



(12) 发明专利

(10) 授权公告号 CN 113067899 B

(45) 授权公告日 2021.07.30

(21) 申请号 202110611553.0

H04L 12/12 (2006.01)

(22) 申请日 2021.06.02

G06Q 40/04 (2012.01)

(65) 同一申请的已公布的文献号

申请公布号 CN 113067899 A

(56) 对比文件

CN 108200210 A, 2018.06.22

CN 109639550 A, 2019.04.16

(43) 申请公布日 2021.07.02

CN 111355780 A, 2020.06.30

(73) 专利权人 支付宝(杭州)信息技术有限公司

CN 112887160 A, 2021.06.01

地址 310000 浙江省杭州市西湖区西溪路

US 2018121909 A1, 2018.05.03

556号8层B段801-11

谷宁静. 基于区块链的电子政务数据共享设计研究.《信息安全与通信保密》.2020,(第4期),

(72) 发明人 陶友贤

审查员 徐苏宁

(74) 专利代理机构 北京博思佳知识产权代理有限公司 11415

代理人 李威

(51) Int. Cl.

H04L 29/08 (2006.01)

H04L 12/44 (2006.01)

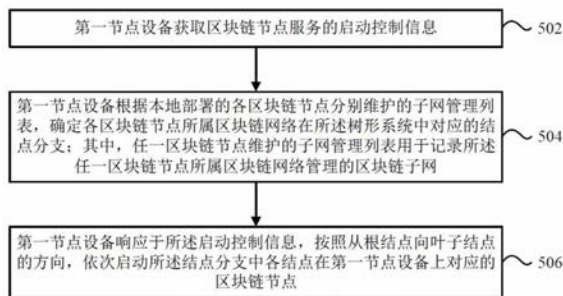
权利要求书4页 说明书29页 附图6页

(54) 发明名称

一种启动/关闭区块链节点服务的方法和装置

(57) 摘要

本说明书一个或多个实施例提供一种启动/关闭区块链节点服务的方法,应用于以区块链主网为根结点、各个区块链子网分别为其他结点的树形系统,树形系统中的任一区块链子网由其父结点对应的区块链网络所管理,启动区块链节点服务的方法包括:第一节点设备获取区块链节点服务的启动控制信息;第一节点设备根据本地部署的各区块链节点分别维护的子网管理列表,确定各区块链节点所属区块链网络在树形系统中对应的结点分支;其中,任一区块链节点维护的子网管理列表用于记录任一区块链节点所属区块链网络管理的区块链子网;第一节点设备响应于启动控制信息,按照从根结点向叶子结点的方向,依次启动所述结点分支中各结点在第一节点设备上对应的区块链节点。



1. 一种启动区块链节点服务的方法,应用于以区块链主网为根结点、各个区块链子网分别为其他结点的树形系统,所述树形系统中的任一区块链子网由其父结点对应的区块链网络所管理,所述方法包括:

第一节点设备获取区块链节点服务的启动控制信息;

第一节点设备根据本地部署的各区块链节点分别维护的子网管理列表,确定各区块链节点所属区块链网络在所述树形系统中对应的结点分支;其中,任一区块链节点维护的子网管理列表用于记录所述任一区块链节点所属区块链网络管理的区块链子网;

第一节点设备响应于所述启动控制信息,按照从根结点向叶子结点的方向,依次启动所述结点分支中各结点在第一节点设备上对应的区块链节点。

2. 根据权利要求1所述的方法,所述启动控制信息中包含第一节点设备对应的身份信息。

3. 根据权利要求1所述的方法,所述第一节点设备启动任一区块链网络中的区块链节点,包括:

第一节点设备在所述任一区块链网络的运行状态为开启状态的情况下,启动所述任一区块链网络中的区块链节点。

4. 根据权利要求3所述的方法,还包括:

第一节点设备通过查询所述任一区块链网络的父结点对应的区块链网络所维护的子网管理列表,确定所述任一区块链网络的运行状态。

5. 根据权利要求1所述的方法,还包括:

在第一节点设备上部署有第一区块链网络中的区块链节点的情况下,第一节点设备若确定第二节点设备部署有第一区块链网络所管理的第二区块链网络中的区块链节点、且未部署第一区块链网络中的区块链节点,则向第二节点设备提供所述启动控制信息,以指示第二节点设备响应于所述启动控制信息而启动第二区块链网络中的区块链节点。

6. 根据权利要求1所述的方法,所述第一节点设备启动任一区块链节点,包括:

第一节点设备启动用于构成所述任一区块链节点的插件模块。

7. 根据权利要求6所述的方法,所述启动用于构成所述任一区块链节点的插件模块,包括:通过所述任一区块链节点对应的插件管理器启动用于构成所述任一区块链节点的插件模块。

8. 根据权利要求7所述的方法,

所述插件管理器为第一节点设备上部署的所有区块链节点对应的全局插件管理器;或者,

所述插件管理器为单独应用于所述任一区块链节点的独立插件管理器。

9. 根据权利要求6所述的方法,所述启动用于构成所述任一区块链节点的插件模块,包括:

确定用于构成所述任一区块链节点各插件模块的插件描述信息,以及各插件模块的运行需求;

在任一插件模块的运行需求为共享运行的情况下,若存在匹配于所述任一插件模块的插件描述信息的已开启插件模块,且所述已开启插件模块的运行模式为共享模式,则将所述已开启插件模块共享至所述任一区块链节点;

在所述任一插件模块的运行需求为共享运行的情况下,若不存在匹配于所述插件描述信息且运行模式为共享模式的已开启插件模块,则启动所述任一插件模块;

在所述任一插件模块的运行需求为独立运行的情况下,启动所述任一插件模块。

10. 根据权利要求9所述的方法,所述各插件模块的运行需求被定义于所述任一区块链节点对应的插件配置信息中。

11. 根据权利要求9所述的方法,所述各插件模块的运行需求为全局预定义信息。

12. 根据权利要求9所述的方法,还包括:在启动所述插件描述信息对应的插件模块时,基于所述运行需求设置所启动的插件模块的运行模式。

13. 根据权利要求6所述的方法,所述启动用于构成所述任一区块链节点的插件模块,包括:根据用于构成所述任一区块链节点各插件模块之间的依赖关系,依次启动用于构成所述任一区块链节点各插件模块。

14. 一种关闭区块链节点服务的方法,应用于以区块链主网为根结点、各个区块链子网分别为其他结点的树形系统,所述树形系统中的任一区块链子网由其父结点对应的区块链网络所管理,所述方法包括:

第一节点设备获取区块链节点服务的关闭控制信息;

第一节点设备根据本地部署的各区块链节点分别维护的子网管理列表,确定各区块链节点所属区块链网络在所述树形系统中对应的结点分支;其中,任一区块链节点维护的子网管理列表用于记录所述任一区块链节点所属区块链网络管理的区块链子网;

第一节点设备响应于所述关闭控制信息,按照从叶子结点向根结点的方向,依次关闭所述结点分支中各结点在第一节点设备上对应的区块链节点。

15. 根据权利要求14所述的方法,所述关闭控制信息中包含第一节点设备对应的身份信息。

16. 根据权利要求14所述的方法,所述第一节点设备关闭任一区块链网络中的区块链节点,包括:

第一节点设备在所述任一区块链网络的运行状态为开启状态的情况下,关闭所述任一区块链网络中的区块链节点。

17. 根据权利要求16所述的方法,还包括:

第一节点设备通过查询所述任一区块链网络的父结点对应的区块链网络所维护的子网管理列表,确定所述任一区块链网络的运行状态。

18. 根据权利要求14所述的方法,还包括:

在第一节点设备上部署有第一区块链网络中的区块链节点的情况下,第一节点设备若确定第二节点设备部署有第一区块链网络所管理的第二区块链网络中的区块链节点、且未部署第一区块链网络中的区块链节点,则向第二节点设备提供所述关闭控制信息,以指示第二节点设备响应于所述关闭控制信息而关闭第二区块链网络中的区块链节点。

19. 根据权利要求14所述的方法,所述第一节点设备关闭任一区块链节点,包括:

第一节点设备关闭用于构成所述任一区块链节点的插件模块。

20. 根据权利要求19所述的方法,所述关闭用于构成所述任一区块链节点的插件模块,包括:通过所述任一区块链节点对应的插件管理器关闭用于构成所述任一区块链节点的插件模块。

21. 根据权利要求20所述的方法，
所述插件管理器为第一节点设备上部署的所有区块链节点对应的全局插件管理器；或者，

所述插件管理器为单独应用于所述任一区块链节点的独立插件管理器。

22. 根据权利要求19所述的方法，所述关闭用于构成所述任一区块链节点的插件模块，包括：

确定用于构成所述任一区块链节点各插件模块的插件描述信息；

若存在匹配于任一插件模块的插件描述信息的已开启插件模块的运行模式为独立模式、或运行模式为共享模式但未被共享至所述任一区块链节点以外的其他区块链节点，则关闭所述已开启插件模块。

23. 根据权利要求22所述的方法，所述任一插件模块的运行模式由所述任一插件模块的运行需求所确定。

24. 根据权利要求23所述的方法，所述任一插件模块的运行需求被定义于所述任一区块链节点对应的插件配置信息中。

25. 根据权利要求23所述的方法，所述任一插件模块的运行需求为全局预定义信息。

26. 根据权利要求19所述的方法，所述关闭用于构成所述任一区块链节点的插件模块，包括：根据用于构成所述任一区块链节点各插件模块之间的依赖关系，依次关闭用于构成所述任一区块链节点各插件模块。

27. 一种启动区块链节点服务的装置，应用于以区块链主网为根结点、各个区块链子网分别为其他结点的树形系统，所述树形系统中的任一区块链子网由其父结点对应的区块链网络所管理，所述装置包括：

控制信息获取模块，用于使第一节点设备获取区块链节点服务的启动控制信息；

结点分支确定模块，用于使第一节点设备根据本地部署的各区块链节点分别维护的子网管理列表，确定各区块链节点所属区块链网络在所述树形系统中对应的结点分支；其中，任一区块链节点维护的子网管理列表用于记录所述任一区块链节点所属区块链网络管理的区块链子网；

节点启动模块，用于使第一节点设备响应于所述启动控制信息，按照从根结点向叶子结点的方向，依次启动所述结点分支中各结点在第一节点设备上对应的区块链节点。

28. 一种关闭区块链节点服务的装置，应用于以区块链主网为根结点、各个区块链子网分别为其他结点的树形系统，所述树形系统中的任一区块链子网由其父结点对应的区块链网络所管理，所述装置包括：

控制信息获取模块，用于使第一节点设备获取区块链节点服务的关闭控制信息；

结点分支确定模块，用于使第一节点设备根据本地部署的各区块链节点分别维护的子网管理列表，确定各区块链节点所属区块链网络在所述树形系统中对应的结点分支；其中，任一区块链节点维护的子网管理列表用于记录所述任一区块链节点所属区块链网络管理的区块链子网；

节点关闭模块，用于使第一节点设备响应于所述关闭控制信息，按照从叶子结点向根结点的方向，依次关闭所述结点分支中各结点在第一节点设备上对应的区块链节点。

29. 一种电子设备，包括：

处理器；

用于存储处理器可执行指令的存储器；

其中,所述处理器通过运行所述可执行指令以实现如权利要求1-26中任一项所述的方法。

30.一种计算机可读存储介质,其上存储有计算机指令,该指令被处理器执行时实现如权利要求1-26中任一项所述方法的步骤。

一种启动/关闭区块链节点服务的方法和装置

技术领域

[0001] 本说明书一个或多个实施例涉及区块链技术领域,尤其涉及一种启动/关闭区块链节点服务的方法和装置。

背景技术

[0002] 区块链技术构建在传输网络(例如点对点网络)之上。区块链网络中的节点利用链式数据结构来验证与存储数据,并采用分布式节点共识算法来生成和更新数据。在基于现有区块链网络系统组建新区块链网络的场景下,同一个节点设备上往往会部署并运行多个区块链网络节点,当用户希望启动或关闭节点设备时,将涉及到对多个相互之间具有管理关系的区块链网络节点的启动或关闭,而如果不能按照一定流程和规则进行,很可能造成节点设备的崩溃。

发明内容

[0003] 有鉴于此,本说明书一个或多个实施例提供一种启动/关闭区块链节点服务的方法、装置、电子设备及存储介质,本说明书一个或多个实施例提供技术方案如下:

[0004] 根据本说明书一个或多个实施例的第一方面,提供了一种启动区块链节点服务的方法,应用于以区块链主网为根结点、各个区块链子网分别为其他结点的树形系统,所述树形系统中的任一区块链子网由其父结点对应的区块链网络所管理,所述方法包括:

[0005] 第一节点设备获取区块链节点服务的启动控制信息;

[0006] 第一节点设备根据本地部署的各区块链节点分别维护的子网管理列表,确定各区块链节点所属区块链网络在所述树形系统中对应的结点分支;其中,任一区块链节点维护的子网管理列表用于记录所述任一区块链节点所属区块链网络管理的区块链子网;

[0007] 第一节点设备响应于所述启动控制信息,按照从根结点向叶子结点的方向,依次启动所述结点分支中各结点在第一节点设备上对应的区块链节点。

[0008] 根据本说明书一个或多个实施例的第二方面,提供了一种关闭区块链节点服务的方法,应用于以区块链主网为根结点、各个区块链子网分别为其他结点的树形系统,所述树形系统中的任一区块链子网由其父结点对应的区块链网络所管理,所述方法包括:

[0009] 第一节点设备获取区块链节点服务的关闭控制信息;

[0010] 第一节点设备根据本地部署的各区块链节点分别维护的子网管理列表,确定各区块链节点所属区块链网络在所述树形系统中对应的结点分支;其中,任一区块链节点维护的子网管理列表用于记录所述任一区块链节点所属区块链网络管理的区块链子网;

[0011] 第一节点设备响应于所述关闭控制信息,按照从叶子结点向根结点的方向,依次关闭所述结点分支中各结点在第一节点设备上对应的区块链节点。

[0012] 根据本说明书一个或多个实施例的第三方面,提供了一种启动区块链节点服务的装置,应用于以区块链主网为根结点、各个区块链子网分别为其他结点的树形系统,所述树形系统中的任一区块链子网由其父结点对应的区块链网络所管理,所述装置包括:

[0013] 控制信息获取模块,用于使第一节点设备获取区块链节点服务的启动控制信息;

[0014] 结点分支确定模块,用于使第一节点设备根据本地部署的各区块链节点分别维护的子网管理列表,确定各区块链节点所属区块链网络在所述树形系统中对应的结点分支;其中,任一区块链节点维护的子网管理列表用于记录所述任一区块链节点所属区块链网络管理的区块链子网;

[0015] 节点启动模块,用于使第一节点设备响应于所述启动控制信息,按照从根结点向叶子结点的方向,依次启动所述结点分支中各结点在第一节点设备上对应的区块链节点。

[0016] 根据本说明书一个或多个实施例的第四方面,提供了一种关闭区块链节点服务的装置,应用于以区块链主网为根结点、各个区块链子网分别为其他结点的树形系统,所述树形系统中的任一区块链子网由其父结点对应的区块链网络所管理,所述装置包括:

[0017] 控制信息获取模块,用于使第一节点设备获取区块链节点服务的关闭控制信息;

[0018] 结点分支确定模块,用于使第一节点设备根据本地部署的各区块链节点分别维护的子网管理列表,确定各区块链节点所属区块链网络在所述树形系统中对应的结点分支;其中,任一区块链节点维护的子网管理列表用于记录所述任一区块链节点所属区块链网络管理的区块链子网;

[0019] 节点关闭模块,用于使第一节点设备响应于所述关闭控制信息,按照从叶子结点向根结点的方向,依次关闭所述结点分支中各结点在第一节点设备上对应的区块链节点。

[0020] 根据本说明书一个或多个实施例的第五方面,提供了一种电子设备,包括:

[0021] 处理器;用于存储处理器可执行指令的存储器;其中,所述处理器通过运行所述可执行指令以实现上述开启/关闭区块链节点服务方法的步骤。

[0022] 根据本说明书一个或多个实施例的第六方面,提供了一种计算机可读存储介质,其上储存有可执行指令;其中,该指令被处理器执行时,实现上述开启/关闭区块链节点服务方法的步骤。

[0023] 应当理解的是,以上的一般描述和后文的细节描述仅是示例性和解释性的,并不能限制本说明书。

附图说明

[0024] 图1是一示例性实施例提供的一种创建智能合约的示意图。

[0025] 图2是一示例性实施例提供的一种调用智能合约的示意图。

[0026] 图3是一示例性实施例提供的一种创建和调用智能合约的示意图。

[0027] 图4是一示例性实施例提供的一种基于区块链主网组建区块链子网的示意图。

[0028] 图5是一示例性实施例提供的一种启动区块链节点服务的方法的流程图。

[0029] 图6是一示例性实施例提供的一种关闭区块链节点服务的方法的流程图。

[0030] 图7是一示例性实施例提供的一种设备的结构示意图。

[0031] 图8是一示例性实施例提供的一种启动区块链节点服务的装置的框图。

[0032] 图9是一示例性实施例提供的一种关闭区块链节点服务的装置的框图。

具体实施方式

[0033] 这里将详细地对示例性实施例进行说明,其示例表示在附图中。下面的描述涉及

附图时,除非另有表示,不同附图中的相同数字表示相同或相似的要素。以下示例性实施例中所描述的实施方式并不代表与本说明书一个或多个实施例相一致的所有实施方式。相反,它们仅是与如所附权利要求书中所详述的、本说明书一个或多个实施例的一些方面相一致的装置和方法的例子。

[0034] 需要说明的是:在其他实施例中并不一定按照本说明书示出和描述的顺序来执行相应方法的步骤。在一些其他实施例中,其方法所包括的步骤可以比本说明书所描述的更多或更少。此外,本说明书中所描述的单个步骤,在其他实施例中可能被分解为多个步骤进行描述;而本说明书中所描述的多个步骤,在其他实施例中也可能被合并为单个步骤进行描述。

[0035] 区块链一般被划分为三种类型:公有链(Public Blockchain),私有链(Private Blockchain)和联盟链(Consortium Blockchain)。此外,还有多种类型的结合,比如私有链+联盟链、联盟链+公有链等不同组合形式。其中去中心化程度最高的是公有链。公有链以比特币、以太坊为代表,加入公有链的参与者可以读取链上的数据记录、参与交易以及竞争新区块的记账权等。而且,各参与者(即节点)可自由加入以及退出网络,并进行相关操作。私有链则相反,该网络的写入权限由某个组织或者机构控制,数据读取权限受组织规定。简单来说,私有链可以为一个弱中心化系统,参与节点具有严格限制且少。这种类型的区块链更适用于特定机构内部使用。联盟链则是介于公有链以及私有链之间的区块链,可实现“部分去中心化”。联盟链中各个节点通常有与之相对应的实体机构或者组织;参与者通过授权加入网络并组成利益相关联盟,共同维护区块链运行。

[0036] 不论是公有链、私有链还是联盟链,都可能提供智能合约的功能。区块链上的智能合约是在区块链系统上可以被交易触发执行的合约。智能合约可以通过代码的形式定义。

[0037] 以以太坊为例,支持用户在以太坊网络中创建并调用一些复杂的逻辑,这是以太坊区别于比特币区块链技术的最大挑战。以太坊作为一个可编程区块链的核心是以太坊虚拟机(EVM),每个以太坊节点都可以运行EVM。EVM是一个图灵完备的虚拟机,这意味着可以通过它实现各种复杂的逻辑。用户在以太坊中发布和调用智能合约就是在EVM上运行的。实际上,虚拟机直接运行的是虚拟机代码(虚拟机字节码,下简称“字节码”)。部署在区块链上的智能合约可以是字节码的形式。

[0038] 例如图1所示,Bob将一个包含创建智能合约信息的交易发送到以太坊网络后,节点1的EVM可以执行这个交易并生成对应的合约实例。图1中的“0x6f8ae93...”代表了合约的地址,交易的data字段保存的可以是字节码,交易的to字段为空。节点间通过共识机制达成一致后,这个合约成功创建,并且可以在后续过程中被调用。合约创建后,区块链上出现一个与该智能合约对应的合约账户,并拥有一个特定的地址,合约代码将保存在该合约账户中。智能合约的行为由合约代码控制。换句话说,智能合约使得区块链上产生包含合约代码和账户存储(Storage)的虚拟账户。

[0039] 如图2所示,仍以以太坊为例,Bob将一个用于调用智能合约的交易发送到以太坊网络后,某一节点的EVM可以执行这个交易并生成对应的合约实例。图2中交易的from字段是交易发起方(即Bob)的账户的地址,to字段中的“0x6f8ae93...”代表了被调用的智能合约的地址,value字段在以太坊中是以太币的值,交易的data字段保存的调用智能合约的方法和参数。调用智能合约后,balance的值可能改变。后续,某个客户端可以通过某一区块链节

点(例如图2中的节点6)查看balance的当前值。智能合约以规定的方式在区块链网络中每个节点独立的执行,所有执行记录和数据都保存在区块链上,所以当交易完成后,区块链上就保存了无法篡改、不会丢失的交易凭证。

[0040] 创建智能合约和调用智能合约的示意图如图3所示。以太坊中要创建一个智能合约,需要经过编写智能合约、编译成字节码、部署到区块链等过程。以太坊中调用智能合约,是发起一笔指向智能合约地址的交易,智能合约代码分布式的运行在以太坊网络中每个节点的虚拟机中。

[0041] 需要说明的是,除了可以由用户创建智能合约,也可以在创世块中由系统设置智能合约。这类合约一般称为创世合约。一般的,创世合约中可以设置一些区块链网络的数据结构、参数、属性和方法。此外,具有系统管理员权限的账户可以创建系统级的合约,或者修改系统级的合约(简称为系统合约)。另外除了以太坊中的EVM外,不同的区块链网络还可能采用各种的虚拟机,这里并不限定。

[0042] 区块链网络中的节点在执行调用智能合约的交易后,会生成相应的收据(receipt),以用于记录与执行该智能合约相关的信息。这样,可以通过查询交易的收据来获得合约执行结果的相关信息。合约执行结果可以表现为收据中的事件(event)。消息机制可以通过收据中的事件实现消息传递,以触发区块链节点执行相应的处理。事件的结构譬如可以为:

[0043] Event:

[0044] [topic][data]

[0045] [topic][data]

[0046]

[0047] 在上述示例中,事件的数量可以为一个或多个;其中,每个事件分别包括主题(topic)和数据(data)等字段。区块链节点可以通过监听事件的topic,从而在监听到预定义的topic的情况下,执行预设处理,或者从相应事件的data字段读取相关内容,以及可以基于读取的内容执行预设处理。

[0048] 上述的事件机制中,相当于在监听方(比如存在监听需求的用户)处存在具有监听功能的客户端,譬如该客户端上运行了用于实现监听功能的SDK等,由该客户端对区块链节点产生的事件进行监听,而区块链节点只需要正常生成收据即可。除了上述的事件机制之外,还可以通过其他方式实现交易信息的透出。例如,可以通过在区块链节点运行的区块链平台代码中嵌入监听代码,使得该监听代码可以监听区块链交易的交易内容、智能合约的合约状态、合约产生的收据等其中的一种或多种数据,并将监听到的数据发送至预定义的监听方。由于监听代码部署于区块链平台代码中,而非监听方的客户端处,因而相比于事件机制而言,这种基于监听代码的实现方式相对更加的主动。其中,上述的监听代码可以由区块链平台的开发人员在开发过程中加入区块链平台代码,也可以由监听方基于自身的需求而嵌入,本说明书并不对此进行限制。

[0049] 区块链技术区别于传统技术的去中心化特点之一,就是在各个节点上进行记账,或者称为分布式记账,而不是传统的集中式记账。区块链系统要成为一个难以攻破的、公开的、不可篡改数据记录的去中心化诚实可信系统,需要在尽可能短的时间内做到分布式数据记录的安全、明确及不可逆。不同类型的区块链网络中,为了在各个记录账本的节点中保

持账本的一致,通常采用共识算法来保证,即前述提到的共识机制。例如,区块链节点之间可以实现区块粒度的共识机制,比如在节点(例如某个独特的节点)产生一个区块后,如果产生的这个区块得到其它节点的认可,其它节点记录相同的区块。再例如,区块链节点之间可以实现交易粒度的共识机制,比如在节点(例如某个独特的节点)获取一笔区块链交易后,如果这笔区块链交易得到其他节点的认可,认可该区块链交易的各个节点可以分别将该区块链交易添加至自身维护的最新版区块中,并且最终能够确保各个节点产生相同的最新版区块。共识机制是区块链节点就区块信息(或称区块数据)达成全网一致共识的机制,可以保证最新版区块被准确添加至区块链。当前主流的共识机制包括:工作量证明(Proof of Work,POW)、股权证明(Proof of Stake,POS)、委任权益证明(Delegated Proof of Stake,DPoS)、实用拜占庭容错(Practical Byzantine Fault Tolerance,PBFT)算法,HoneyBadgerBFT算法等。

[0050] 由于区块链网络的去中心化特性,使得区块链网络中的所有区块链节点均会维护相同的区块数据,无法满足部分节点的特殊需求。以联盟链为例,所有联盟成员(即联盟内的节点成员)可以组成一区块链网络,所有联盟成员在该区块链网络中分别存在对应的区块链节点,并可以通过对应的区块链节点获得该区块链网络上发生的所有交易和相关数据。但在一些情况下,可能存在部分联盟成员希望完成一些具有保密需求的交易,这些联盟成员既希望这些交易能够在区块链上存证或借助于区块链技术的其他优势,又能够避免其他联盟成员查看到这些交易和相关数据。虽然这些联盟成员可以额外组建一新的区块链网络,其建立方式与上述包含所有联盟成员的区块链网络类似,但是从头开始建立一条新的区块链网络需要消耗大量的资源,且无论是该区块链网络的建立过程或是建成后的配置过程都非常耗时。联盟成员之间的需求往往是临时的或者具有一定的时效性,使得新建的区块链网络很快就会由于需求消失而失去存在的意义,从而进一步增加了上述区块链网络的建链成本。而联盟成员之间的需求经常会变化,而每一需求所对应的联盟成员也往往不同,因而每当联盟成员发生变化时就可能需要组建一新的区块链网络,从而造成资源和时间的大量浪费。

[0051] 为此,可以将已组建的区块链网络作为区块链主网,并在该区块链主网的基础上组建区块链子网。那么,在诸如上述的联盟链场景下,联盟成员可以在已经参与区块链主网的情况下,基于自身需求而在区块链主网的基础上组建所需的区块链子网。由于区块链子网是在区块链主网的基础上所建立,使得区块链子网的组建过程相比于完全独立地组建一条区块链网络,所消耗的资源 and 所需的耗时等都极大地降低,灵活性极高。

[0052] 基于区块链主网快捷组建区块链子网的过程如下:区块链主网中的各主网节点分别获取组建区块链子网的交易以透出配置信息,当所述配置信息包含第一主网节点对应节点成员的身份信息时,部署第一主网节点的节点设备基于所述包含所述配置信息的创世块并启动属于所述区块链子网的第一子网节点。

[0053] 组建区块链子网的交易可由区块链主网的管理员发起,即仅允许管理员在区块链主网的基础上组建区块链子网,而避免将区块链子网的组建权限开放给普通用户,以防止由此导致的安全性问题。在一些情况下,也可以允许区块链主网的普通用户发起上述组建区块链子网的交易,以满足普通用户的组网需求,使得普通用户能够在管理员不便于发起交易的情况下依然能够快捷地组建区块链子网。

[0054] 以图4所示为例,区块链主网为subnet0,该subnet0包含的区块链节点为nodeA、nodeB、nodeC、nodeD和nodeE等。假定nodeA、nodeB、nodeC 和nodeD希望组建一区块链子网:如果nodeA为管理员且仅允许管理员发起组建区块链子网的交易,那么可由nodeA向subnet0发起上述组建区块链子网的交易;如果nodeE为管理员且仅允许管理员发起组建区块链子网的交易,那么nodeA~nodeD需要向nodeE进行请求,使得nodeE向subnet0发起上述组建区块链子网的交易;如果nodeE为管理员但允许普通用户发起组建区块链子网的交易,那么nodeA~nodeE均可以向subnet0发起上述组建区块链子网的交易。当然,不论是管理员或者普通用户,发起组建区块链子网的交易的区块链节点并不一定参与所组建的区块链子网,比如虽然最终由nodeA、nodeB、nodeC 和nodeD组建区块链子网,但可由nodeE向subnet0发起上述组建区块链子网的交易,而并不一定由nodeA~nodeD来发起该组建区块链子网的交易。

[0055] 在区块链主网的基础上组建区块链子网时,容易理解的是,会使得该区块链子网与区块链主网之间存在逻辑上的层次关系。比如在图4所示的subnet0上组建区块链子网subnet1时,可以认为subnet0处于第一层、subnet1处于第二层,subnet0为subnet1的父网,subnet1为subnet0的子网。并且区块链子网也可以组建对应的区块链子网,例如可以在图4中subnet1的基础上进一步组建另一区块链子网subnet3,此时可以认为subnet处于第三层,subnet1为subnet3对应的父网,subnet3为subnet1的子网,而subnet3则为subnet0的孙子网,同样的,subnet3仍然可以在其基础上新的组建区块链子网,使得各区块链网络之间构成这种多层次树形结构,而在本说明书中,任一区块链网络是由其对应的父网所管理,也即由组建该任一区块链网络的区块链网络所管理,因此在如图4这种由以区块链主网为根结点(根结点的层级最低)、各个区块链子网分别为其他结点的区块链网络树形系统中,任一结点代表的区块链子网由其父结点对应的区块链网络所管理,而作为特例,区块链主网为底层区块链网络时,区块链主网由区块链主网自身进行管理。本说明书中的区块链主网可以为底层区块链网络,底层区块链网络是指并非在其他区块链网络的基础上组建的区块链子网,因此除该区块链主网以外不存在其他区块链网络能够对区块链主网进行管理,比如图4中的subnet0可以认为属于底层区块链网络类型的区块链主网,subnet0管理subnet0自身,当然,区块链主网也可以为其他区块链网络的子网,本说明书对此不作任何限制。上述区块链网络树形系统通过父结点管理对应子结点的方式,实现了逐层管理,降低了区块链主网的管理压力,同时避免向下层网络暴露上层网络的子网信息,从而实现各级网络的隐秘管理。

[0056] 上述组建区块链子网的交易在被发送至区块链主网后,由区块链主网内的共识节点进行共识,并在通过共识后由各主网节点执行该交易,以完成区块链子网的组建。共识过程取决于所采用的共识机制,譬如上文所述的任一共识机制,本说明书并不对此进行限制。

[0057] 通过在上述组建区块链子网的交易中包含配置信息,该配置信息可以用于对所组建的区块链子网进行配置,使得组建的区块链子网符合组网需求。例如,通过在配置信息中包含节点成员的身份信息,可以指定组建的区块链子网包含哪些区块链节点。

[0058] 节点成员的身份信息可以包括节点的公钥,或者采用节点ID等其他能够表征节点身份的信息,本说明书并不对此进行限制。以公钥为例,每个区块链节点都存在对应的一组或多组公私钥对,由区块链节点持有私钥而公钥被公开且唯一对应于该私钥,因而可以通

过公钥来表征相应区块链节点的身份。因此,对于希望作为区块链子网的节点成员的区块链节点,可以将这些区块链节点的公钥添加至上述组建区块链子网的交易中,以作为上述节点成员的身份信息。上述的公私钥对可以用于签名验证的过程。例如,在采用有签名的共识算法中,譬如subnet1上述的nodeA1采用自身维护的私钥对消息进行签名后,将经过签名的消息在subnet1中广播,而nodeB1、nodeC1和nodeD1可以用nodeA1的公钥对收到的消息进行签名验证,以确认自身收到的消息确实来自nodeA1且没有经过篡改。

[0059] 第一主网节点可以为区块链主网上属于配置信息所指示的节点成员的区块链节点。在组建区块链子网时,并非由第一主网节点直接参与组建区块链子网、成为其节点成员,而是需要由用于部署该第一主网节点的节点设备生成第一子网节点,并由第一子网节点成为区块链子网中的节点成员。第一主网节点和第一子网节点对应于同一个区块链成员,比如在联盟链场景下对应于同一联盟链成员,但第一主网节点属于区块链主网、第一子网节点属于区块链子网,使得该区块链成员可以分别参与到区块链主网和区块链子网的交易中;并且,由于区块链主网和区块链子网属于相互独立的两个区块链网络,使得第一主网节点生成的区块与第一子网节点生成的区块分别存入所述节点设备上的不同存储(采用的存储譬如可以为数据库),实现了第一主网节点与第一子网节点分别使用的存储之间的相互隔离,因而区块链子网所产生的数据仅会在区块链子网的节点成员之间同步,使得仅参与了区块链主网的区块链成员无法获得区块链子网上产生的数据,实现了区块链主网与区块链子网之间的数据隔离,满足了部分区块链成员(即参与区块链子网的区块链成员)之间的交易需求。

[0060] 可见,第一主网节点和第一子网节点是在逻辑上划分出来的区块链节点,而从物理设备的角度来说,相当于上述部署了第一主网节点和第一子网节点的节点设备同时参与了区块链主网和区块链子网。由于区块链主网与区块链子网之间相互独立,使得这两个区块链网络的身份体系也相互独立,因而即便第一主网节点和第一子网节点可以采用完全相同的公钥,仍然应当将两者视为不同的区块链节点。譬如在图4中,subnet0中的nodeA相当于第一主网节点,而部署该nodeA的节点设备生成了属于subnet1的nodeA1,该nodeA1相当于第一子网节点。可见,由于身份体系相互独立,所以即便第一子网节点所采用的公钥区别于第一主网节点,也不影响本说明书方案的实施。

[0061] 当然,区块链子网的节点成员并不一定只是区块链主网的部分节点成员。在一些情况下,区块链子网的节点成员可以与区块链主网的节点成员完全一致,此时所有的区块链成员都可以获得区块链主网和区块链子网上的数据,但是区块链主网与区块链子网所产生的数据依然可以相互隔离,比如可以通过在区块链主网上实现一类业务、在区块链子网上实现另一类业务,从而可以使得这两类业务分别产生的业务数据之间相互隔离。

[0062] 除了上述的节点成员的身份信息之外,配置信息还可以包括下述至少之一:所述区块链子网的网络标识、所述区块链子网的管理人员的身份信息、针对区块链平台代码的属性配置等,本说明书并不对此进行限制。网络标识用于唯一表征该区块链子网,因而该区块链子网的网络标识应当区别于区块链主网和该区块链主网上组建的其他区块链子网。区块链子网的管理人员的身份信息,譬如可以为作为管理人员的节点成员的公钥;其中,区块链主网与区块链子网的管理人员可以相同,也可以不同。

[0063] 通过区块链主网来组建区块链子网的优势之一,就是由于生成第一子网节点的节

点设备上已经部署了第一主网节点,因而可以将第一主网节点所使用的区块链平台代码复用在第一子网节点上,免去了区块链平台代码的重复部署,极大地提高了区块链子网的组建效率。那么,如果配置信息中未包含针对区块链平台代码的属性配置,第一子网节点可以复用第一主网节点上采用的属性配置;如果配置信息中包含了针对区块链平台代码的属性配置,第一子网节点可以采用该属性配置,使得第一子网节点所采用的属性配置不受限于第一主网节点的属性配置、与第一主网节点无关。针对区块链平台代码的属性配置可以包括下述至少之一:代码版本号、是否需要共识、共识算法类型、区块大小等,本说明书并不对此进行限制。

[0064] 组建区块链子网的交易包括调用合约的交易。该交易中可以指明被调用的智能合约的地址、调用的方法和传入的参数。例如,调用的合约可以为前述的创世合约或系统合约,调用的方法可以为组建区块链子网的方法,传入的参数可以包括上述的配置信息。在一实施例中,该交易可以包含如下信息:

```
[0065] from:Administrator
[0066] to:Subnet
[0067] method:AddSubnet(string)
[0068] string:genesis
```

[0069] 其中,from字段为该交易的发起方的信息,譬如Administrator表明该发起方为管理员;to字段为被调用的智能合约的地址,譬如该智能合约可以为Subnet合约,则to字段具体为该Subnet合约的地址;method字段为调用的方法,譬如在Subnet合约中用于组建区块链子网的方法可以为AddSubnet(string),而string为AddSubnet()方法中的参数,上述示例中通过genesis表征该参数的取值,该genesis具体为前述的配置信息。

[0070] 以Subnet0上的节点nodeA~nodeE执行调用Subnet合约中AddSubnet()方法的交易为例。在交易通过共识后,nodeA~nodeE分别执行AddSubnet()方法并传入配置信息,得到相应的执行结果。

[0071] 合约的执行结果可以包括所述配置信息,该执行结果可以处于前文所述的收据中,该收据中可以包含与执行AddSubnet()方法相关的event,即组网事件。组网事件的topic可以包含预定义的组网事件标识,以区别于其他的事件。譬如在与执行AddSubnet()方法相关的event中,topic的内容为关键词subnet,且该关键词区别于其他方法所产生event中的topic。那么,nodeA~nodeE通过监听生成的收据中各个event所含的topic,可以在监听到包含关键词subnet的topic的情况下,确定监听到与执行AddSubnet()方法相关的event,即组网事件。例如,收据中的event如下:

```
[0072] Event:
[0073] [topic:other][data]
[0074] [topic:subnet][data]
[0075] .....
```

[0076] 那么,nodeA~nodeE在监听到第1条event时,由于所含topic的内容为other,确定该event与AddSubnet()方法无关;以及,nodeA~nodeE在监听到第2条event时,由于所含topic的内容为subnet,确定该event与AddSubnet()方法相关,并进而读取该event对应的data字段,该data字段包含上述的配置信息。以配置信息包括区块链子网的节点成员的公

钥为例,data字段的内容例如可以包括:

```
[0077]   {subnet1;  
[0078]   nodeA的公钥,nodeA的IP、nodeA的端口号…;  
[0079]   nodeB的公钥,nodeB的IP、nodeB的端口号…;  
[0080]   nodeC的公钥,nodeC的IP、nodeC的端口号…;  
[0081]   nodeD的公钥,nodeD的IP、nodeD的端口号…;  
[0082]   }
```

[0083] 其中,subnet1为希望创建的区块链子网的网络标识。区块链主网中的各个区块链节点可以记录该区块链主网上已创建的所有区块链子网的网络标识,或者与这些区块链子网相关的其他信息,这些信息譬如可以维护在上述的Subnet合约中,具体可以对应于该Subnet合约所含的一个或多个合约状态的取值。那么,nodeA~nodeE可以根据记录的已创建的所有区块链子网的网络标识,确定上述的subnet1是否已经存在;如果不存在,说明subnet1是当前需要创建的新区块链子网,如果存在则说明subnet1已经存在。

[0084] 除了采用希望创建的新的区块链子网的网络标识之外,还可以采用预定义的新建网络标识,该新建网络标识表明相应的组网事件用于组建新的区块链子网。例如,可以将上述的subnet1替换为newsubnet,该newsubnet为预定义的新建网络标识,nodeA~nodeE在识别到data字段包含newsubnet时,即可确定包含该newsubnet的event为组网事件,需要创建新的区块链子网。

[0085] 除了网络标识subnet1之外,上述data字段中还包含各个节点成员的身份信息等内容。部署第一主网节点的节点设备可以监听生成的收据,并在监听到所述组网事件且所述组网事件的内容表明第一主网节点属于所述节点成员的情况下,由部署第一主网节点的节点设备获取所述组网事件包含的配置信息或创世块。例如,nodeA~nodeE在确定subnet1是需要新组建的区块链子网的情况下,会进一步识别data字段中包含的节点成员的身份信息,以确定自身的处理方式。比如,nodeA~nodeD会发现在data字段包含自身的公钥、IP地址和端口号等身份信息,假定nodeA~nodeD分别部署在节点设备1~4上,以nodeA和节点设备1为例:nodeA会触发节点设备1,使得节点设备1基于上述的消息机制从data字段获得配置信息并生成包含该配置信息的创世块,且节点设备1会在本地部署nodeA1,进而由nodeA1加载生成的创世块,从而形成为subnet1中的1个节点成员;类似地,nodeB会触发节点设备2生成nodeB1、nodeC会触发节点设备3生成nodeC1、nodeD会触发节点设备4生成nodeD1。以及,nodeE会发现data字段包含的身份信息与自身均不匹配,假定nodeE部署在节点设备5上,那么该节点设备5不会根据data字段中的配置信息生成创世块,也不会生成subnet1中的节点。

[0086] 如前所述,第一主网节点与第一子网节点并不一定采用相同的身份信息。因此,在上述实施例中,data字段中可以包含预先为nodeA1~nodeD1生成的身份信息,且区别于nodeA~nodeD的身份信息。仍以nodeA和节点设备1为例:节点设备1如果在data字段中发现了nodeA1的身份信息,可以生成创世块、部署nodeA1,并由nodeA1加载该创世块,或者,nodeA如果在data字段中发现了nodeA1的身份信息,那么nodeA会触发节点设备1生成创世块、部署nodeA1,并由nodeA1加载该创世块;nodeB~nodeD的处理方式类似,此处不再一一赘述。

[0087] 除了配置信息之外,合约的执行结果可以包括创世块。换言之,除了可以在data字段中包含配置信息,还可以直接在执行合约调用的过程中生成包含配置信息的创世块,从而将创世块包含于data字段中,那么对于上述的nodeA~nodeD而言,相应的节点设备1~4可以通过消息机制直接从data字段获得创世块,而无需自行生成,可以提升对nodeA1~nodeD1的部署效率。

[0088] 在本说明书中,组建区块链子网的交易可以并非是调用智能合约的交易,使得不支持智能合约的区块链网络也可以实现本说明书的技术方案,从而在区块链主网的基础上快捷地创建出区块链子网。例如,可以预先定义一组网交易类型标识,当交易包含该组网交易类型标识时,就表明该交易用于组建新的区块链子网,即该交易为组建区块链子网的交易。区块链平台代码可以包含相关的用于组件区块链子网的处理逻辑,使得运行该区块链平台代码的第一主网节点在执行交易时,如果发现该交易中包含上述的组网交易类型标识,且第一主网节点属于该交易中的配置信息所指示的节点成员,可以基于上述处理逻辑来触发部署第一主网节点的节点设备生成包含该配置信息的创世块并启动第一子网节点,由第一子网节点加载该创世块,以形成为区块链子网中的区块链节点。

[0089] 节点设备通过在该进程中创建一个运行区块链平台代码的实例,实现在该节点设备上部署一区块链节点。对于第一主网节点而言,由节点设备在上述进程中创建第一实例,并由该第一实例运行区块链平台代码而形成。类似地,对于第一子网节点而言,由节点设备在上述进程中创建区别于第一实例的第二实例,并由该第二实例运行区块链平台代码而形成。例如,节点设备可以首先在进程中创建第一实例,以形成区块链主网中的第一区块链节点;而当该节点设备对应的节点成员希望参与组建区块链子网时,可以在上述进程中创建第二实例,该第二实例区别于上述的第一实例,并由该第二实例形成区块链子网中的第二区块链节点。当第一实例与第二实例位于同一进程时,由于不涉及跨进程交互,可以降低对第一子网节点的部署难度、提高部署效率。当然,第二实例也可能与第一实例分别处于节点设备上的不同进程中,本说明书并不对此进行限制。事实上,本说明书实施例中涉及的任一节点设备上部署的各区块链节点均为运行在所述任一节点设备上的不同的区块链实例,任一节点设备上部署的各区块链节点生成的区块分别存入所述任一节点设备上的不同存储(例如数据库),且任一节点设备部署的各区块链节点分别使用的存储之间相互隔离;例如,节点设备可以在第一进程中创建第一实例,以形成区块链主网中的第一区块链节点;而当该节点设备对应的节点成员希望参与组建区块链子网时,可以启动区别于第一进程的第二进程,并在该第二进程中创建第二实例,该第二实例区别于上述的第一实例,进而由该第二实例形成区块链子网中的第二区块链节点。

[0090] 通过上述方式,可以在区块链主网上创建出区块链子网。以图4为例,subnet0原本包含nodeA~nodeE,而在subnet0的基础上可以组建出subnet1,该subnet1包含nodeA1~nodeD1,且nodeA与nodeA1、nodeB与nodeB1、nodeC与nodeC1、nodeD与nodeD1分别部署在同一节点设备上。类似地,还可以在subnet0上组建出subnet2或更多的区块链子网,其中subnet2包含nodeA2、nodeB2、nodeC2和nodeE2,且nodeA与nodeA1、nodeA2,nodeB与nodeB1、nodeB2,nodeC与nodeC1,nodeD与nodeD1,nodeE与nodeE2分别部署在同一节点设备上。以及,可以将subnet1、subnet2等作为区块链主网,并在此基础上进一步组建出区块链子网,例如在subnet1的基础上组建出区块链子网subnet3,其过程与subnet1或subnet2的组建相

似,仅仅是将区块链主网替换为区块链子网subnet1,此处不再赘述,最后得到subnet3包含nodeA3、nodeB3和nodeC3,使得且nodeA与nodeA1、nodeA2、nodeA3,nodeB与nodeB1、nodeB2、nodeB3,nodeC与nodeC1、nodeC2、nodeC3分别部署在同一节点设备上。

[0091] 通过上述方式来组建区块链子网,该区块链子网中包含的节点成员由前文所述的配置信息所确定。因此,在上述由区块链主网和多个区块链子网构成的多层次的树形系统场景下,同一节点设备上往往部署有多个区块链节点,例如图4中nodeA、nodeA1、nodeA2和nodeA3就部署在同一节点设备1上,那么当用户如管理员希望启动或关闭节点设备1的区块链节点服务时,就会涉及对节点设备1中部署的各区块链节点进行启动或关闭,具体而言,是需要控制节点设备1部署的中nodeA、nodeA1、nodeA2和nodeA3进行启动或关闭,而中nodeA、nodeA1、nodeA2和nodeA3分别对应的subnet0、subnet1、subnet2和subnet3具有一定的层级关系,体现为高层级的区块链网络受到相邻的低层级的区块链网络的管理,例如subnet1和subnet2受到subnet0的管理,subnet1仅受到subnet1的管理而不会受到subnet0的管理,使得subnet0可以控制启动或关闭subnet1和subnet2但不能控制subnet3,因此如果不按照一定的流程顺序对各区块链节点进行启动或关闭,可能会导致部分区块链节点无法正常受到管理,甚至导致节点设备崩溃。

[0092] 为此,本说明书提供了一种启动/关闭区块链节点服务的控制方法,能够在任一区块链子网由其父结点对应的区块链网络所管理的树形系统下,控制节点设备按照其所部署的各区块链节点所属区块链网络的管理顺序,依次开启/关闭各区块链节点,实现安全、无损地启动/关闭节点设备的区块链节点服务。

[0093] 图5是一示例性实施例提供的一种启动区块链节点服务的方法的流程图。如图5所示,该方法应用于以区块链主网为根结点、各个区块链子网分别为其他结点的树形系统,所述树形系统中的任一区块链子网由其父结点对应的区块链网络所管理,所述方法包括:

[0094] 步骤502,第一节点设备获取区块链节点服务的启动控制信息。

[0095] 第一节点设备可以部署有区块链主网或区块链子网对应的主网节点或子网节点,也可以部署有除区块链主网或区块链子网以外的其他区块链网络对应的其他区块链节点(即区别于所述树形系统中任一区块链网络对应的区块链节点),这种情况下,第一节点设备可以通过监听节点服务启动交易的收据的方式获取区块链节点服务的启动控制信息,需要指出的是,本说明书实施例中的区块链节点服务指的是预先确定的区块链主网和区块链子网所构成的区块链树形系统,因此虽然第一节点设备在开启区块链节点服务之前未启动相关的区块链主网或区块链子网对应的区块链节点,但第一节点设备可能部署有除区块链主网或区块链子网以外的其他区块链网络对应的其他区块链节点,且所述其他区块链节点处于开启状态,因此理论上第一节点设备在未启动区块链节点服务的情况下,也可以通过监听其他区块链节点上执行的节点服务启动交易的收据,从而获取区块链节点服务的启动控制信息。在一实施例中,部署有主网节点的节点设备均部署有所述其他区块链网络对应的其他区块链节点,因此,可以通过在所述其他区块链网络上发起节点服务启动交易,以使各节点设备均可以通过监听该交易的收据获取区块链节点服务的启动控制信息并控制主网节点和子网节点的启动,从而实现区块链主网和区块链子网构成的区块链树形系统整体的启动过程。

[0096] 第一节点设备还可以通过多种途径获取区块链节点服务的启动控制信息,例如,

可以直接通过接收所述树形系统的管理员发送的节点服务启动消息从而获取区块链节点服务的启动控制信息,也可以通过接收所述树形系统的普通用户发送的节点服务启动消息获取,本说明书实施例对此并不做任何限制。

[0097] 在所述启动控制信息中可以包含第一节点设备对应的身份信息,例如第一节点设备的IP地址、端口号或者所持有的用于表明自身身份的公钥。如前所述,启动节点交易的收据或者节点服务启动消息中可以包含身份信息,使得第一节点设备获取的区块链节点服务的启动控制信息中包含该身份信息,当第一节点设备通过检测所述启动控制信息中包含第一节点设备自身的身份信息时,才会进一步执行后续启动区块链节点服务的过程,从而使得在其他区块链网络上发起的节点服务启动交易或者对各节点设备进行广播的节点服务启动消息能够控制特定的节点设备启动区块链节点服务。

[0098] 步骤504,第一节点设备根据本地部署的各区块链节点分别维护的子网管理列表,确定各区块链节点所属区块链网络在所述树形系统中对应的结点分支;其中,任一区块链节点维护的子网管理列表用于记录所述任一区块链节点所属区块链网络管理的区块链子网。

[0099] 本说明书涉及的确定各区块链节点所属区块链网络在所述树形系统中对应的结点分支,实际上指的是确定第一节点设备本地部署的各区块链节点有哪些以及这些区块链节点之间的管理关系,该过程需要通过查找第一节点设备根据本地部署的各区块链节点分别维护的子网管理列表实现。第一节点设备首先查看本地维护的主网管理列表,其中,主网管理列表中记录有区块链主网中各区块链节点的身份信息,第一节点设备在自身维护的身份信息包含于主网管理列表中的情况下,确定本地部署有区块链主网对应的第一主网节点,然后查找第一主网节点维护的子网管理列表,其中,该列表中记录有区块链主网所管理的全部区块链子网的子网信息,包括子网标识、运行状态、节点成员身份信息,第一节点设备在自身维护的身份信息包含于该子网管理列表中的情况下,确定本地部署有区块链主网所管理的一个或多个区块链子网对应的区块链节点,从而通过查找这一个或多个区块链节点所维护的子网管理列表中是否包含第一节点设备自身维护的身份信息,进一步确定本地部署的区块链节点,以此类推,直至最新确定的本地部署的区块链节点没有维护相应的子网管理列表或相应子网管理列表中不包含第一节点设备自身维护的身份信息,停止上述查找和确定本地部署的区块链节点的过程,最后确定得到的各区块链节点所属的区块链网络即为所述树形系统中对应的结点分支,任一区块链节点维护的子网管理列表所记载的区块链网络对应于该任一区块链节点所属区块链网络的子结点。

[0100] 步骤506,第一节点设备响应于所述启动控制信息,按照从根结点向叶子结点的方向,依次启动所述结点分支中各结点在第一节点设备上对应的区块链节点。

[0101] 在本说明书实施例中,第一节点设备上部署的各区块链节点实质是运行在第一节点设备上且由各自插件模块构成的不同区块链实例,同时,第一节点设备部署有任一区块链节点对应的插件管理器,用于管理第一节点设备上部署的所述任一区块链节点(主网节点或子网节点)在运行时所依赖的插件模块,也即用于构成所述任一区块链节点的插件模块。因此,本说明书实施例涉及的第一节点设备启动或关闭任一区块链节点,是指第一节点设备可以通过控制所述任一区块链节点对应的插件管理器启动或关闭用于构成所述任一区块链节点的插件模块。

[0102] 本说明书实施例中,第一节点设备上部署的插件管理器维护有第一节点设备上部署的各区块链节点的运行状态,因此,在第一节点设备可以通过查看本地部署的各区块链节点对应的插件管理器获取该区块链节点的运行状态,从而确定该区块链节点是否已经启动完成。

[0103] 第一节点设备在依次启动所述结点分支中各节点对应的区块链节点时,需要等待任一区块链网络对应的区块链节点启动完成后,才会开始启动所述任一区块链网络的子结点(由所述任一区块链网络所管理的区块链子网)对应的区块链节点。例如,在第一节点设备所确定的结点分支包括作为树形系统根结点的区块链主网时,那么将首先响应所述启动控制信息并启动第一节点设备上部署的第一主网节点(需要确定第一节点设备本地部署有第一主网节点),第一主网节点启动后,会自动与其他区块链主网中的其他节点建立网络连接,当区块链主网中的所有主网节点均启动完成后,可以认为区块链主网启动完成,此时可以在区块链主网上发起交易,使得交易能够正常被共识和执行,但对于第一节点设备而言,当第一主网节点启动后,由于第一主网节点理论上已经携带有所有区块链主网的信息,因此第一节点设备可以通过发起本地交易的方式获取区块链主网中的各类信息,而无需等待区块链主网完全启动完成,所以第一节点设备在第一主网节点启动完成后就可以通过发起子网信息查询的本地交易来获取区块链主网所管理的区块链子网的子网信息。

[0104] 在一实施例中,第一节点设备本地预先存储有各区块链节点分别维护的子网管理列表,因此,可以预先确定第一节点设备本地部署的各区块链节点所属区块链网络在所述树形系统中对应的结点分支,从而在获取区块链节点服务的启动控制信息后,按照从根结点向叶子结点的方向,依次启动所述预先确定的结点分支中各结点在第一节点设备上对应的区块链节点。

[0105] 以图4为例,假定subnet0上nodeA~nodeE分别部署在节点设备1~5上,因而节点设备1除了部署有主网节点nodeA,还部署有属于subnet1的子网节点nodeA1、subnet2的子网节点nodeA2以及subnet3的子网节点nodeA3,节点设备1因此自身维护有本地已经部署的nodeA、nodeA1、nodeA2和nodeA3的身份信息,且对于节点设备1而言,其本地可以预先存储nodeA、nodeA1、nodeA2和nodeA3分别维护的共4张子网管理列表,其中,nodeA维护的子网管理列表中记录有subnet1和subnet2的子网标识及其包含的节点成员的成员身份,subnet1包含nodeA1~nodeD1,subnet2包含nodeA2、nodeB2、nodeC2和nodeE2,由于其中的nodeA1和nodeA2均属于节点设备1自身维护的身份信息,所以节点设备1可以确定自身同时属于subnet1和subnet2的节点成员,本地部署有nodeA1和nodeA2,且nodeA1和nodeA2分别所属的subnet1和subnet2受subnet0的管理;nodeA1维护的子网管理列表中记录有subnet3的子网标识及其包含的节点成员的身份信息,包括nodeA3~nodeC3,由于其中的nodeA3属于节点设备1自身维护的身份信息之一,所以节点设备1可以确定自身属于subnet3的节点成员,本地部署有nodeA3且nodeA3所属的subnet3受到subnet1的管理;nodeA2和nodeA3维护的子网管理列表为空。根据上述4张子网管理列表,可以整理得到由subnet0、subnet1、subnet2和subnet3构成的结点分支,其中,subnet1和subnet2为subnet0的子结点,subnet3为subnet1的子结点,因此,节点设备1按照根结点向叶子结点的方向确定“nodeA→nodeA1→nodeA3”和“nodeA→nodeA2”的节点启动顺序,其中,“nodeA→nodeA1→nodeA3”的启动过程和“nodeA→nodeA2”的启动过程彼此相互独立,并行处理。具体而言,节点设备1获取到区块链

节点服务的启动控制信息并确定上述节点启动顺序后,响应于所述启动控制信息,首先通过插件管理器检查自身是否部署有主网节点nodeA,在确定部署有nodeA的情况下,响应于所述启动控制信息而启动本地部署的nodeA,然后依然通过插件管理器检查nodeA是否启动完成,在nodeA启动完成的情况下,分别控制nodeA1和nodeA2对应的插件管理器启动nodeA1和nodeA2,在nodeA1启动完成后,再控制nodeA3对应的插件管理器启动nodeA3。

[0106] 而对于节点设备5而言,其本地部署有属于subnet0的nodeE和属于subnet2的nodeE2,节点设备5因此自身维护有本地已经部署的nodeE和nodeE2的身份信息,且其本地可以预先存储nodeE和nodeE2分别维护的共2张子网管理列表,其中,nodeE维护的子网管理列表中记录有subnet1和subnet2的子网标识及其包含的节点成员的成员身份,subnet1包含nodeA1~nodeD1,subnet2包含nodeA2、nodeB2、nodeC2和nodeE2,由于其中的nodeE2属于节点设备5自身维护的身份信息,所以节点设备5可以确定自身属于subnet2的节点成员,本地部署有nodeE2,且nodeE2所属的subnet2受subnet0的管理;nodeE2维护的子网管理列表为空。根据上述2张子网管理列表,可以整理得到由subnet0和subnet2构成的结点分支,其中 subnet2为subnet0的子结点,因此,节点设备5按照根结点向叶子结点的方向确定“nodeE→nodeE2”的节点启动顺序,并按照该节点启动顺序启动相应的区块链节点。类似的,节点设备2、3和4在获取到区块链节点服务的启动控制信息后也会执行上述流程以开启相应的主网节点和子网节点,使得节点设备完成开启区块链节点服务的过程。

[0107] 在另一实施例中,第一节点设备需要即时去获取各区块链节点分别维护的子网管理列表,从而边启动区块链节点边确定后续所需要的启动的区块链节点,在按照从根结点向叶子结点的方向依次启动区块链节点的过程中,逐渐确定第一节点设备本地部署的各区块链节点所属区块链网络在所述树形系统中对应的结点分支。

[0108] 以图4为例,假定subnet0上nodeA~nodeE分别部署在节点设备1~5上,因而节点设备1除了部署有主网节点nodeA,还部署有属于subnet1的子网节点nodeA1、subnet2的子网节点nodeA2以及subnet3的子网节点nodeA3,节点设备1因此自身维护有本地已经部署的nodeA、nodeA1、nodeA2和nodeA3的身份信息。对于节点设备1而言,在获取到区块链节点服务的启动控制信息后,响应于所述启动控制信息,首先通过插件管理器检查自身是否部署有主网节点nodeA,在确定部署有nodeA的情况下,响应于所述启动控制信息而启动本地部署的nodeA,然后依然通过插件管理器检查nodeA是否启动完成,在nodeA启动完成的情况下,查找nodeA维护的子网管理列表,该列表记录有subnet1和subnet2的子网标识及其包含的节点成员的成员身份,subnet1包含nodeA1~nodeD1,subnet2包含nodeA2、nodeB2、nodeC2和nodeE2,由于其中的nodeA1和nodeA2均属于节点设备1自身维护的身份信息,所以节点设备1可以确定自身同时属于subnet1和subnet2的节点成员,本地部署有nodeA1和nodeA2,且nodeA1和nodeA2分别所属的subnet1和subnet2受subnet0的管理,因此确定subnet1和sbunet2属于所述结点分支的一部分且均为subnet0对应的子结点,那么节点设备1则会并行地启动nodeA1和nodeA2,类似的,在nodeA1和nodeA2启动完成后,节点设备1会查找nodeA1和nodeA2分别维护的子网管理列表,其中nodeA1维护的子网管理列表中记录有subnet3的子网标识及其包含的节点成员的身份信息,包括nodeA3~nodeC3,由于其中的nodeA3属于节点设备1自身维护的身份信息之一,所以节点设备1可以确定自身属于subnet3的节点成员,本地部署有nodeA3且nodeA3所属的subnet3受到subnet1的管理,而

nodeA2维护的子网管理列表为空,因此确定subnet3属于所述结点分支的一部分且为subnet1对应的子结点,于是,节点设备1响应于所述启动控制信息而启动本地部署的nodeA3,最后,由于nodeA3所维护子网管理列表为空,因此停止搜寻和启动本地部署的区块链节点的过程,并最终可以整理得到由subnet0、subnet1、subnet2和subnet3构成的结点分支,其中,subnet1和subnet2为subnet0的子结点,subnet3为subnet1的子结点,并且,上述过程也在整体上完成了按照根结点向叶子结点的方向启动该节点分支中各结点在第一节点设备上对应的区块链节点的过程。

[0109] 如前所述,第一节点设备部署的任一区块链节点所维护的子网管理列表中还记录有第一节点设备本地部署的任一区块链节点所属区块链网络的子结点对应区块链子网的运行状态,如果任一区块链节点所维护的子网管理列表中虽然记录有某一子网节点所属区块链子网对应的子网标识,但该区块链子网对应的运行状态为关闭状态,这说明在任一区块链节点所属区块链网络的管理下,该子网节点所属的区块链子网被要求设置为关闭状态,因此可以不启动该子网节点,从而保证区块链节点的实际运行状态与任一区块链节点所属区块链网络的管理记录相一致,而如果该子网节点所属区块链子网对应的运行状态为开启状态,则说明该子网节点所属的区块链子网被要求设置为开启状态,此时可以启动该子网节点。通过在任一区块链网络的运行状态为开启状态的情况下,启动所述任一区块链网络中的区块链节点,可以确保正确、符合要求地启动第一节点设备区块链节点服务。

[0110] 以图4中的节点设备1为例,其本地部署的nodeA、nodeA1、nodeA2和nodeA3分别维护有子网管理列表,其中,nodeA维护的子网管理列表除了记录有subnet1和subnet2的子网标识及其包含的节点成员的成员身份,还记录有subnet1和subnet2的运行状态;nodeA1维护的子网管理列表中除了记录有subnet3的子网标识及其包含的节点成员的身份信息,还记录有subnet3的运行状态;nodeA2和nodeA3维护的子网管理列表为空。节点设备1根据上述4张子网管理列表,可以整理得到由subnet0、subnet1、subnet2和subnet3构成的结点分支以及subnet0、subnet1、subnet2和subnet3分别对应的运行状态,其中,subnet1和subnet2为subnet0的子结点,subnet3为subnet1的子结点,假设subnet0、subnet1和subnet3的运行状态为开启状态而subnet2的运行状态为关闭状态,因此,节点设备1根据树形系统所预期的启停要求只会启动subnet0、subnet1和subnet3而不会启动subnet2,并且按照根结点向叶子结点的方向,确定“nodeA→nodeA1→nodeA3”的节点启动顺序,并按照该节点启动顺序启动相应的区块链节点;或者,节点设备1首先根据自身维护的节点身份中包含nodeA确定本地部署有主网节点nodeA,此时节点设备1首先启动nodeA,在根据nodeA对应的插件管理器确定nodeA启动完成后,查找nodeA所维护的本地子网列表,确定subnet0所管理的subnet1和subnet2的子网标识、节点成员身份信息和运行状态,从而判断本地部署有subnet1的nodeA1和subnet2的nodeA2,假设subnet1的运行状态为开启状态而subnet2的运行状态为关闭状态,则节点设备1响应于所述启动控制信息只会启动subnet1对应的nodeA1,而不会对nodeA2做任何操作,以维持subnet2所应该处于的关闭状态,然后,在nodeA1启动完成后,进一步查找nodeA1维护的子网管理列表,确定subnet1所管理的subnet3的子网标识、节点成员身份信息和运行状态,从而判断本地部署有subnet3的nodeA3,假设subnet3的运行状态为开启状态,则节点设备1响应于所述启动控制信息启动subnet3对应的nodeA3。

[0111] 本说明书实施例中,第一节点设备本地部署的任一区块链节点所维护的子网管理列表可以由第一节点设备查询任一区块链节点所属区块链网络上部署的子网管理合约生成,具体而言,第一节点设备部署的任一区块链节点所维护的子网管理列表可以通过第一节点设备监听任一区块链节点所属区块链网络上的子网信息查询事件得到,所述子网信息查询事件通过任一区块链节点执行调用任一区块链节点所属区块链网络上子网管理合约的子网信息查询交易生成,所述子网信息查询事件包含任一区块链节点所属区块链网络管理下的各区块链子网的子网信息,所述子网信息包括子网标识、节点成员身份信息、运行状态和插件配置信息等,以图4为例,节点设备1在nodeA启动后,可以通过向subnet0上子网管理合约发起子网信息查询交易(本地交易,不会参与共识),并监听子网管理合约执行该交易后生成的子网信息查询事件,其topic关键词为ListSubnet,data字段包含subnet0管理下的各区块链子网的子网信息,包括subnet1和subnet2的子网标识、节点成员身份信息、运行状态、插件配置信息等,子网信息查询事件可以表示为如下形式:

[0112] [topic: ListSubnet][data]

[0113] 其被节点设备1监听到后,读取data字段中的内容(仅展示区块链子网的子网标识、节点成员身份信息和运行状态部分)为:

[0114] {subnet1:on;nodeA1,nodeB1,nodeC1,nodeD1;

[0115] subnet2:off;nodeA2,nodeB2,nodeC2,nodeE2;}

[0116] 其中,前缀subnet1、subnet2代表区块链子网的子网标识,冒号后的“on”或“off”代表运行状态,后缀nodeA1等代表子网节点的节点身份信息,例如节点公钥。节点设备1可以直接将data字段中的各区块链子网的子网信息作为subnet0对应的子网管理列表,或者,节点设备1将自身维护的各个节点的身份信息与上述data字段中的内容进行对比,找到自身所部署的子网节点所属的区块链子网,例如节点设备1维护有nodeA、nodeA1、nodeA2和nodeA3的身份信息,而data字段中相应子网节点对应的区块链子网的子网标识为subnet1和subnet2,因此节点设备1可以确定本地已经部署有subnet1和subnet2对应的子网节点,于是将subnet1与subnet2加入子网管理列表,在另一实施例中,节点设备1可以预先向subnet0上的子网管理合约发起所述子网信息查询交易,从而预先保存有nodeA维护的子网管理列表,因此节点设备1在nodeA启动后,便无需再向subnet0上的子网管理合约发起子网信息查询交易。所述树形系统中任一区块链网络上的子网管理合约的合约状态中维护有所述任一区块链网络管理的各区块链子网的子网合约状态,每一区块链子网对应的子网合约状态中记录有对应区块链子网的子网标识、节点成员的公钥和共识类型信息、插件配置信息、创世块等,事实上,子网管理合约可以与前文所述的Subnet合约为同一合约。

[0117] 或者,第一节点设备可以从本地部署的任一区块链节点所属区块链网络对应的数据库中读取所述子网管理合约的合约状态,并从读取到的合约状态包含的子网信息生成所述任一区块链节点所维护的子网管理列表,例如节点设备1可以直接从任一区块链节点对应的数据库中,读取到部署于所述任一区块链节点所属区块链网络的子网管理合约的合约状态,由于该合约状态中记载有任一区块链节点所属区块链网络管理下的各区块链子网的子网信息,因此节点设备1可以根据子网管理合约的合约状态以及节点设备1维护的nodeA1和nodeA2的身份信息,从而将与nodeA1和nodeA2分别对应的subnet1与subnet2对应的子网标识和运行状态分别加入nodeA维护的子网管理列表。上述子网管理合约也可以是前述的

Subnet合约,也可以是部署于任一区块链节点所属区块链网络上的其他智能合约。

[0118] 如前所述,第一节点设备可以通过查询本地部署的任一区块链节点所维护的子网管理列表,确定任一区块链节点所属区块链网络所管理的区块链网络的子网标识和运行状态,而在另一实施例中,当获取上述任一区块链节点所属区块链网络上子网信息查询事件的data字段后,第一节点设备也可以直接根据data字段中的各区块链子网的子网信息确定任一区块链节点所属区块链网络所管理的区块链网络的子网标识和运行状态,而无需利用或整理出nodeA所维护的子网管理列表,例如,节点设备1在获取区块链节点服务的启动控制信息并启动nodeA后,节点设备1会直接在上述data字段中进行查询,得到subnet1所对应的子网节点的身份信息为nodeA1、nodeB1、nodeC1和nodeD1,运行状态为开启状态,subnet2所对应的子网节点的身份信息为nodeA2、nodeB2、nodeC2和nodeE2,运行状态为关闭状态,然后将自身维护的身份信息nodeA1和nodeA2与之比对,判断自身维护的身份信息既包含于subnet1对应的子网节点成员,又包含于subnet2对应的子网节点成员,因此节点设备1可以确定本地部署有subnet1和subnet2分别对应的nodeA1和nodeA2,以及subnet1的运行状态为开启状态,subnet2的运行状态为关闭状态。

[0119] 第一区块链网络管理下的各区块链子网可以包括在第一区块链网络下组建的区块链子网,也可以包括不在第一区块链网络下组建的区块链子网。一般而言,在任一区块链网络上组建的区块链子网将受到该任一区块链网络的管理,例如在subnet1的基础上组建有subnet3,那么subnet1将获取subnet3的管理权,可以对subnet3的运行状态进行控制,而subnet0则无法对subnet3进行管理,这种场景下,节点设备会对不同区块链网络采用不同的权限管理,当监听到事件属于subnet0时,则节点设备在响应该事件完成各类操作时,只能读取subnet0对应权限的子网管理列表,该子网管理列表只会包含受subnet0管理的区块链子网的子网标识,即包含subnet1和subnet2,而不会包含subnet3。

[0120] 然而通过一些方法,可以实现subnet0的跨层管理,使得subnet0能够管理subnet3。例如,subnet0可以通过跨链技术将从subnet3的父结点subnet1处获得的有关subnet3的子网信息添加至subnet0上子网管理合约的合约状态中,从而使得subnet0获取subnet3的管理权,后续subnet0上的主网节点可以通过子网信息查询交易将subnet3的子网信息写入子网信息查询事件,以使各节点设备将subnet3的子网信息加入subnet0对应权限的子网管理列表;或者,可以在subnet1上发出管理权转移交易,使得各节点设备监听到所述管理权转移交易执行后生成的管理权转移事件后,在节点设备层面将subnet1对应权限的子网管理列表中subnet3的子网信息添加至subnet0对应权限的子网管理列表中,使得subnet0上子网管理合约的合约状态虽然尚未记载subnet3的子网信息,但对于节点设备而言却可以通过监听subnet0发生的事件时从subnet0对应权限的子网管理列表中读取到subnet3的子网信息,因此实质上使subnet0获取了获得对subnet3的管理权。

[0121] 如前所述,通过在现有区块链网络的基础上发出组建区块链子网的交易从而组建区块链子网,可以使得新组建的区块链网络受到组建该区块链网络的现有区块链的管理,然而,除了上述发组建区块链子网交易的方式(后续简称为交易组网方式),还可以通过其他手段组建区块链子网,并使得其受到现有区块链网络的管理。例如,可以通过注册方式在现有区块链网络上组建区块链子网(后续简称注册组网方式),待组建区块链子网的子网信息被直接注册至现有区块链网络,使得现有区块链子网获取待组建区块链子网的相关信

息,例如待组建区块链子网的子网标识和运行状态,其中各节点成员的公钥和插件配置信息、各节点设备的IP地址和端口信息等,这些信息会被写入现有区块链网络对应的系统合约的合约状态中,由此现有区块链网络将获取该待组建区块链子网的管理权,在完成注册后,便意味着区块链子网组建完成。由于注册方式并不需要通过交易在现有区块链网络的区块链节点之间传递信息,因此通过注册组网方式组建的区块链子网中的子网节点可以与部署在现有区块链网络中各节点的节点设备完全不同或部分不同,例如图4中subnet0以注册方式组建了一个subnet4,假设subnet0自身所包含的主网节点nodeA~nodeE分别部署于节点设备1~5,那么subnet4对应的子网节点可以部署于除节点设备1~5外的其他任意节点设备上,或者,subnet4中的子网节点nodeA4部署于节点设备1,而subnet4中的子网节点nodeF4部署于节点设备6,当然,subnet4中的子网节点也可以均部署于节点设备1~5之中。

[0122] 在一实施例中,在第一节点设备上部署有第一区块链网络中的区块链节点的情况下,第一节点设备若确定第二节点设备部署有第一区块链网络所管理的第二区块链网络中的区块链节点、且未部署第一区块链网络中的区块链节点,则向第二节点设备提供所述启动控制信息,以指示第二节点设备响应于所述启动控制信息而启动第二区块链网络中的区块链节点。上述第一区块链网络为第一节点设备本地部署的任一区块链节点所属的区块链网络,如前所述,第一节点设备可以通过监听第一区块链网络上子网管理合约执行子网信息查询交易后生成的子网信息查询事件,获取其中data字段包含的第一区块链网络管理下的所有区块链子网(而非本地部署的子网节点所属的区块链子网)的子网信息,因此第一节点设备通过该data字段中的内容从而确定任一区块链子网中的各子网节点所在节点设备的IP地址,通过将这些IP地址与第一区块链网络中的各区块链节点所在节点设备的IP地址进行对比(第一区块链网络中的各区块链节点的IP地址记录在管理第一区块链网络的第一区块链父网对应区块链节点所维护的子网管理列表中),从而确定特定区块链子网(第二区块链网络)中的特定子网节点所在的节点设备不属于第一区块链网络中的各区块链节点所在的节点设备,即确定该特定节点设备(第二节点设备)部署有第一区块链网络所管理的第二区块链网络中的区块链节点、且未部署第一区块链网络中的区块链节点,并且,该特定区块链子网是通过注册组网方式组建或者该特定区块链子网是通过交易组网方式组建但其中包含以注册方式加入的部署于第二节点设备的子网节点,在这种情况下,第一节点设备可以进一步通过上述data字段中记录的第二节点设备的IP地址,向第二节点设备提供所述启动控制信息,以指示所述第二节点设备启动第二节点设备本地部署的所述区块链子网中的子网节点,从而同步启动以注册方式组网的区块链子网中的子网节点。

[0123] 如前所述,第一节点设备可以通过控制任一区块链节点对应的插件管理器,以启动或关闭用于构成所述任一区块链节点的插件模块。在一种情况下,第一节点设备只会部署一个共享插件管理器,用于管理该节点设备下部署的所有区块链节点对应的插件模块,具体而言,共享插件管理器中维护有本地部署的各区块链节点的插件配置信息,而由于同一区块链网络中的不同区块链节点的插件模块往往相同,所以区块链节点的插件配置信息与该区块链节点所属的区块链网络的插件配置信息一般是一致的,当然同一区块链网络中的不同区块链节点的插件模块也可能不同。区块链节点的插件配置信息用于指示构成所述某一区块链节点的具体的插件模块,例如可以包括:业务网络插件、服务插件、P2P(peer-to-peer,点对点通讯)插件、区块链子网管理插件、Cache(高速缓存存储器)插件、验证插

件、事件管理插件、共识插件、同步插件、执行插件、区块链插件、存储插件等,本说明书对此并不做任何限制。

[0124] 在另一种情况下,第一节点设备会根据该节点设备部署的每个区块链节点均单独分配一个对应的独立插件管理器,使得独立插件管理器仅维护有对应区块链节点的插件配置信息,专门负责管理对应区块链节点的插件模块,例如开启、关闭、替换相应区块链节点的插件模块。

[0125] 如前所述,子网状态查询事件中可以包含第一节点设备本地部署的任一区块链网络(包括区块链主网以及各区块链子网)的插件配置信息,而任一区块链网络的插件配置信息中包含有该区块链网络下的各区块链节点的插件配置信息,这使得第一节点设备可以根据自身身份信息确定本地部署的子网节点的插件配置信息,所以第一节点设备维护的各子网管理列表中可以记录有本地部署的各区块链节点所属区块链子网的插件配置信息,因此,第一节点设备中部署的插件管理器能够通过读取各子网管理列表获取第一节点设备本地部署的各区块链节点的插件配置信息。

[0126] 第一节点设备在响应于启动/关闭控制信息而启动本地部署的任一区块链节点时,会控制任一区块链节点对应的插件管理器启动或关闭用于构成任一区块链节点的插件模块。例如,当节点设备1响应于区块链节点服务的启动控制信息而启动nodeA时,在一种情况下,节点设备1可以控制subnet0对应的主网节点即nodeA的独立插件管理器,以使nodeA的独立插件管理器根据nodeA的插件配置信息,启动用于构成nodeA的插件模块;在另一种情况下,节点设备1可以控制共享插件管理器,以使共享插件管理器查找得到subnet0对应的主网节点即nodeA的插件配置信息,并根据nodeA的插件配置信息启动用于构成nodeA0的插件模块;类似的,当节点设备1响应于区块链节点服务的关闭控制信息而关闭nodeA时,与上述启动过程类似,仅是将上述过程中的启动换成关闭;或者,当节点设备1响应于区块链节点服务的启动控制信息而启动子网节点nodeA1时,也与上述启动过程类似,仅是将nodeA的插件管理器替换成nodeA1的插件管理器,nodeA的插件配置信息替换成nodeA1的插件配置信息。

[0127] 第一节点设备上不同的区块链节点可能在运行时依赖于同一个插件模块,而这样的插件就处于共享状态。例如节点设备1上目前处于开启状态的区块链节点包括nodeA和nodeA1,用于构成nodeA的插件模块包括插件1.0、插件2.0、插件3.0和插件4.0,用于构成nodeA1的插件模块包括插件1.1、插件2.0、插件3.1、插件4.1和插件5.1,其中,插件1.0和插件1.1、插件3.0和插件3.1、插件4.0和插件4.1虽然属于同一类型的插件模块,具有相同的插件功能,但依然是两个不同的插件模块被分别嵌入不同的区块链节点,彼此不会共享信息,本质上属于不同的插件模块。而插件2.0则是同时被nodeA和nodeA1所依赖和使用,因此,像插件2.0这种同时被两个或两个以上的区块链节点共同使用的插件模块就处于共享状态,与之对应的,仅被一个区块链节点单独依赖的插件模块则处于独享状态,例如上述除插件2.0以外的插件均处于独享状态。

[0128] 另一方面,第一节点设备上可能运行有运行模式为共享模式的特定插件模块的,这意味着,假如第一节点设备上待启动的新区块链节点的构成里包含该特定插件模块,且新区块链节点对应的插件配置信息中对应该特定插件模块的运行需求为共享运行的情况下,那么在启动新区块链节点的过程中,将不会启动该特定插件模块,而是直接将已经运行

的该特定插件模块共享至新区块链节点,以供新区块链节点直接调用。而假设上述特定插件模块的运行模式为独享模式,那么在启动新区块链节点的过程中,由于该特定插件模块的运行模式为独享模式,不支持共享运行,因此无法直接供新区块链节点直接调用,而需要重新启动一个具有相同的插件模块以分配至新区块链节点,以供新区块链节点调用。

[0129] 第一节点设备上部署的插件管理器,会根据本地部署的各区块链节点的插件配置信息以及运行状态,维护有所管理的各区块链节点对应的插件信息属性列表,用于标注所需使用的各插件模块的属性信息,包括运行状态是开启状态还是关闭状态,运行模式是共享模式还是独享模式,处于共享状态还是独享状态,与其他插件模块的依赖关系等。插件信息属性列表是实时更新的,例如,当插件管理器读取子网管理列表后,发现当前运行状态为开启状态的区块链子网只有subnet0和subnet1,那么就会进一步查找得到本地部署的subnet0和subnet1分别对应的nodeA与nodeA1的插件配置信息,进而比对可以得到nodeA与nodeA1所共同使用的插件模块为插件2.0,于是将插件信息属性列表中的插件2.0标注为共享状态,而当subnet1关闭后,插件管理器再次读取nodeA维护的子网管理列表发现运行状态为开启状态的区块链子网只有subnet0,那么就只能查找得到nodeA的插件配置信息,由于插件2.0目前已经仅由nodeA所依赖,于是将插件信息属性列表中的插件2.0从共享状态更改为独享状态,同时将用于构成nodeA1除插件2.0以外的插件模块的运行状态从开启状态更改为关闭状态。除了通过子网管理列表中各区块链网络的子网信息获取本地部署的各区块链节点的插件配置信息、运行状态以外,插件管理器也可以单独维护本地部署的主网节点和各子网节点的插件配置信息与运行状态,而在完成对相应子网节点的运行状态的控制后,自行更新维护的运行状态等信息,从而可以更加及时、准确地更新插件信息属性列表中的各插件模块的属性信息。

[0130] 由于存在处于共享状态的插件以及运行模式为共享模式的插件,所以第一节点设备在启动用于构成任一区块链节点的插件模块时,需要避免重复启动运行模式为共享模式的插件,从而保证任一区块链节点中各插件模块的共享需求被正确满足,进而确保任一区块链节点的正常运行;而在关闭用于构成任一区块链节点的插件模块时,需要避免关闭处于共享状态的插件模块,从而防止影响第一子网节点以外的其他区块链节点的正常运行。即第一节点设备在启动用于构成所述任一区块链节点的插件模块时,需要按照以下流程进行:

[0131] 确定用于构成所述任一区块链节点各插件模块的插件描述信息,以及各插件模块的运行需求。在本实施例中,可以通过查询任一区块链节点对应的插件信息属性列表,查找得到用于构成所述任一区块链节点各插件模块的插件描述信息以及运行需求,其中,插件描述信息包括插件类型、插件名、插件功能中至少一项,用于指向具体的插件模块。

[0132] 在任一插件模块的运行需求为共享运行的情况下,若存在匹配于所述任一插件模块的插件描述信息的已开启插件模块,且所述已开启插件模块的运行模式为共享模式,则将所述已开启插件模块共享至所述任一区块链节点。其中,所述各插件模块的运行需求被定义于所述任一区块链节点对应的插件配置信息中,若插件配置信息中任一插件模块的运行需求为共享运行时,说明需要使该任一插件模块以共享模式运行,并且,如果在启动该任一插件模块之前,第一节点设备就已经通过插件信息属性列表查找到该任一插件模块的插件描述信息对应的已开启插件模块,那么会进一步在确定该已开启插件模块的运行模式为

共享模式的情况下将该已开启插件模块共享至该任一插件模块所属的区块链节点。

[0133] 在所述任一插件模块的运行需求为共享运行的情况下,若不存在匹配于所述插件描述信息且运行模式为共享模式的已开启插件模块,则启动所述任一插件模块。若插件配置信息中任一插件模块的运行需求为共享运行时,说明需要使该任一插件模块以共享模式运行,并且,如果在启动该任一插件模块之前,第一节点设备就已经通过插件信息属性列表查找到不存在该任一插件模块的插件描述信息对应的已开启插件模块,或者虽然存在该已开启插件模块但其运行模式不为共享模式,那么此时认为不存在匹配于所述插件描述信息且运行模式为共享模式的已开启插件模块,因此需要启动该任一插件模块,同时,在启动所述插件描述信息对应的插件模块时,还会基于所述运行需求设置所启动的插件模块的运行模式,例如由于该任一插件模块的共享需求为共享运行,因此可以将启动后的任一插件模块的运行模式设置为共享模式,以使后续需要使用该任一插件模块的区块链节点在启动时可以将该任一插件模块共享至后续启动的区块链节点。

[0134] 在所述任一插件模块的运行需求为独立运行的情况下,启动所述任一插件模块。若插件配置信息中任一插件模块的运行需求为独立运行时,说明需要使该任一插件模块以独享模式运行,因此可以直接启动该任一插件模块并将其运行模式设置为独享模式,以使后续需要使用该任一插件模块的区块链节点在启动时无法将该任一插件模块共享至后续启动的区块链节点,从而满足其独享模式的运行模式,确保插件配置信息中该任一插件模块独立运行的运行需求。

[0135] 在上述情况下,各插件模块的运行需求被定义于所述任一区块链节点对应的插件配置信息中,而不同区块链节点对应的各个插件配置信息可以有所差异,因此对于不同的区块链节点可以分配不同的插件共享策略,这种对于不同的区块链节点的差异化管理能够灵活地满足用户的各类实际需求。在另一种情况下,各插件模块的运行需求为全局预定义信息,因此任一区块链节点对应的插件配置信息中可以无需记录各插件模块的运行需求,插件配置器根据不同的插件描述信息维护有各个插件模块的运行需求,那么在通过上述方式启动插件模块时,使得预设的插件模块总以共享模式运行,当需要启动用于构成任一区块链节点的插件模块时,如果该插件模块的插件描述信息表明该插件模块具有共享需求,则可以直接将对应该插件描述信息的已开启插件模块共享至该任一区块链节点,从而提供统一的共享策略。

[0136] 具体而言,当节点设备1启动第一主网节点nodeA时,节点设备1可以通过插件管理器查看nodeA的插件配置信息,获取用于构成nodeA的插件模块的插件描述信息包括插件1、插件2、插件3和插件4,此时由于没有其他区块链节点,因此直接将上述插件描述信息对应的插件模块启动得到的插件1.0、插件2.0、插件3.0和插件4.0分配至nodeA从而启动nodeA,而在进一步通过插件配置信息确定插件1、插件3和插件4的运行需求为独立运行且插件2的运行需求为共享运行后,将插件1.0、插件3.0和插件4.0的运行模式设置为独享模式,以及将插件2.0的运行模式设置为共享模式。在nodeA启动完成后,假设节点设备1确定本地部署有nodeA1和nodeA2,且nodeA1所属的subnet1的运行状态为开启状态,而nodeA2所属的subnet2的运行状态为关闭状态,那么此时节点设备就会通过插件管理器查看nodeA1的插件配置信息,获取用于构成nodeA1的插件模块的插件描述信息包括插件1、插件2、插件3、插件4和插件5,此时节点设备1已经运行有相同插件描述信息的插件1.0、插件2.0、插件3.0和

插件4.0,则需要进一步查看各插件模块的运行需求,通过nodeA1的插件配置信息可以得知插件1、插件3和插件4的共享需求均为独立运行,而插件2和插件5的共享需求为共享运行,因此根据共享需求直接启动插件1.1、插件3.1和插件4.1并分配至nodeA1,同时,由于插件2描述信息对应的已启动的插件2.0的运行模式为共享模式,因此无需重新启动插件2描述信息对应的新插件,而是将nodeA所使用的插件2.0直接共享至nodeA1,而插件5的共享需求虽然为共享运行,但节点设备1目前并不存在插件5描述信息对应的已开启插件,因此需要启动插件5.1并将其运行模式设定为共享模式,与插件2.0以及其他的运行模式被设置为独立模式的插件1.1、插件3.1、插件4.1和插件5.1共同构成nodeA1。在nodeA1启动完成后,假设节点设备1确定本地部署有nodeA3,且nodeA3所属的subnet3的运行状态为开启状态,那么此时节点设备1就会通过插件管理器查看nodeA3的插件配置信息,获取用于构成nodeA3的插件模块的插件描述信息包括插件1、插件2和插件5,此时节点设备1已经运行有相同插件描述信息的插件1.0、插件1.1、插件2.0和插件5.1,则需要进一步查看各插件模块的运行需求,通过nodeA3的插件配置信息可以得知插件1的共享需求均为独立运行,而插件2和插件5的共享需求为共享运行,因此根据共享需求直接启动插件1.2并分配至nodeA3,同时,由于插件2描述信息对应的已启动的插件2.0的运行模式为共享模式,插件5描述信息对应的已启动的插件5.1,因此无需重新启动插件2或插件5描述信息对应的新插件,而是将nodeA和nodeA1所共同使用的插件2.0直接共享至nodeA3,将nodeA1所使用的插件5.1共享至nodeA3,与运行模式被设置为独立模式的插件1.2共同构成nodeA1。

[0137] 插件管理器中维护有各插件模块之间的依赖关系。假如插件A的正常运行必须要求插件B已经运行,那么可以认为插件A与插件B之间具有依赖关系,并且插件A依赖于插件B,可以表示为“插件A→插件B”。插件管理器中涉及各插件模块之间的依赖关系可以通过系统预制或用于预先定义,并记录在插件信息属性列表中,如表1所示,根据各插件之间的依赖关系,可以整理得到两条依赖关系:“插件A→插件B→插件D”、“插件C→插件D和插件E”。

插件名	依赖插件	其他
A	B	...
B	D	...
C	D、E	...
D	无	...
...

[0138]

表1

[0139] 当第一节点设备根据所述控制信息启动或关闭用于构成任一区块链节点的插件模块时,可以按照构成任一区块链节点的各个插件模块之间的启停顺序,启动或关闭第一子网节点的插件模块;其中,所述启停顺序与各个插件模块之间的依赖关系相关。具体来说,假设构成任一区块链节点的插件模块为插件A、插件B和插件D,其依赖关系为“插件A→插件B→插件D”,那么在关闭用于构成第一子网节点的插件模块的情况下,关闭顺序与依赖顺序相同,也是“插件A→插件B→插件D”,这样通过先关闭有依赖的上层插件模块,后关闭无依赖的底层插件模块;而在启动用于构成第一子网节点的插件模块的情况下,启动顺序将与依赖顺序相反,也即“插件D→插件B→插件A”,这样通过先开启自身无依赖的上层插件

模块,后开启有依赖的上层插件模块。在上述方案中,通过按照各个插件模块之间的依赖关系决定启停顺序,可以使在启停过程中不会出现插件模块因缺少依赖插件而运行出错的情况,避免导致数据丢失甚至节点崩溃,同时充分地使用和释放内存资源。

[0140] 例如,当节点设备1获取区块链节点服务的关闭控制信息,节点设备1可以通过插件管理器,获取subnet1对应的子网节点nodeA1由插件1.1、插件2.0、插件3.1、插件4.1和插件5.1构成,以及相应的依赖关系为“插件1.1→插件3.1→插件2.0”、“插件4.1→插件5.1”,由于需要关闭nodeA1,因此根据前述规则确定得到“插件1.1→插件3.1→插件2.0”、“插件4.1→插件5.1”的关闭顺序,其中,按照“插件1.1→插件3.1→插件2.0”关闭插件1.1、插件3.1和插件2.0,与按照“插件4.1→插件5.1”关闭插件4.1和5.1之间相互独立,节点设备1按照上述关闭顺序关闭相应的插件模块,从而起到无损关闭的效果,当然,当插件2.0为共享插件时,可以将“插件1.1→插件3.1→插件2.0”的关闭顺序替换为“插件1.1→插件3.1”,即不关闭作为共享插件的插件2.0,从而不影响其他子网节点的正常运行。

[0141] 图6是一示例性实施例提供的一种关闭区块链节点服务的方法的流程图。如图6所示,该方法应用于以区块链主网为根结点、各个区块链子网分别为其他结点的树形系统,所述树形系统中的任一区块链子网由其父结点对应的区块链网络所管理,所述方法包括:

[0142] 步骤602,第一节点设备获取区块链节点服务的关闭控制信息;

[0143] 步骤604,第一节点设备根据本地部署的各区块链节点分别维护的子网管理列表,确定各区块链节点所属区块链网络在所述树形系统中对应的结点分支;其中,任一区块链节点维护的子网管理列表用于记录所述任一区块链节点所属区块链网络管理的区块链子网;

[0144] 步骤606,第一节点设备响应于所述关闭控制信息,按照从叶子结点向根结点的方向,依次关闭所述结点分支中各结点在第一节点设备上对应的区块链节点。

[0145] 可选的,所述关闭控制信息中包含第一节点设备对应的身份信息。

[0146] 可选的,所述第一节点设备关闭任一区块链网络中的区块链节点,包括:

[0147] 第一节点设备在所述任一区块链网络的运行状态为开启状态的情况下,关闭所述任一区块链网络中的区块链节点。

[0148] 可选的,还包括:

[0149] 第一节点设备通过查询所述任一区块链网络的父结点对应的区块链网络所维护的子网管理列表,确定所述任一区块链网络的运行状态。

[0150] 可选的,还包括:

[0151] 在第一节点设备上部署有第一区块链网络中的区块链节点的情况下,第一节点设备若确定第二节点设备部署有第一区块链网络所管理的第二区块链网络中的区块链节点、且未部署第一区块链网络中的区块链节点,则向第二节点设备提供所述关闭控制信息,以指示第二节点设备响应于所述关闭控制信息而关闭第二区块链网络中的区块链节点。

[0152] 可选的,所述第一节点设备关闭任一区块链节点,包括:

[0153] 第一节点设备关闭用于构成所述任一区块链节点的插件模块。

[0154] 可选的,所述关闭用于构成所述任一区块链节点的插件模块,包括:通过所述任一区块链节点对应的插件管理器关闭用于构成所述任一区块链节点的插件模块。

[0155] 可选的,

[0156] 所述插件管理器为第一节点设备上部署的所有区块链节点对应的全局插件管理器;或者,

[0157] 所述插件管理器为单独应用于所述任一区块链节点的独立插件管理器。

[0158] 可选的,所述关闭用于构成所述任一区块链节点的插件模块,包括:

[0159] 确定用于构成所述任一区块链节点各插件模块的插件描述信息;

[0160] 若存在匹配于所述任一插件模块的插件描述信息的已开启插件模块的运行模式为独立模式、或运行模式为共享模式但未被共享至所述任一区块链节点以外的其他区块链节点,则关闭所述已开启插件模块。如前所述,如果某一插件模块的运行模式为共享模式但未被共享至所述任一区块链节点以外的其他区块链节点,则说明该插件模块虽然运行模式为共享模式但并未处于共享状态,因此在关闭该插件时不用考虑对其他区块链节点的干扰从而直接关闭该插件模块,因为此时该插件模块并未被其他区块链节点所依赖,因此可以将其看做是一个独立模式运行的插件模块。

[0161] 可选的,所述任一插件模块的运行模式由所述任一插件模块的运行需求所确定。

[0162] 可选的,所述任一插件模块的运行需求被定义于所述任一区块链节点对应的插件配置信息中。

[0163] 可选的,所述任一插件模块的运行需求为全局预定义信息。

[0164] 可选的,所述关闭用于构成所述任一区块链节点的插件模块,包括:根据用于构成所述任一区块链节点各插件模块之间的依赖关系,依次关闭用于构成所述任一区块链节点各插件模块。

[0165] 与前述方法的实施例相对应,本说明书还提供了装置、电子设备以及存储介质的实施例。

[0166] 图7是一示例性实施例提供的一种设备的示意结构图。请参考图7,在硬件层面,该设备包括处理器702、内部总线704、网络接口706、内存708以及非易失性存储器710,当然还可能包括其他业务所需要的硬件。本说明书一个或多个实施例可以基于软件方式来实现,比如由处理器702从非易失性存储器710中读取对应的计算机程序到内存708中然后运行。当然,除了软件实现方式之外,本说明书一个或多个实施例并不排除其他实现方式,比如逻辑器件抑或软硬件结合的方式等等,也就是说以下处理流程的执行主体并不限于各个逻辑单元,也可以是硬件或逻辑器件。

[0167] 请参考图8,图8是本说明书根据一示例性实施例示出的启动区块链节点服务的装置的框图,启动区块链节点服务的装置可以应用于如图7所示的设备中,以实现本说明书的技术方案,所述装置应用于以区块链主网为根结点、各个区块链子网分别为其他结点的树形系统,所述树形系统中的任一区块链子网由其父结点对应的区块链网络所管理,所述装置包括:

[0168] 控制信息获取模块801,用于使第一节点设备获取区块链节点服务的启动控制信息;

[0169] 结点分支确定模块802,用于使第一节点设备根据本地部署的各区块链节点分别维护的子网管理列表,确定各区块链节点所属区块链网络在所述树形系统中对应的结点分支;其中,任一区块链节点维护的子网管理列表用于记录所述任一区块链节点所属区块链网络管理的区块链子网;

[0170] 节点启动模块803,用于使第一节点设备响应于所述启动控制信息,按照从根结点向叶子结点的方向,依次启动所述结点分支中各结点在第一节点设备上对应的区块链节点。

[0171] 可选的,所述启动控制信息中包含第一节点设备对应的身份信息。

[0172] 可选的,所述节点启动模块803用于使第一节点设备启动任一区块链网络中的区块链节点,包括:

[0173] 第一节点设备在所述任一区块链网络的运行状态为开启状态的情况下,启动所述任一区块链网络中的区块链节点。

[0174] 可选的,还包括:

[0175] 运行状态确定模块804,用于使第一节点设备通过查询所述任一区块链网络的父结点对应的区块链网络所维护的子网管理列表,确定所述任一区块链网络的运行状态。

[0176] 可选的,还包括:

[0177] 控制信息提供模块805,用于使第一节点设备在第一节点设备上部署有第一区块链网络中的区块链节点的情况下,第一节点设备若确定第二节点设备部署有第一区块链网络所管理的第二区块链网络中的区块链节点、且未部署第一区块链网络中的区块链节点,则向第二节点设备提供所述启动控制信息,以指示第二节点设备响应于所述启动控制信息而启动第二区块链网络中的区块链节点。

[0178] 可选的,所述第一节点设备启动任一区块链节点,包括:

[0179] 第一节点设备启动用于构成所述任一区块链节点的插件模块。

[0180] 可选的,所述启动用于构成所述任一区块链节点的插件模块,包括:通过所述任一区块链节点对应的插件管理器启动用于构成所述任一区块链节点的插件模块。

[0181] 可选的,

[0182] 所述插件管理器为第一节点设备上部署的所有区块链节点对应的全局插件管理器;或者,

[0183] 所述插件管理器为单独应用于所述任一区块链节点的独立插件管理器。

[0184] 可选的,所述启动用于构成所述任一区块链节点的插件模块,包括:

[0185] 确定用于构成所述任一区块链节点各插件模块的插件描述信息,以及各插件模块的运行需求;

[0186] 在任一插件模块的运行需求为共享运行的情况下,若存在匹配于所述任一插件模块的插件描述信息的已开启插件模块,且所述已开启插件模块的运行模式为共享模式,则将所述已开启插件模块共享至所述任一区块链节点;

[0187] 在所述任一插件模块的运行需求为共享运行的情况下,若不存在匹配于所述插件描述信息且运行模式为共享模式的已开启插件模块,则启动所述任一插件模块;

[0188] 在所述任一插件模块的运行需求为独立运行的情况下,启动所述任一插件模块。

[0189] 可选的,所述各插件模块的运行需求被定义于所述任一区块链节点对应的插件配置信息中。

[0190] 可选的,所述各插件模块的运行需求为全局预定义信息。

[0191] 可选的,还包括:运行模式设置模块806,用于使第一节点设备在启动所述插件描述信息对应的插件模块时,基于所述运行需求设置所启动的插件模块的运行模式。

[0192] 可选的,所述启动用于构成所述任一区块链节点的插件模块,包括:根据用于构成所述任一区块链节点各插件模块之间的依赖关系,依次启动用于构成所述任一区块链节点各插件模块。

[0193] 请参考图9,图9是本说明书根据一示例性实施例示出的关闭区块链节点服务的装置的框图,关闭区块链节点服务的装置可以应用于如图7所示的设备中,以实现本说明书的技术方案,所述装置应用于以区块链主网为根结点、各个区块链子网分别为其他结点的树形系统,所述树形系统中的任一区块链子网由其父结点对应的区块链网络所管理,所述装置包括:

[0194] 控制信息获取模块901,用于使第一节点设备获取区块链节点服务的关闭控制信息;

[0195] 结点分支确定模块902,用于使第一节点设备根据本地部署的各区块链节点分别维护的子网管理列表,确定各区块链节点所属区块链网络在所述树形系统中对应的结点分支;其中,任一区块链节点维护的子网管理列表用于记录所述任一区块链节点所属区块链网络管理的区块链子网;

[0196] 节点关闭模块903,用于使第一节点设备响应于所述关闭控制信息,按照从叶子结点向根结点的方向,依次关闭所述结点分支中各结点在第一节点设备上对应的区块链节点。

[0197] 可选的,所述关闭控制信息中包含第一节点设备对应的身份信息。

[0198] 可选的,所述节点关闭模块903用于使第一节点设备关闭任一区块链网络中的区块链节点,包括:

[0199] 第一节点设备在所述任一区块链网络的运行状态为开启状态的情况下,关闭所述任一区块链网络中的区块链节点。

[0200] 可选的,还包括:

[0201] 运行状态确定模块904,用于使第一节点设备通过查询所述任一区块链网络的父结点对应的区块链网络所维护的子网管理列表,确定所述任一区块链网络的运行状态。

[0202] 可选的,还包括:

[0203] 控制信息提供模块905,用于使第一节点设备在第一节点设备上部署有第一区块链网络中的区块链节点的情况下,第一节点设备若确定第二节点设备部署有第一区块链网络所管理的第二区块链网络中的区块链节点、且未部署第一区块链网络中的区块链节点,则向第二节点设备提供所述关闭控制信息,以指示第二节点设备响应于所述关闭控制信息而关闭第二区块链网络中的区块链节点。

[0204] 可选的,所述第一节点设备关闭任一区块链节点,包括:

[0205] 第一节点设备关闭用于构成所述任一区块链节点的插件模块。

[0206] 可选的,所述关闭用于构成所述任一区块链节点的插件模块,包括:通过所述任一区块链节点对应的插件管理器关闭用于构成所述任一区块链节点的插件模块。

[0207] 可选的,

[0208] 所述插件管理器为第一节点设备上部署的所有区块链节点对应的全局插件管理器;或者,

[0209] 所述插件管理器为单独应用于所述任一区块链节点的独立插件管理器。

[0210] 可选的,所述关闭用于构成所述任一区块链节点的插件模块,包括:

[0211] 确定用于构成所述任一区块链节点各插件模块的插件描述信息;

[0212] 若存在匹配于所述任一插件模块的插件描述信息的已开启插件模块的运行模式为独立模式、或运行模式为共享模式但未被共享至所述任一区块链节点以外的其他区块链节点,则关闭所述已开启插件模块。

[0213] 可选的,所述任一插件模块的运行模式由所述任一插件模块的运行需求所确定。

[0214] 可选的,所述任一插件模块的运行需求被定义于所述任一区块链节点对应的插件配置信息中。

[0215] 可选的,所述任一插件模块的运行需求为全局预定义信息。

[0216] 可选的,所述关闭用于构成所述任一区块链节点的插件模块,包括:根据用于构成所述任一区块链节点各插件模块之间的依赖关系,依次关闭用于构成所述任一区块链节点各插件模块。

[0217] 相应的,本说明书还提供一种装置,所述装置包括有处理器;用于存储处理器可执行指令的存储器;其中,所述处理器被配置为实现上述全部方法实施例提供的启动/关闭区块链节点服务的方法的步骤。

[0218] 相应的,本说明书还提供一种计算机可读存储介质,其上存储有可执行的指令;其中,该指令被处理器执行时,实现上述全部方法实施例提供的启动/关闭区块链节点服务的方法的步骤。

[0219] 对于装置实施例而言,由于其基本对应于方法实施例,所以相关之处参见方法实施例的部分说明即可。以上所描述的装置实施例仅仅是示意性的,其中所述作为分离部件说明的模块可以是或者也可以不是物理上分开的,作为模块显示的部件可以是或者也可以不是物理模块,即可以位于一个地方,或者也可以分布到多个网络模块上。可以根据实际的需要选择其中的部分或者全部模块来实现本说明书方案的目的。本领域普通技术人员在不付出创造性劳动的情况下,即可以理解并实施。

[0220] 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。

[0221] 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,在实施本说明书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。

[0222] 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。

[0223] 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。

[0224] 本说明书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说明书,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。

[0225] 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。

[0226] 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。在一个典型的配置中,计算机包括一个或多个处理器(CPU)、输入/输出接口、网络接口和内存。

[0227] 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。

[0228] 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、磁盘存储、量子存储器、基于石墨烯的存储介质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。

[0229] 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。

[0230] 上述对本说明书特定实施例进行了描述。其它实施例在所附权利要求书的范围内。在一些情况下,在权利要求书中记载的动作或步骤可以按照不同于实施例中的顺序来执行并且仍然可以实现期望的结果。另外,在附图中描绘的过程不一定要求示出的特定顺序或者连续顺序才能实现期望的结果。在某些实施方式中,多任务处理和并行处理也是可以的或者可能是有利的。

[0231] 在本说明书一个或多个实施例使用的术语是仅仅出于描述特定实施例的目的,而非旨在限制本说明书一个或多个实施例。在本说明书一个或多个实施例和所附权利要求书中所使用的单数形式的“一种”、“所述”和“该”也旨在包括多数形式,除非上下文清楚地表示其他含义。还应当理解,本文中使用的术语“和/或”是指并包含一个或多个相关联的列出项目的任何或所有可能组合。

[0232] 应当理解,尽管在本说明书一个或多个实施例可能采用术语第一、第二、第三等来描述各种信息,但这些信息不应限于这些术语。这些术语仅用来将同一类型的信息彼此区分开。例如,在不脱离本说明书一个或多个实施例范围的情况下,第一信息也可以被称为第二信息,类似地,第二信息也可以被称为第一信息。取决于语境,如在此所使用的词语“如果”可以被解释成为“在……时”或“当……时”或“响应于确定”。

[0233] 以上所述仅为本说明书一个或多个实施例的较佳实施例而已,并不用以限制本说明书一个或多个实施例,凡在本说明书一个或多个实施例的精神和原则之内,所做的任何修改、等同替换、改进等,均应包含在本说明书一个或多个实施例保护的范围之内。

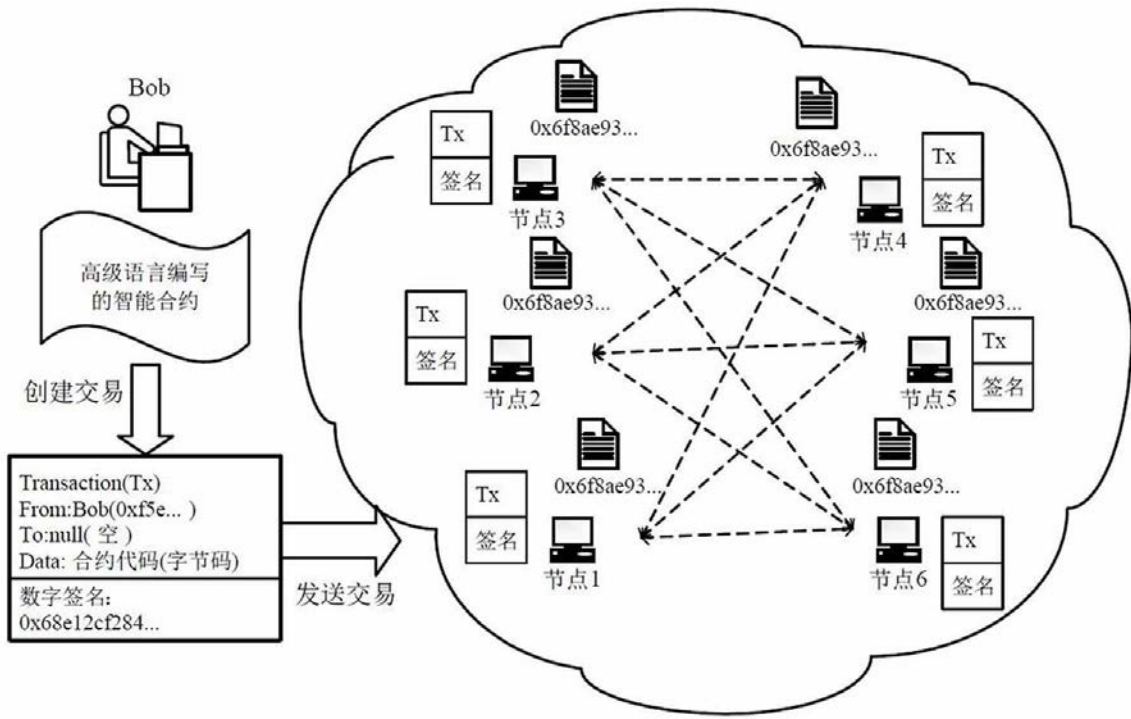


图1

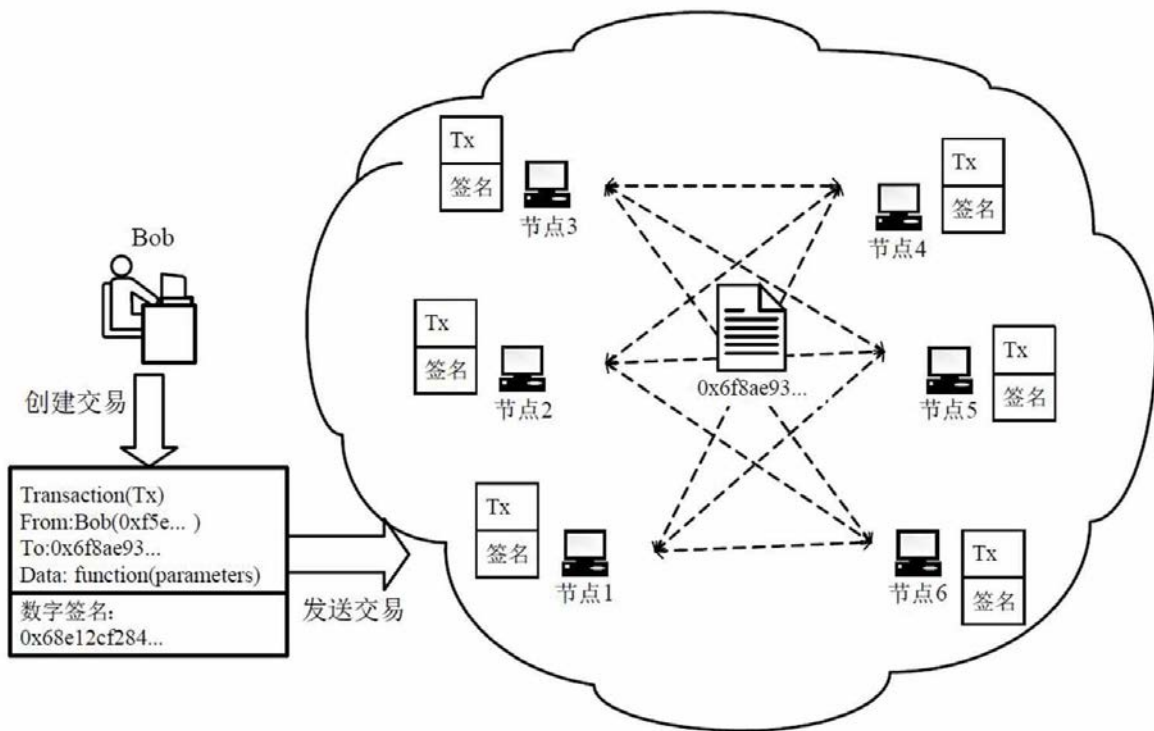


图2

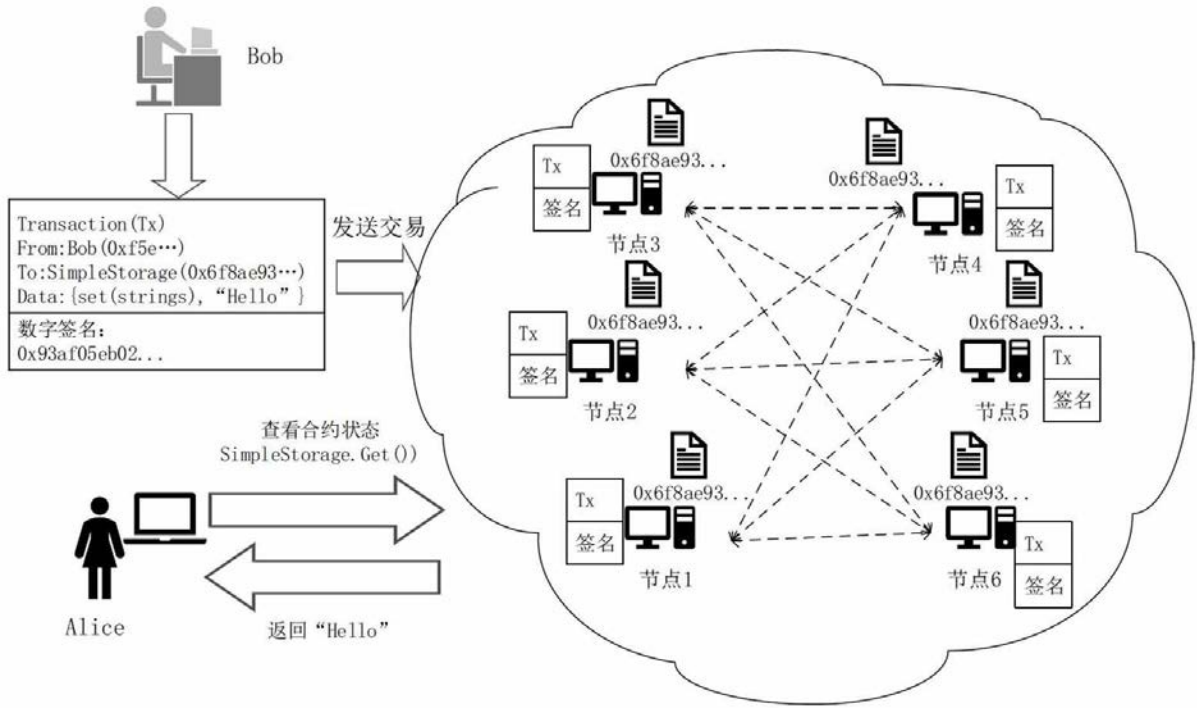


图3

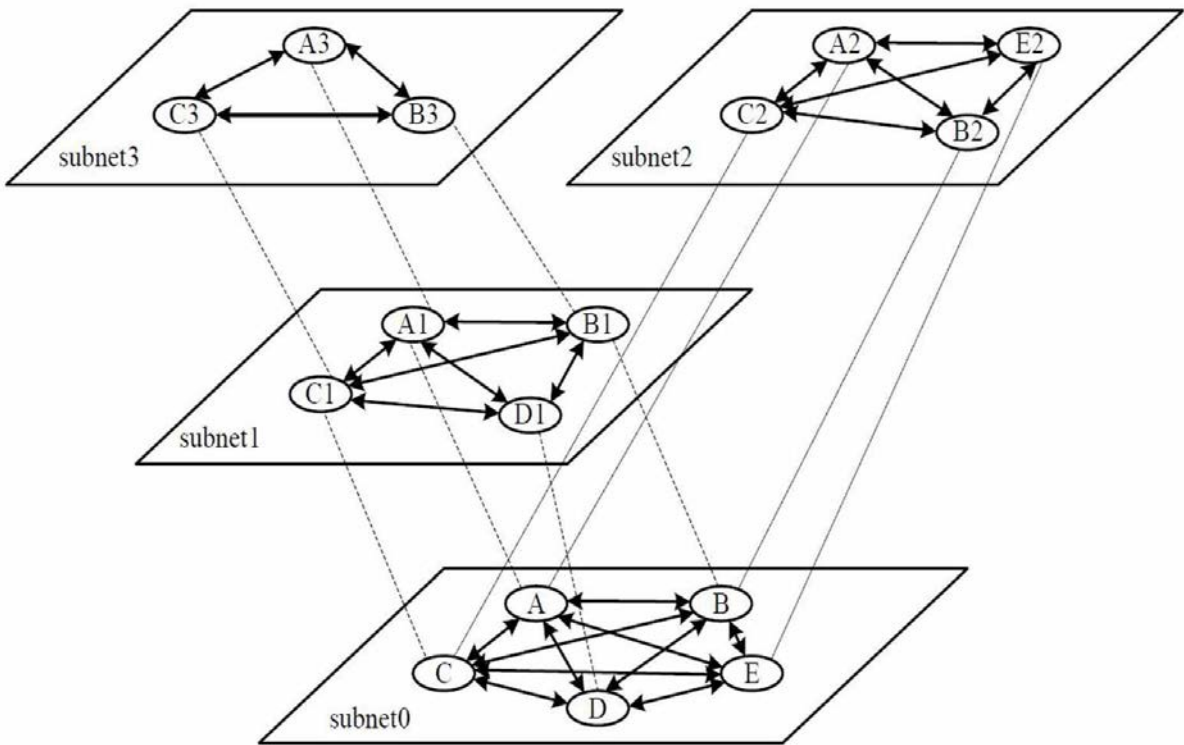


图4

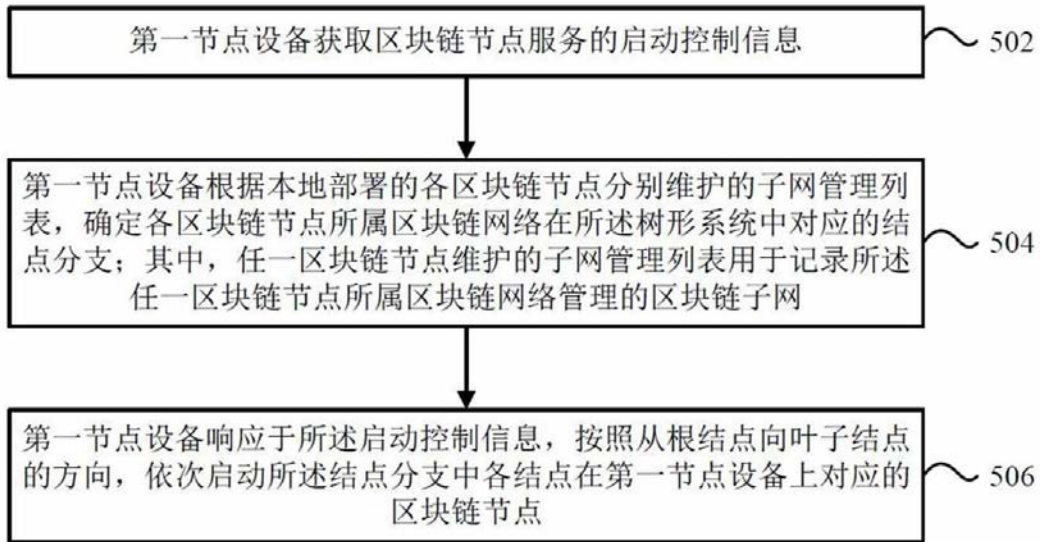


图5

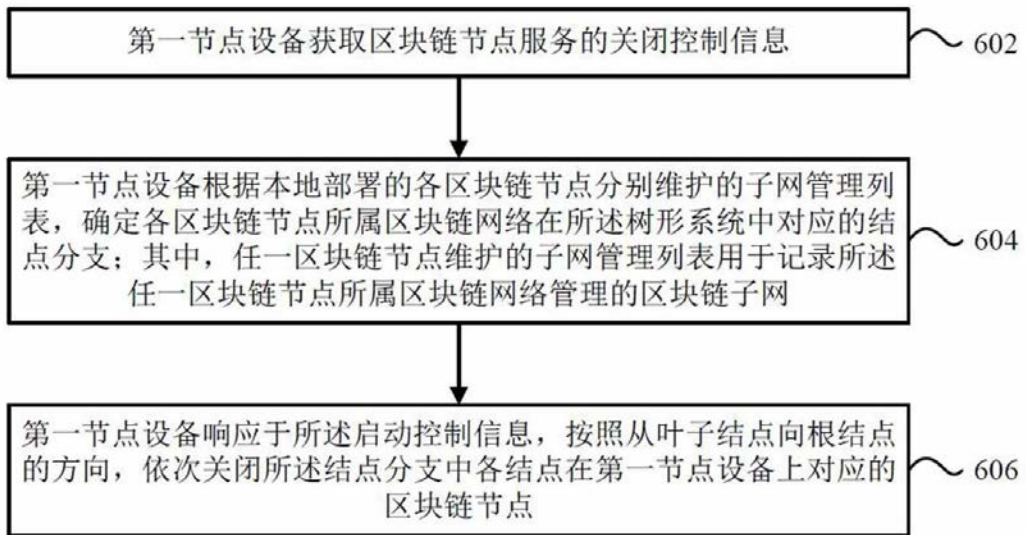


图6

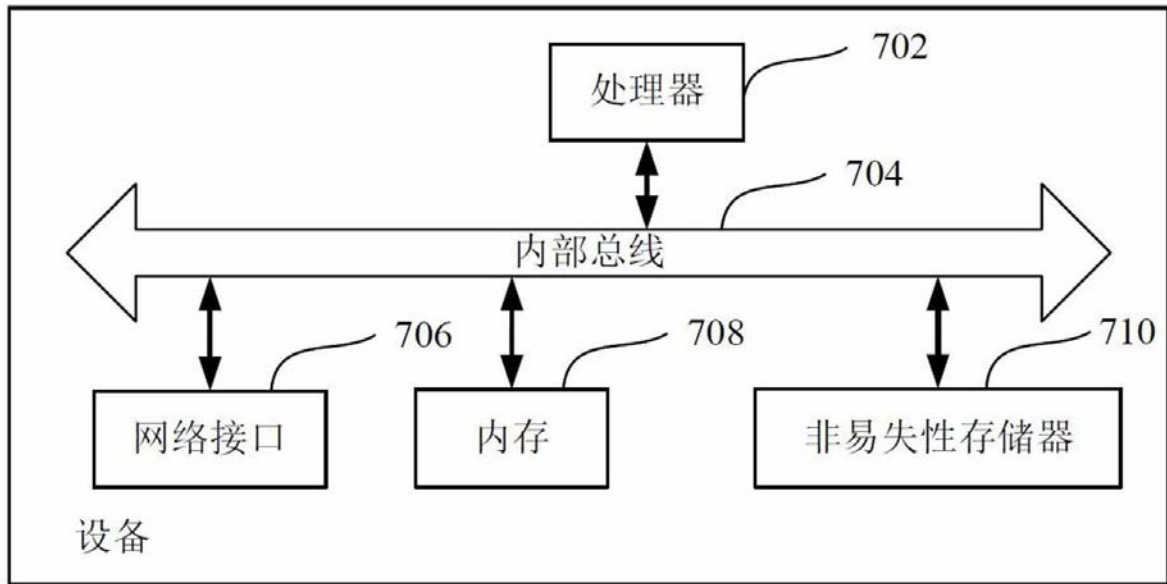


图7

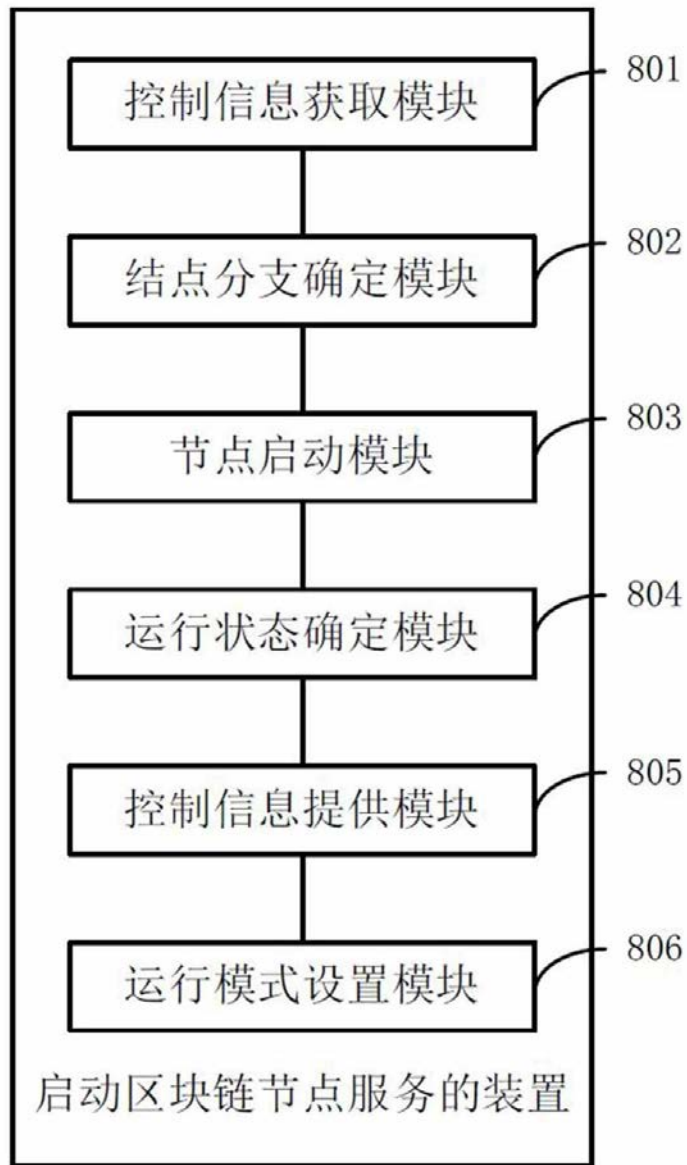


图8

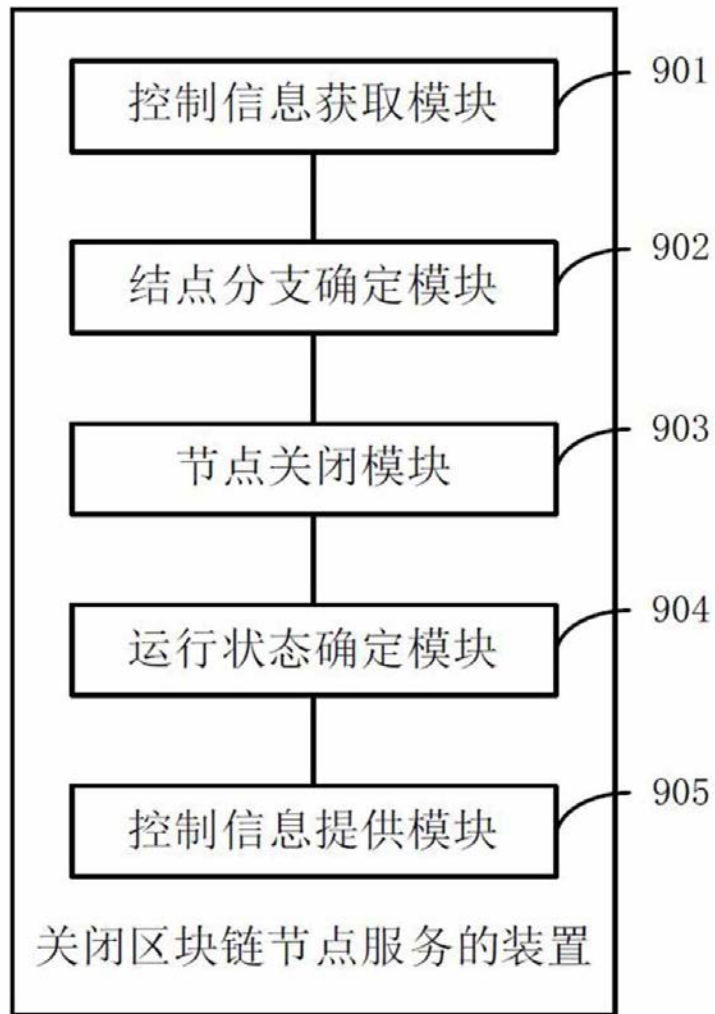


图9