



(12)发明专利

(10)授权公告号 CN 107368507 B

(45)授权公告日 2020.03.27

(21)申请号 201710190786.1

G06F 9/46(2006.01)

(22)申请日 2017.03.28

(56)对比文件

(65)同一申请的已公布的文献号  
申请公布号 CN 107368507 A

CN 106452785 A,2017.02.22,

CN 102724272 A,2012.10.10,

CN 106209877 A,2016.12.07,

(43)申请公布日 2017.11.21

CN 101523355 A,2009.09.02,

(73)专利权人 创新先进技术有限公司

CN 103220165 A,2013.07.24,

地址 开曼群岛大开曼岛西湾路802号木槿

CN 103744809 A,2014.04.23,

街大展览馆31119号邮箱邮编KY1-1205

CN 101394306 A,2009.03.25,

CN 105912618 A,2016.08.31,

(72)发明人 唐强

CN 106339639 A,2017.01.18,

US 2016125403 A1,2016.05.05,

(74)专利代理机构 北京晋德允升知识产权代理有限公司 11623

审查员 刘亭

代理人 王戈

(51)Int.Cl.

G06F 16/27(2019.01)

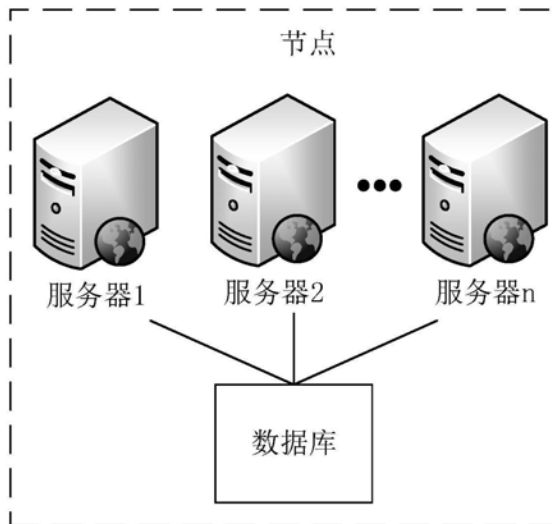
权利要求书2页 说明书9页 附图5页

(54)发明名称

一种基于区块链的共识方法及装置

(57)摘要

本申请实施例公开了一种基于区块链的共识方法及装置,一个区块链节点包括第一服务器、第二服务器和至少一个数据库,所述方法包括:数据库存储执行共识所需的共识数据,供第一服务器和第二服务器在执行共识的过程中进行调用,第一服务器在执行共识前或执行共识过程中出现异常,第二服务器代替第一服务器,从数据库中获取共识数据,并根据共识数据执行共识,生成共识结果,第二服务器将共识结果存储在数据库中。利用本申请实施例,节点中正常的服务器能够接替异常的服务器,从数据库中获取共识数据执行共识,从而保证了共识的正常进行,能够在一定程度上提升共识过程的成功率,进而提升区块链的业务处理效率。



1. 一种基于区块链的共识方法,其特征在于,一个区块链节点包括第一服务器、第二服务器和至少一个数据库,所述方法包括:

所述数据库存储执行共识所需的共识数据,供所述第一服务器和所述第二服务器在执行共识的过程中进行调用;

所述第一服务器在执行共识前或执行共识过程中出现异常时,所述第二服务器代替所述第一服务器,从所述数据库中获取共识数据,并根据所述共识数据执行共识,生成共识结果,具体包括:

当所述第一服务器在执行共识过程中出现异常时,所述第二服务器代替所述第一服务器,接收参与共识过程的其他设备重试发送共识消息,根据所述共识消息,从所述数据库中获取与所述共识消息对应的共识数据;

其中,当所述其他设备在发送共识消息并经过设定时间未接收到响应时,则重试发送所述共识消息;所述第二服务器将所述共识结果存储在所述数据库中。

2. 如权利要求1所述的方法,其特征在于,所述第一服务器在执行共识前出现异常时,所述第二服务器代替所述第一服务器,从所述数据库中获取共识数据,具体包括:

当所述第一服务器在执行共识前出现异常时,所述第二服务器代替所述第一服务器,接收参与共识过程的其他设备发送的共识消息,根据所述共识消息,从所述数据库中获取与所述共识消息对应的共识数据。

3. 如权利要求2所述的方法,其特征在于,所述共识消息中包含业务请求的标识;

根据所述共识消息,从所述数据库中获取与所述共识消息对应的共识数据,具体包括:

所述第二服务器根据所述共识消息中包含的业务请求的标识,从所述数据库中查找并获取与所述标识对应的共识数据。

4. 如权利要求2所述的方法,其特征在于,所述区块链节点还包括网关;

接收参与共识过程的其他设备发送的共识消息,具体包括:

当所述网关确定出所述第一服务器在执行共识前出现异常时,所述网关将接收到的、参与共识过程的其他设备发送的共识消息,转发给正常运行的第二服务器;

所述第二服务器接收所述网关转发的、参与共识过程的其他设备发送的共识消息;

接收参与共识过程的其他设备重试发送共识消息,具体包括:

当所述网关确定出所述第一服务器在执行共识过程中出现异常时,所述网关将接收到的、参与共识过程的其他设备重试发送共识消息,转发给正常运行的第二服务器;

所述第二服务器接收所述网关转发的、参与共识过程的其他设备重试发送共识消息。

5. 如权利要求4所述的方法,其特征在于,所述网关通过以下方式确定所述第一服务器及第二服务器的运行状态:

所述网关接收所述第一服务器及第二服务器按照预设周期向所述网关发送运行状态消息;

所述网关根据所述运行状态消息,确定所述第一服务器及第二服务器的运行状态。

6. 一种基于区块链的共识装置,其特征在于,一个区块链节点包括第一服务器、第二服务器和至少一个数据库,其中,

所述数据库存储执行共识所需的共识数据,供所述第一服务器和所述第二服务器在执行共识的过程中进行调用;

所述第一服务器在执行共识前或执行共识时出现异常；

所述第二服务器包括：

获取模块，从所述数据库中获取共识数据；

共识执行模块，根据所述共识数据执行共识，生成共识结果；

存储模块，将所述共识结果存储在所述数据库中；

接收模块，当所述第一服务器在执行共识过程中出现异常时，所述接收模块，接收参与共识过程的其他设备重试发送共识消息；所述获取模块，根据所述共识消息，从所述数据库中获取与所述共识消息对应的共识数据；

其中，当所述其他设备在发送共识消息并经过设定时间未接收到响应时，则重试发送所述共识消息。

7. 如权利要求6所述的装置，其特征在于，所述装置还包括：接收模块；

当所述第一服务器在执行共识前出现异常时，所述接收模块，接收参与共识过程的其他设备发送的共识消息；所述获取模块，根据所述共识消息，从所述数据库中获取与所述共识消息对应的共识数据。

8. 如权利要求7所述的装置，其特征在于，所述共识消息中包含业务请求的标识；

所述获取模块，根据所述共识消息中包含的业务请求的标识，从所述数据库中查找并获取与所述标识对应的共识数据。

9. 如权利要求8所述的装置，其特征在于，所述区块链节点还包括网关；

所述装置还包括：运行状态确定模块及转发模块；

当所述运行状态确定模块确定出所述第一服务器在执行共识前出现异常时，所述转发模块，将接收到的、参与共识过程的其他设备发送的共识消息，转发给正常运行的第二服务器；

当所述运行状态确定模块确定出所述第一服务器在执行共识过程中出现异常时，所述转发模块，将接收到的、参与共识过程的其他设备重试发送共识消息，转发给正常运行的第二服务器。

10. 如权利要求9所述的装置，其特征在于，所述装置还包括：网关接收模块，接收所述第一服务器及第二服务器按照预设周期向所述网关发送运行状态消息；

所述运行状态确定模块，根据所述运行状态消息，确定所述第一服务器及第二服务器的运行状态。

## 一种基于区块链的共识方法及装置

### 技术领域

[0001] 本申请涉及计算机技术领域,尤其涉及一种基于区块链的共识方法及装置。

### 背景技术

[0002] 目前,区块链技术得到了广泛应用,其去中心化的模式保证了数据不易被篡改,从而提升了安全性。

[0003] 在实际应用场景下,区块链能够为客户端提供相应的业务服务,其中的区块链节点可以针对用户的业务请求进行处理,生成相应的处理结果。但区块链中可能包含恶意节点或故障节点,这无疑将影响客户端所获得的业务服务。为此,区块链中各节点之间可进行诸如拜占庭容错算法(Practical Byzantine Fault Tolerance, PBFT)的共识过程,以使得各节点能够共同认可正确的处理结果。

[0004] 以基于PBFT的共识过程为例,PBFT的共识过程包含预准备阶段(pre-prepare)、准备阶段(prepare)、确认阶段(commit),如图1所示。接收到客户端(Client,图1中以C表示)的业务请求的节点(如图中编号为0的节点),会将该业务请求发送给其他节点(如图中编号为1、2、3的节点)进行共识。在每个阶段,各节点均会向彼此发送共识消息,使得各节点进行共识。经过三个阶段的共识过程,可视为共识达成,各节点便会分别处理业务请求,并向客户端反馈处理结果。

[0005] 在现有技术中的一些场景下,为了能够处理大量的共识过程,上述区块链的每个节点中,通常设置多个服务器,不同的服务器可分别参与不同的共识过程,以提升区块链的处理量及处理效率。

[0006] 但是,在实际应用中,节点中的服务器存在下线或重启等异常现象,以PBFT的共识过程为例,一旦服务器出现异常现象,将导致服务器不能继续参与共识并影响共识达成的概率,若某次共识未达成,则无论区块链处于任何一种共识阶段,均会从预准备阶段起重新发起共识。显然,这无疑会影响区块链共识的效率,并进一步影响区块链对业务处理的效率。

### 发明内容

[0007] 本申请实施例提供一种基于区块链的共识方法及装置,用以解决目前节点中的服务器发生异常时导致共识效率较低的问题。

[0008] 本申请实施例提供的一种基于区块链的共识方法,一个区块链节点包括第一服务器、第二服务器和至少一个数据库构成,所述方法包括:

[0009] 所述数据库存储执行共识所需的共识数据,供所述第一服务器和所述第二服务器在执行共识的过程中进行调用;

[0010] 所述第一服务器在执行共识前或执行共识时出现异常;

[0011] 所述第二服务器代替所述第一服务器,从所述数据库中获取共识数据,并根据所述共识数据执行共识,生成共识结果;

- [0012] 所述第二服务器将所述共识结果存储在所述数据库中。
- [0013] 本申请实施例提供一种基于区块链的共识装置,一个区块链节点包括第一服务器、第二服务器和至少一个数据库,其中,
- [0014] 所述数据库存储执行共识所需的共识数据,供所述第一服务器和所述第二服务器在执行共识的过程中进行调用;
- [0015] 所述第一服务器在执行共识前或执行共识时出现异常;
- [0016] 所述装置包括:
- [0017] 获取模块,根据所述共识消息,从所述数据库中获取与所述共识消息对应的共识数据;
- [0018] 共识执行模块,根据所述共识数据执行共识,生成共识结果;
- [0019] 存储模块,将所述共识消息及所述共识结果存储在所述数据库中。
- [0020] 本申请实施例提供一种基于区块链的共识方法及装置,对于区块链节点中的每一服务器而言,参与了某次共识过程的服务器,会将不同共识阶段的共识消息或自身所生成的当前阶段的共识结果,进行“公开”保存,即,存储在区块链节点内的数据库中,而该数据库对于区块链节点内的所有服务器均可使用,这样一来,如果参与某次共识的服务器出现下线或重启等异常现象,该服务器的共识数据可被区块链节点内的其他服务器使用,并可代替出现异常现象的服务器继续执行相应的共识过程。
- [0021] 显然,相较于现有技术,区块链节点中的各服务器将共识数据存储在区块链节点内数据库的方式,即使在某一服务器出现异常时,其他正常运行的服务器也能够从数据库中获取到相应的共识数据,接替异常的服务器完成共识,也就保证了共识过程的正常进行,在减少重新发起共识次数的同时,能够在一定程度上提升共识过程的成功率,进而提升区块链的业务处理效率。

### 附图说明

- [0022] 此处所说明的附图用来提供对本申请的进一步理解,构成本申请的一部分,本申请的示意性实施例及其说明用于解释本申请,并不构成对本申请的不当限定。在附图中:
- [0023] 图1为现有技术中基于PBFT的共识过程;
- [0024] 图2a为本申请实施例提供的基于区块链的共识过程中区块链的架构;
- [0025] 图2b为本申请实施例提供的基于区块链的共识过程中区块链节点的架构。
- [0026] 图2c为本申请实施例提供的在服务器侧的基于区块链的共识过程;
- [0027] 图2d为本申请实施例提供的另一种区块链节点的架构;
- [0028] 图3a和3b为本申请实施例提供的一种同一共识过程中的服务器变更过程示意图;
- [0029] 图4为本申请实施例提供的一种在服务器侧的基于区块链的共识装置结构示意图。

### 具体实施方式

- [0030] 为使本申请的目的、技术方案和优点更加清楚,下面将结合本申请具体实施例及相应的附图对本申请技术方案进行清楚、完整地描述。显然,所描述的实施例仅是本申请一部分实施例,而不是全部的实施例。基于本申请中的实施例,本领域普通技术人员在没有做

出创造性劳动前提下所获得的所有其他实施例,都属于本申请保护的范畴。

[0031] 正如前述,在任意一次区块链内各节点之间的共识过程中,任一节点内参与本次共识的服务器可能会出现下线或重启等异常现象,导致该服务器不能继续参与共识,从而有可能会降低共识成功率,特别是在采用PBFT共识的情况下,一定次数的共识失败,便可能触发PBFT下的垃圾回收机制,清除各服务器中的共识数据,显然,这无疑将影响区块链的业务处理。

[0032] 基于此,本申请实施例中提供一种基于区块链的共识方法,通过将共识数据存储在区块链节点中的数据库,使得当服务器出现异常(下线或重启)时,该区块链节点中其他正常运行的服务器也能够读取已存储在数据库中的共识数据,代替出现异常现象的服务器继续执行共识。当然,本申请实施例中的基于区块链的共识方法并不仅限于基于PBFT的共识过程,还可用于诸如Paxos等共识算法的共识过程。

[0033] 需要说明的是,在本申请实施例中,基于区块链的共识方法所采用的架构如图2a所示,从图2a中可见,区块链中包含多个区块链节点,为了便于后续描述,以下将区块链节点简称为:节点。

[0034] 多个客户端可与该区块链进行业务交互,其中,所述的区块链,可以是联盟链和/或私有链,能够向用户提供业务服务。所述的客户端,可包括浏览器、应用等,客户端可以运行在诸如终端、服务器、数据库中,这里不作具体限定。

[0035] 基于图2a所示的架构,对于任一节点而言,其架构可如图2b所示。在图2b中可见,节点中包含n个服务器以及一个共用于各服务器的数据库。不同的服务器可参与不同的共识过程,各服务器之间可以彼此独立运行。所述的数据库用于为该节点中的各服务器提供数据存储服务。也就是说,各服务器在共识过程中,可将相应的共识数据存储在该数据库中。当然,图2b所示的节点中的数据库的数量仅是一种通用数量,在实际应用中,节点中的数据库的数量可以根据实际应用中的需要进行调整。此外,在一些场景下,节点中的服务器可替换为计算机等具有计算处理功能的设备。

[0036] 还需要说明的是,为了便于描述,在下文中,将可能在运行中出现异常的服务器称为:第一服务器。将能够正常运行的服务器称为:第二服务器。那么,在图2b所示的架构中,也就可视为包含了两类服务器,即,第一服务器和第二服务器。以上内容并不应作为对本申请的限定。

[0037] 基于图2b中所示的关系架构,本申请实施例提供的一种基于区块链的共识过程,如图2c所示,该过程具体包括以下步骤:

[0038] S201:所述数据库存储执行共识所需的共识数据,供所述第一服务器和所述第二服务器在执行共识的过程中进行调用。

[0039] 基于如图2b中的架构,所述的数据库设置于该服务器所在的节点中,且共用于该节点中的所有服务器,该节点中的任一服务器在接收到共识消息或生成了共识结果后,会存储在数据库中。那么,服务器便可从该数据库中获取执行共识所需的共识数据。其中,所述的共识数据,可包括服务器所接收到的来自于其他节点中服务器发送的共识消息、服务器自身所生成的共识结果、由客户端发送的能够触发共识过程的业务请求等。

[0040] 这里需要说明的是,正是采用将共识数据存储于数据库中的方式,保证了共识数据对节点中的所有服务器可用。也就是说,如果第一服务器出现异常,虽然第一服务器不能

继续参与共识,但节点中的其他第二服务器可基于数据库中存储的共识数据,代替出现异常的服务器执行共识。

[0041] S202:所述第一服务器在执行共识前或执行共识过程中出现异常时,所述第二服务器代替所述第一服务器,从所述数据库中获取共识数据,并根据所述共识数据执行共识,生成共识结果。

[0042] 在实际运行中,参与共识的第一服务器可能会出现异常(如:宕机、重启等)。发生异常的时机通常是随机的,但无论异常发生在执行共识前或执行共识时,都将会影响共识过程的完成。此时,为了保证共识过程不受影响,故区块链节点内的其他正常运行的第二服务器便可以代替该异常的第一服务器执行共识。

[0043] 正是由于第一服务器出现了异常,不能继续执行共识,所以,第二服务器可代替第一服务器接收共识消息,参与到原本由第一服务器所参与的共识过程。

[0044] 如前所述,节点中的任一服务器均会将共识数据存储于数据库中,所以,第二服务器便可以在数据库中查找并获取共识过程所需的共识数据,以代替出现异常的第一服务器执行共识,并进一步生成相应的共识结果。

[0045] 显然,由正常运行的服务器代替异常的服务器执行共识,保证了共识过程不受影响。

[0046] S203:所述第二服务器将所述共识结果存储于所述数据库中。

[0047] 在本申请实施例中,代替第一服务器执行共识的第二服务器,同样会按照本申请中的数据存储机制,将共识结果作为一种共识数据,存储于数据库中。

[0048] 通过上述步骤,对于区块链节点中的每一服务器而言,参与了某次共识过程的服务器,会将不同共识阶段的共识消息或自身所生成的当前阶段的共识结果,进行“公开”保存,即,存储于该节点内的数据库中,而该数据库对于该节点内的所有服务器均可使用,这样一来,如果参与某次共识的服务器出现下线或重启等异常现象,该服务器的共识数据可被节点内的其他服务器使用,并可代替出现异常现象的服务器继续执行相应的共识过程。

[0049] 显然,相较于现有技术,节点中的各服务器将共识数据存储于节点内数据库的方式,即使在某一服务器出现异常时,其他正常运行的服务器也能够从数据库中获取到相应的共识数据,接替异常的服务器完成共识,也就保证了共识过程的正常进行,在减少重新发起共识次数的同时,能够在一定程度上提升共识过程的成功率,进而提升区块链的业务处理效率。

[0050] 这里需要说明的是,在共识时,对于节点中的任一服务器而言,将会接收到参与同一共识过程的其他设备所发送的共识消息。其中,所述的其他设备包括但不限于:参与共识过程的其他节点和/或客户端。所述的共识消息,可以包括客户端发送的、能够触发共识的业务请求,也可以包括其他节点在共识过程中发送的共识数据。

[0051] 显然,针对某一次共识过程,如果第一服务器出现了异常,则第二服务器将会代替该第一服务器接收共识消息,以便参与到第一服务器的共识过程中。

[0052] 所以,所述第一服务器在执行共识前出现异常时,所述第二服务器代替所述第一服务器,从所述数据库中获取共识数据,可包括:当所述第一服务器在执行共识前出现异常时,所述第二服务器代替所述第一服务器,接收参与共识过程的其他设备发送的共识消息,根据所述共识消息,从所述数据库中获取与所述共识消息对应的共识数据。

[0053] 此外,在本申请实施例中,存在一种重试机制,具体而言,参与共识过程的其他设备,在发送了某一共识消息或业务请求后,若在设定的时间内为接收到对方的响应,则会再次发送相同的共识消息或业务请求。可见,如果第一服务器在执行共识过程中出现了异常,便不能够在设定的时间向其他设备做出响应。

[0054] 所以,当所述第一服务器在执行共识过程中出现异常时,所述第二服务器代替所述第一服务器,接收参与共识过程的其他设备重试发送共识消息,根据所述共识消息,从所述数据库中获取与所述共识消息对应的共识数据。

[0055] 第二服务器在从数据库中获取共识数据时,通常将基于共识消息中的相应标识,下面将说明从数据库中获取共识数据的过程:

[0056] 在PBFT共识中,一个业务请求对应一次共识过程,而接收到客户端业务请求的节点(也称为主节点)会针对业务请求进行编号,那么,也就是说,业务请求的编号可以唯一对应一次共识过程。

[0057] 具体来说,节点中的任一服务器所参与的某次共识过程,该共识过程中的业务请求的编号(即,业务请求的标识),能够唯一地标识一次共识过程,通过业务请求的编号,可以区别于节点中其他服务器所参与的共识过程。所以,如果服务器接收到了携带有某一业务请求编号的共识消息,便可根据该业务请求编号在数据库中获取具有相同业务请求编号的共识数据(获取到的这些共识数据与共识消息属于同一共识过程)。

[0058] 例如:在共识过程中,共识消息的格式可以为:<共识阶段标识,视图编号,业务请求编号,业务请求摘要>。假设,服务器接收到某一共识消息为:<commit,v,n,D(m)>,其中,commit表示该节点已进入确认阶段,v表示视图编号,n表示业务请求的编号,D(m)表示发出通知消息的节点对业务请求的签名。则服务器可根据编号“n”,在数据库中查找该编号“n”对应的所有共识数据。

[0059] 基于此,第二服务器从所述数据库中获取与所述共识消息对应的共识数据的过程为:第二服务器根据所述共识消息中包含的业务请求的标识,从所述数据库中查找并获取与所述标识对应的共识数据。

[0060] 除了上述内容,在实际应用时,为了避免共识消息发送给已出现异常的第一服务器的情况,故在本申请实施例中,可通过节点上的网关实现共识消息的调配。也即,所述区块链节点还包括网关,此时,节点的架构可如图2d所示,可见,网关负责为节点中的服务器调配共识消息,一旦第一服务器出现异常,则网关便不会将共识消息发送给该异常的第一服务器,而是将该共识消息发送给正常运行的第二服务器。

[0061] 所以,具体而言,对于第二服务器,接收共识消息的过程可以为:

[0062] 当所述网关确定出所述第一服务器在执行共识前出现异常时,所述网关将接收到的、参与共识过程的其他设备发送的共识消息,转发给正常运行的第二服务器。那么,所述第二服务器接收所述网关转发的、参与共识过程的其他设备发送的共识消息。

[0063] 而当所述网关确定出所述第一服务器在执行共识过程中出现异常时,所述网关将接收到的、参与共识过程的其他设备重试发送共识消息,转发给正常运行的第二服务器。相应地,所述第二服务器接收所述网关转发的、参与共识过程的其他设备重试发送共识消息。

[0064] 可以理解地,在本申请实施例中,网关会将共识消息转发给节点中未出现异常的第二服务器,换言之,网关需要获知节点中出现异常的服务器,以及仍保持正常运行的服务



器。

[0065] 在本申请实施例中,节点中的各服务器可通过与网关之间采用心跳机制进行通信,以使得网关能够获知各服务器的运行状态,即,所述网关通过以下方式确定所述第一服务器及第二服务器的运行状态:

[0066] 所述网关接收所述第一服务器及第二服务器按照预设周期向所述网关发送运行状态消息,所述网关根据所述运行状态消息,确定所述第一服务器及第二服务器的运行状态。

[0067] 也就是说,如果网关能够接收到周期性的运行状态消息,则网关可以确定发送该运行状态消息的服务器正常工作,而如果网关在一定时间内都未接收到某服务器的运行状态消息,则网关可以确定该服务器出现了异常。

[0068] 下面以一应用实例进行说明:

[0069] 例如:如图3a所示,假设在某一次共识过程中,节点1中的服务器11和节点2中的服务器21均参与了该次共识过程(图3a中用灰色表示两个服务器处于同一共识过程中)。在此假设,服务器21下线(即,在节点2中,服务器21为第一服务器),此时,服务器11向服务器21发送了某一共识消息,显然,由于服务器21下线,致使该服务器21并不会作出任何反馈,那么,服务器11等待了一定时间后,将发起重试(该重试又可看作是节点1发起的重试),即重新发送该共识消息,当节点2的网关接收到了节点1重试的共识消息后,将选定其内部某一正常工作的服务器来处理节点1的重试,如图3b所示,在本示例中,节点2中的服务器22被选中(即,服务器22为第二服务器),也即,网关会将重试发送的共识消息转发给服务器22,并由该服务器22接替服务器21执行共识。

[0070] 另外,对于第二服务器将共识数据存储至服务器的过程而言,作为本申请实施例中的一种方式,第二服务器可在接收到共识消息(该共识消息可认为是一种共识数据)后即将接收到的共识消息存储在数据库中,并在执行共识生成了对应于该共识消息的共识结果(该共识结果同样也可认为是一种共识数据)后,再将共识结果存储在数据库中。

[0071] 而作为本申请实施例中的另一种方式,第二服务器接收到共识消息后,将经过前述过程,等待生成了共识结果后,在将共识消息和共识结果一并存储在数据库中。

[0072] 以上两种存储方式并不构成对本申请的限定。

[0073] 以上为本申请实施例提供的基于区块链的共识方法,基于同样的思路,本申请实施例还提供一种基于区块链的共识装置,如图4所示,其中,一个区块链节点包括多个服务器和至少一个数据库,其中,所述数据库存储执行共识所需的共识数据,供所述第一服务器和所述第二服务器在执行共识的过程中进行调用;

[0074] 所述第一服务器在执行共识前或执行共识时出现异常;

[0075] 所述基于区块链的共识装置至少包括:接收模块401、获取模块402、共识执行模块403及存储模块404,其中:

[0076] 获取模块402,从所述数据库中获取共识数据;

[0077] 共识执行模块403,根据所述共识数据执行共识,生成共识结果;

[0078] 存储模块404,将所述共识结果存储在所述数据库中。

[0079] 当所述第一服务器在执行共识前出现异常时,所述接收模块401,接收参与共识过程的其他设备发送的共识消息;所述获取模块,根据所述共识消息,从所述数据库中获取与

所述共识消息对应的共识数据。

[0080] 当所述第一服务器在执行共识过程中出现异常时,所述接收模块401,接收参与共识过程的其他设备重试发送共识消息;所述获取模块,根据所述共识消息,从所述数据库中获取与所述共识消息对应的共识数据。

[0081] 其中,当所述其他设备在发送共识消息并经过设定时间未接收到响应时,则重试发送所述共识消息。

[0082] 所述共识消息中包含业务请求的标识;所述获取模块402,根据所述共识消息中包含的业务请求的标识,在所述数据库中,查找并获取与所述标识对应的共识数据。

[0083] 此外,一个区块链节点中还可以包括网关,在此情况下,所述装置至少还包括:网关接收模块405、运行状态确定模块406及转发模块407。其中:

[0084] 当所述运行状态确定模块406确定出所述第一服务器在执行共识前出现异常时,所述转发模块407,将接收到的、参与共识过程的其他设备发送的共识消息,转发给正常运行的第二服务器。

[0085] 当所述运行状态确定模块406确定出所述第一服务器在执行共识过程中出现异常时,所述转发模块407,将接收到的、参与共识过程的其他设备重试发送共识消息,转发给正常运行的第二服务器。

[0086] 网关接收模块405,接收所述第一服务器及第二服务器按照预设周期向所述网关发送运行状态消息;

[0087] 所述运行状态确定模块406,根据所述运行状态消息,确定所述第一服务器及第二服务器的运行状态。

[0088] 在20世纪90年代,对于一个技术的改进可以很明显地区分是硬件上的改进(例如,对二极管、晶体管、开关等电路结构的改进)还是软件上的改进(对于方法流程的改进)。然而,随着技术的发展,当今的很多方法流程的改进已经可以视为硬件电路结构的直接改进。设计人员几乎都通过将改进的方法流程编程到硬件电路中来得到相应的硬件电路结构。因此,不能说一个方法流程的改进就不能用硬件实体模块来实现。例如,可编程逻辑器件(Programmable Logic Device,PLD)(例如现场可编程门阵列(Field Programmable Gate Array,FPGA))就是这样一种集成电路,其逻辑功能由用户对器件编程来确定。由设计人员自行编程来把一个数字系统“集成”在一片PLD上,而不需要请芯片制造厂商来设计和制作专用的集成电路芯片。而且,如今,取代手工地制作集成电路芯片,这种编程也多半改用“逻辑编译器(logic compiler)”软件来实现,它与程序开发撰写时所用的软件编译器相类似,而要编译之前的原始代码也得用特定的编程语言来撰写,此称之为硬件描述语言(Hardware Description Language,HDL),而HDL也并非仅有一种,而是有许多种,如ABEL(Advanced Boolean Expression Language)、AHDL(Altera Hardware Description Language)、Confluence、CUPL(Cornell University Programming Language)、HDCal、JHDL(Java Hardware Description Language)、Lava、Lola、MyHDL、PALASM、RHDL(Ruby Hardware Description Language)等,目前最普遍使用的是VHDL(Very-High-Speed Integrated Circuit Hardware Description Language)与Verilog。本领域技术人员也应该清楚,只需要将方法流程用上述几种硬件描述语言稍作逻辑编程并编程到集成电路中,就可以很容易得到实现该逻辑方法流程的硬件电路。

[0089] 控制器可以按任何适当的方式实现,例如,控制器可以采取例如微处理器或处理器以及存储可由该(微)处理器执行的计算机可读程序代码(例如软件或固件)的计算机可读介质、逻辑门、开关、专用集成电路(Application Specific Integrated Circuit, ASIC)、可编程逻辑控制器和嵌入微控制器的形式,控制器的例子包括但不限于以下微控制器:ARC 625D、Atmel AT91SAM、Microchip PIC18F26K20以及Silicone Labs C8051F320,存储器控制器还可以被实现为存储器的控制逻辑的一部分。本领域技术人员也知道,除了以纯计算机可读程序代码方式实现控制器以外,完全可以通过将方法步骤进行逻辑编程来使得控制器以逻辑门、开关、专用集成电路、可编程逻辑控制器和嵌入微控制器等的形式来实现相同功能。因此这种控制器可以被认为是一种硬件部件,而对其内包括的用于实现各种功能的装置也可以视为硬件部件内的结构。或者甚至,可以将用于实现各种功能的装置视为既可以是实现方法的软件模块又可以是硬件部件内的结构。

[0090] 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。

[0091] 为了描述的方便,描述以上装置时以功能分为各种单元分别描述。当然,在实施本申请时可以把各单元的功能在同一个或多个软件和/或硬件中实现。

[0092] 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。

[0093] 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。

[0094] 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。

[0095] 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。

[0096] 在一个典型的配置中,计算设备包括一个或多个处理器(CPU)、输入/输出接口、网络接口和内存。

[0097] 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。

[0098] 计算机可读介质包括永久性和非永久性、可移动和非可移动媒体可以由任何方法或技术来实现信息存储。信息可以是计算机可读指令、数据结构、程序的模块或其他数据。计算机的存储介质的例子包括,但不限于相变内存(PRAM)、静态随机存取存储器(SRAM)、动态随机存取存储器(DRAM)、其他类型的随机存取存储器(RAM)、只读存储器(ROM)、电可擦除可编程只读存储器(EEPROM)、快闪记忆体或其他内存技术、只读光盘只读存储器(CD-ROM)、数字多功能光盘(DVD)或其他光学存储、磁盒式磁带,磁带磁磁盘存储或其他磁性存储设备或任何其他非传输介质,可用于存储可以被计算设备访问的信息。按照本文中的界定,计算机可读介质不包括暂存电脑可读媒体(transitory media),如调制的数据信号和载波。

[0099] 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。

[0100] 本领域技术人员应明白,本申请的实施例可提供为方法、系统或计算机程序产品。因此,本申请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,本申请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。

[0101] 本申请可以在由计算机执行的计算机可执行指令的一般上下文中描述,例如程序模块。一般地,程序模块包括执行特定任务或实现特定抽象数据类型的例程、程序、对象、组件、数据结构等等。也可以在分布式计算环境中实践本申请,在这些分布式计算环境中,由通过通信网络而被连接的远程处理设备来执行任务。在分布式计算环境中,程序模块可以位于包括存储设备在内的本地和远程计算机存储介质中。

[0102] 本说明书中的各个实施例均采用递进的方式描述,各个实施例之间相同相似的部分互相参见即可,每个实施例重点说明的都是与其他实施例的不同之处。尤其,对于系统实施例而言,由于其基本相似于方法实施例,所以描述的比较简单,相关之处参见方法实施例的部分说明即可。

[0103] 以上所述仅为本申请的实施例而已,并不用于限制本申请。对于本领域技术人员来说,本申请可以有各种更改和变化。凡在本申请的精神和原理之内所作的任何修改、等同替换、改进等,均应包含在本申请的权利要求范围之内。

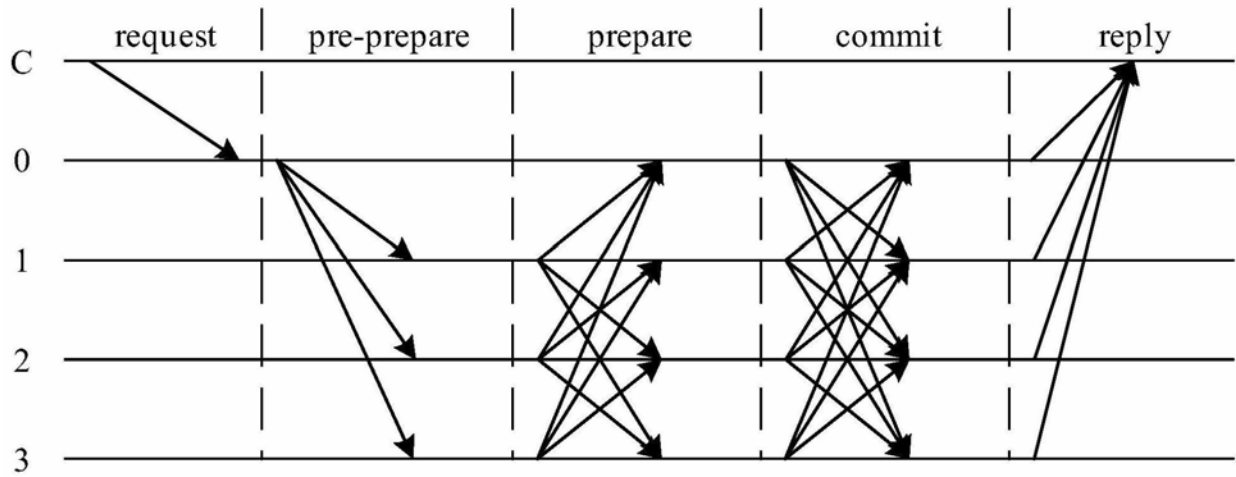


图1

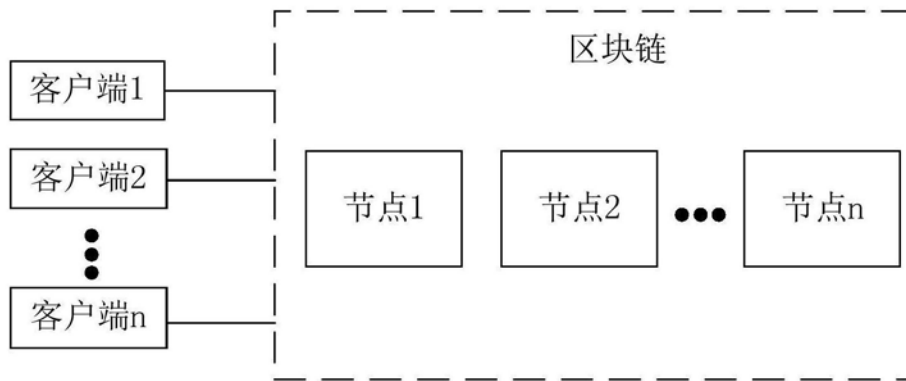


图2a

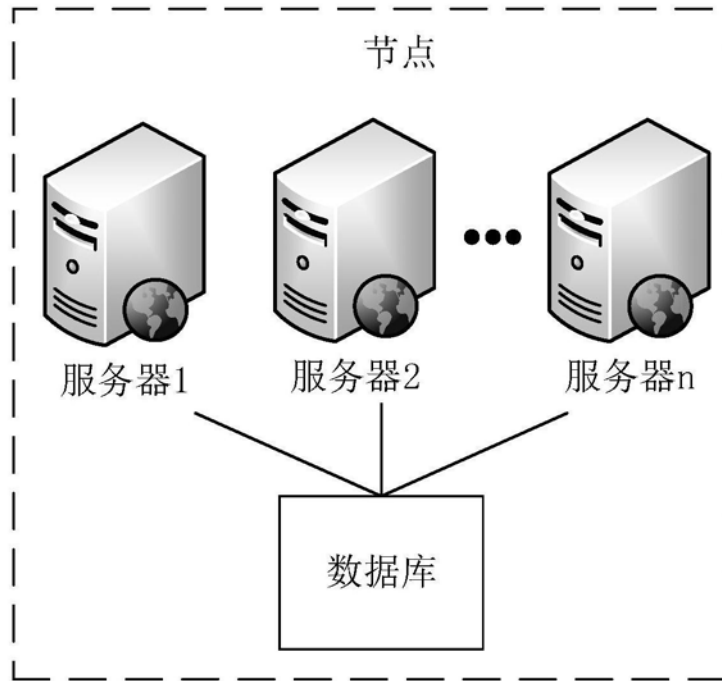


图2b

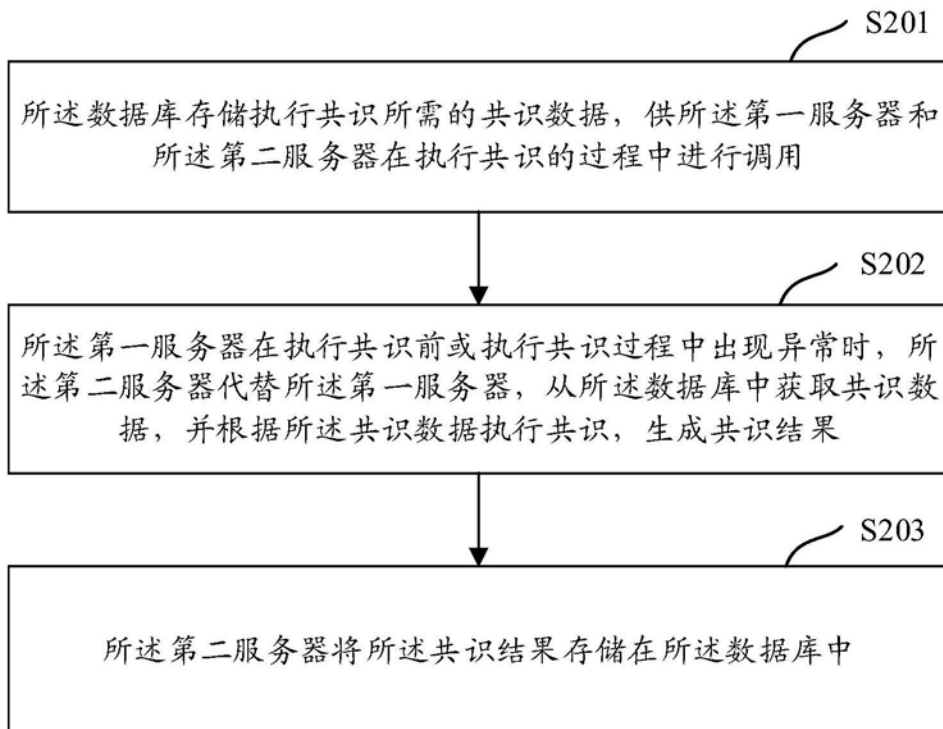


图2c

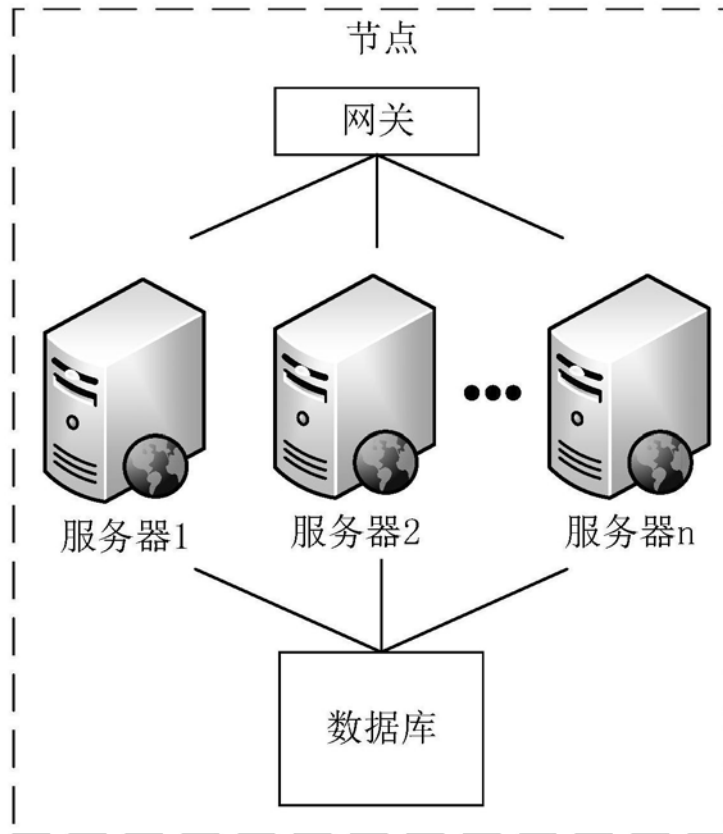


图2d

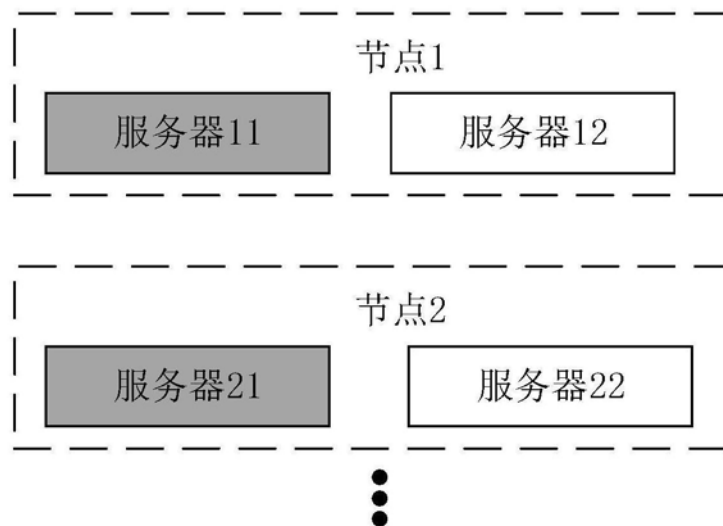


图3a

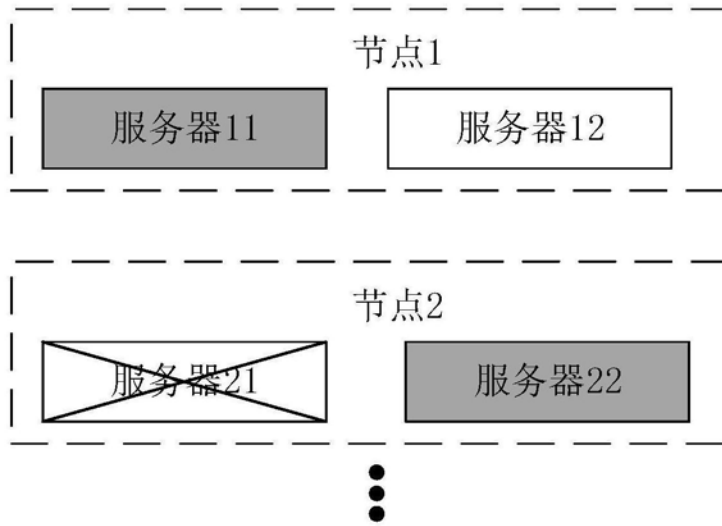


图3b



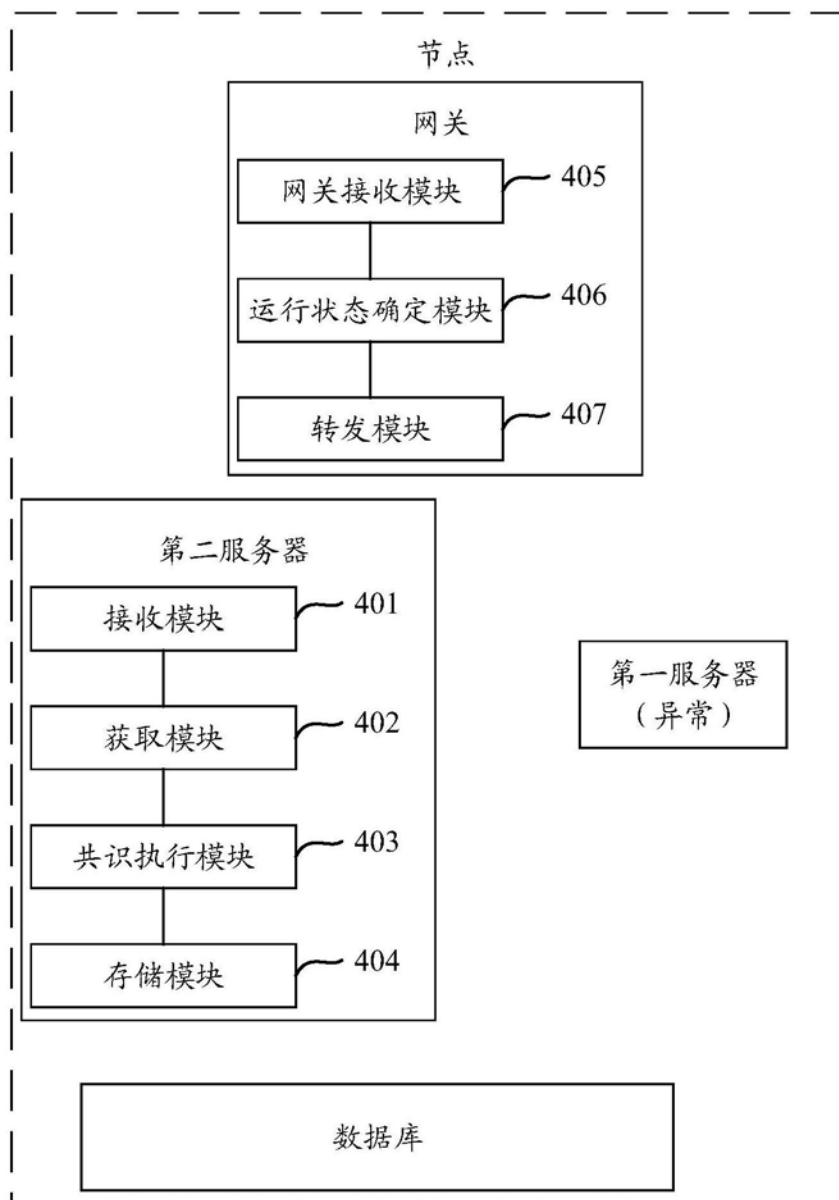


图4