W_{rc} 为0.5。

[0073] 在一个示例中,第j节点的贡献度占比被计算为:

$$[0074] \quad \text{ratio}_j = \frac{e^{\text{score}_j}}{\sum_j^K e^{\text{score}_j}}$$

[0075] 其中, ratio_j 表示第j节点的贡献度占比,K表示第二参与节点的节点数量, score_j 表示第j节点的算力贡献度。

[0076] 参与节点12的贡献度占比与奖励机制相关。参与节点12的贡献度越高,则会获得更多的奖励。例如,参与节点12的贡献度越高,则该参与节点12可优先匹配算力资源买单或算力资源卖单,或者该参与节点12可优先匹配价格更合适的算力资源买单或算力资源卖单。这种奖励机制可以激励参与节点12提供更多的计算资源,从而维护网络的稳定性和安全性,形成良性循环的数联网交易生态,帮助数联网健康运行。

[0077] 此外,在算力租用方与目标算力出租方进行算力交易的过程中,算力租用方与第一枢纽节点、第一枢纽节点与目标枢纽节点、目标枢纽节点与目标算力出租方之间的通信均遵循安全计算的原则,对于算力交易中所涉及的数据均执行隐私保护计算。

[0078] 在本公开的其它实施例中,还提供了一种存储有计算机程序的计算机可读存储介质,其中,计算机程序在由处理器执行时能够实现如图2所示的方法的步骤。

[0079] 综上所述,根据本公开的实施例的用于数联网的算力撮合方法能够发现数联网中的闲赋算力资源,进行闲赋算力资源的高效弹性整合,快速准确地在拥有海量参与节点12的数联网中实现算力撮合,在单个算力资源买单的需求较大的情况下能够高效地组合多个参与节点12的算力资源来满足单个算力资源买单的需求。此外,根据本公开的实施例的用于数联网的算力撮合方法还基于算力贡献度来设置鼓励机制,从而提高参与节点12在算力

撮合过程中的参与度,维护网络的稳定性和安全性,形成良性循环的数联网交易生态,帮助数联网健康运行。

[0080] 附图中的流程图和框图显示了根据本公开的多个实施例的装置和方法的可能实现的体系架构、功能和操作。在这点上,流程图或框图中的每个方框可以代表一个模块、程序段或指令的一部分,所述模块、程序段或指令的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。在有些作为替换的实现中,方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如,两个连续的方框实际上可以基本并行地执行,它们有时也可以按相反的顺序执行,这依所涉及的功能而定。也要注意的,框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合,可以用执行规定的功能或动作的专用的基于硬件的系统来实现,或者可以用专用硬件与计算机指令的组合来实现。

[0081] 除非上下文另外明确地指出,否则在本文和所附权利要求中所使用的词语的单数形式包括复数,反之亦然。因而,当提及单数时,通常包括相应术语的复数。相似地,措辞“包含”和“包括”将解释为包含在内而不是独占性地。同样地,术语“包括”和“或”应当解释为包括在内的,除非本文中明确禁止这样的解释。在本文中使用术语“示例”之处,特别是当其位于一组术语之后时,所述“示例”仅仅是示例性的和阐述性的,且不应当被认为是独占性的或广泛性的。

[0082] 适应性的进一步的方面和范围从本文中提供的描述变得明显。应当理解,本申请的各个方面可以单独或者与一个或多个其它方面组合实施。还应当理解,本文中的描述和特定实施例旨在仅说明的目的并不旨在限制本申请的范围。

[0083] 以上对本公开的若干实施例进行了详细描述,但显然,本领域技术人员可以在不脱离本公开的精神和范围的情况下对本公开的实施例进行各种修改和变型。本公开的保护范围由所附的权利要求限定。

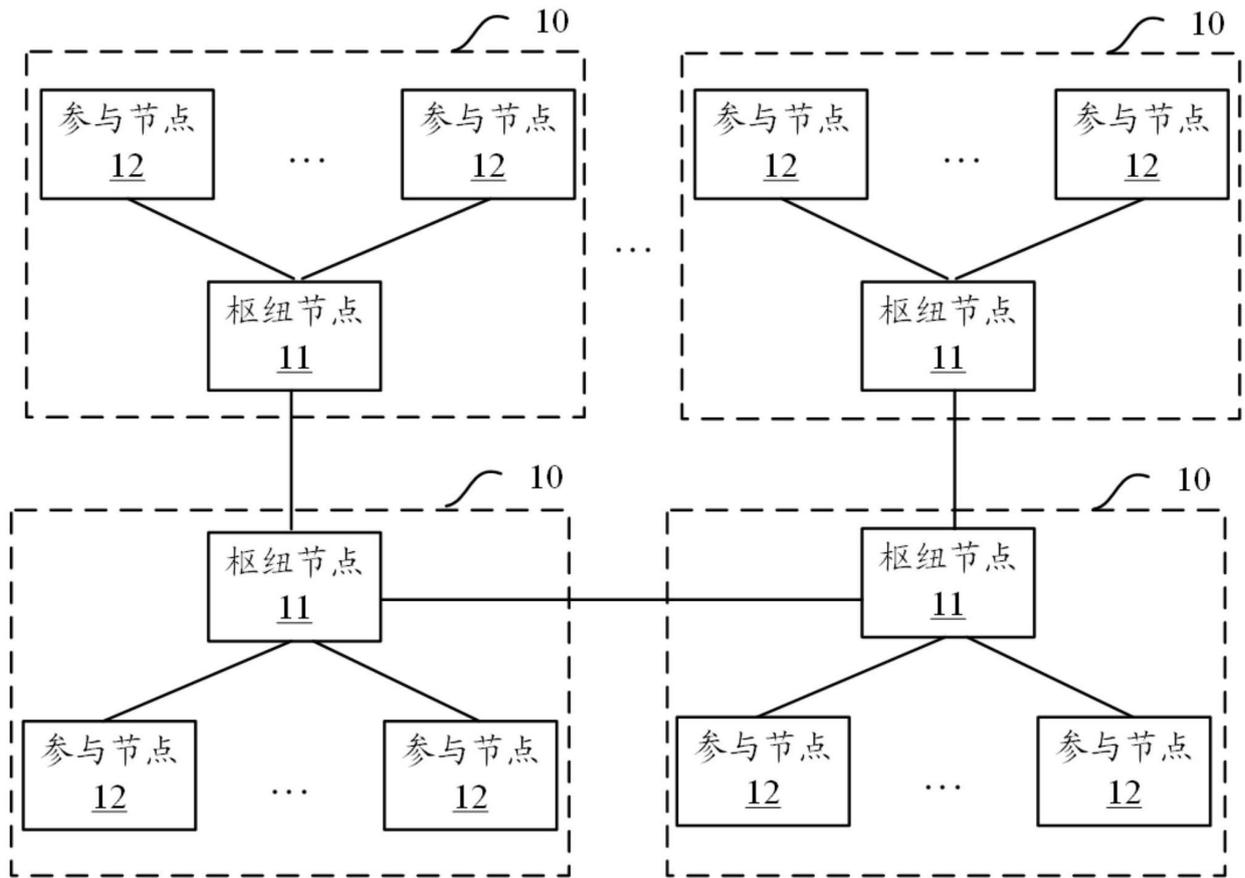


图1

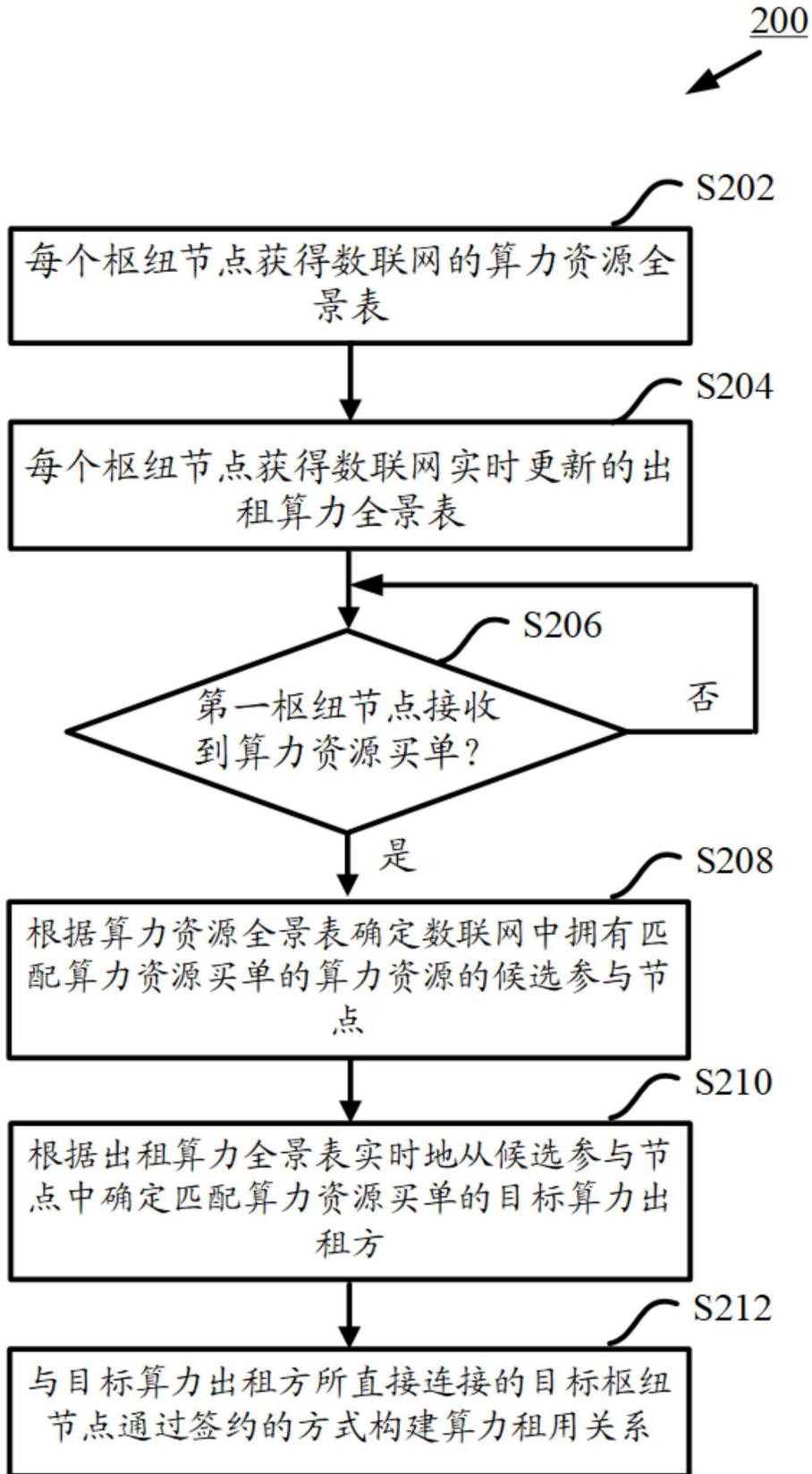


图2