

基于节点内存的数据处理方法、装置、设备以及介质

本申请要求于 2021 年 01 月 18 日提交中国专利局、申请号 202110063372.9、申请名称为“基于节点内存的数据处理方法、装置、设备以及介质”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。

5 技术领域

本申请涉及数据处理的技术领域，尤其涉及一种基于节点内存的数据处理技术。

背景技术

10 目前，当区块链系统中的矿工节点在挖矿时，由于哈希值的产生是随机的，再加上网络的延迟，就存在多个（例如，两个）矿工节点同时计算出满足条件的哈希值，进而存在同时挖出多个区块（例如，区块 1A 和区块 1B）的可能，当发生这种情况时，如果这两个矿工节点一起记账，即这两个矿工节点一起将区块 1A 和区块 1B 写入区块链的区块链数据库，则会造成该区块链的分叉现象。

15 由此可见，当将区块 1A 和区块 1B 中的数据一并写入区块链的区块链数据库之后，若需要查询或者验证某个区块（例如，区块 1A）中的交易业务，则需要从链上访问到该区块 1A 中的交易业务的相关数据，进而降低了数据访问的效率。

发明内容

本申请提供了一种基于节点内存的数据处理方法、装置、设备以及介质，可以提高数据访问效率。

20 本申请一方面提供了一种基于节点内存的数据处理方法，该方法由区块链节点执行，方法包括：

获取第一区块中的交易业务，在节点内存的链读缓存中获取与交易业务相关联的数据读取合约；第一区块为在节点内存中待写入分叉区块链中的区块；

调用数据读取合约获取与第一区块相关联的写缓存映射表，基于第一区块在写缓存映射表中的第一块缓存，确定与交易业务相关联的目标读取数据；

25 基于目标读取数据执行交易业务，将交易业务对应的目标执行结果作为目标缓存数据，将目标缓存数据存储至第一块缓存，将第一区块写入分叉区块链。

本申请一方面提供了一种基于节点内存的数据处理装置，该装置部署在区块链节点上，装置包括：

30 业务获取模块，用于获取第一区块中的交易业务，在节点内存的链读缓存中获取与交易业务相关联的数据读取合约；第一区块为在节点内存中待写入分叉区块链中的区块；

合约调用模块，用于调用数据读取合约获取与第一区块相关联的写缓存映射表，基于第一区块在写缓存映射表中的第一块缓存，确定与交易业务相关联的目标读取数据；

业务执行模块，用于基于目标读取数据执行交易业务，将交易业务对应的目标执行结果作为目标缓存数据，将目标缓存数据存储至第一块缓存，将第一区块写入分叉区块链。

5 本申请实施例一方面提供了一种计算机设备，包括：处理器和存储器；

处理器与存储器相连，其中，存储器用于存储计算机程序，计算机程序被处理器执行时，使得该计算机设备执行本申请实施例提供的方法。

10 本申请实施例一方面提供了一种计算机可读存储介质，计算机可读存储介质存储有计算机程序，该计算机程序适于由处理器加载并执行，以使得具有该处理器的计算机设备执行本申请实施例提供的方法。

本申请实施例一方面提供了一种计算机程序产品或计算机程序，该计算机程序产品或计算机程序包括计算机指令，该计算机指令存储在计算机可读存储介质中。计算机设备的处理器从计算机可读存储介质读取该计算机指令，处理器执行该计算机指令，使得该计算机设备执行本申请实施例提供的方法。

15 本申请实施例中的区块链节点（例如，区块链网络中的全量节点）在获取到第一区块（例如，打包生成的待写入分叉区块链的区块）时，可以预先为这个第一区块创建一个写缓存，并可以将为该第一区块创建的写缓存统称为第一块缓存，进而可以将该第一区块与该第一块缓存之间的映射关系添加至写缓存映射表。这样，当区块链节点需要对该第一区块中的交易业务进行验证时，可以快速从节点内存中的链读缓存中获取到与该交易业务相关联的数据读取合约（即热点合约），进而可以调用该数据读取合约获取前述写缓存映射表，

20 以在该写缓存映射表中快速对该第一区块的第一块缓存进行访问。此时，区块链节点可以快速获取到与该交易业务相关联的目标读取数据，进而可以提高数据访问的效率。区块链节点可以基于该目标读取数据执行该第一区块中记录的交易业务，从而可以在成功完成对这些交易业务的验证时，将相应的交易业务的交易执行结果分别写入该第一区块的第一块缓存。比如，当区块链节点按照打包时的交易执行顺序遍历执行这些交易业务时，可以快速从第一块缓存进行本地读取，进而可以提高数据访问的效率。此外，本申请实施例通过引入写缓存映射表（即一种支持分叉的缓存管理机制），可以在对该第一区块进行共识确认的过程中，将暂未确定属于最长链的第一区块中的交易业务的交易执行结果，作为目标缓存数据存储至第一块缓存，这样，该区块链节点可以在完成对该第一区块的共识确认时，

25 允许将该第一区块写入节点内存中的分叉区块链（这里指能够在节点内存中支持分叉的本地区块链），从而可以在节点内存中通过存储的这些区块的哈希值构成多级缓存，进而可以提高数据读取的效率。

30

附图说明

-3-

为了更清楚地说明本申请或现有技术中的技术方案，下面将对实施例或现有技术描述中所需要使用的附图作简单地介绍，显而易见地，下面描述中的附图仅仅是本申请的一些实施例，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他的附图。

5 图 1 是本申请实施例提供的一种网络架构的示意图；

图 2 是本申请实施例提供的一种在节点内存中执行第一区块以及提交目标区块的场景示意图；

图 3 是本申请提供的一种基于节点内存的数据处理方法的流程示意图；

图 4 是本申请实施例提供的一种获取第一区块的场景示意图；

10 图 5 是本申请实施例提供的一种在节点内存中创建链读缓存的场景示意图；

图 6 是本申请实施例提供的一种在节点内存中创建写缓存的场景示意图；

图 7 是本申请实施例提供的一种区块链数据结构的场景示意图；

图 8 是本申请实施例提供的一种基于块链结构实现多级缓存的场景示意图；

图 9 是本申请实施例提供的一种调用数据读取合约进行数据读取的场景示意图；

15 图 10 是本申请实施例提供的一种基于节点内存的数据处理方法的流程示意图；

图 11 是本申请实施例提供的一种提交目标区块的场景示意图；

图 12 是本申请实施例提供的一种进行数据删除的场景示意图；

图 13 是本申请提供的一种基于节点内存的数据处理装置的结构示意图；

图 14 是本申请实施例提供的一种计算机设备的结构示意图。

20 具体实施方式

下面将结合本申请中的附图，对本申请中的技术方案进行清楚、完整地描述，显然，所描述的实施例仅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作出创造性劳动前提下所获得的所有其他实施例，都属于本申请保护的范围。

25 请参见图 1，图 1 是本申请实施例提供的一种网络架构的示意图。如图 1 所示的网络架构可以应用于区块链系统，该区块链系统可以由多个节点通过网络通信的形式连接形成的分布式系统。其中，该区块链系统可以包含但不限于联盟链所对应的区块链系统。

其中，可以理解的是，区块链是一种分布式数据存储、点对点传输、共识机制以及加密算法等计算机技术的新型应用模式，主要用于对数据按时间顺序进行整理，并加密成账本，使其不可被篡改和伪造，同时可进行数据的验证、存储和更新。区块链本质上是一个

30

-4-

去中心化的数据库，该数据库中的每个节点均存储一条相同的区块链，区块链网络将节点区分为核心节点和轻节点，其中核心节点可以负责区块链全网的共识，也就是说核心节点可以为区块链网络中的共识节点。对于区块链网络中交易数据被写入账本（例如，本地账本）的过程可以为，客户端发送交易数据至轻节点，随后该交易数据以接力棒的方式在区块链网络中的轻节点之间传递，直到共识节点收到该交易数据，共识节点再将该交易数据打包进区块，以便于后续可以与其他共识节点之间进行共识，从而可以在共识通过后，将携带该交易数据的区块写入本地账本（比如，写入分叉区块链），以便于后续可以将携带该交易数据的区块写入至目标区块链所对应的区块链数据库（也可以简称为数据库）。其中，这里的目标区块链可以理解为这些共识节点根据支持分叉特性的分叉区块链所构建得到的最长链。

其中，可以理解的是，区块链系统中可以包括有智能合约，该智能合约在区块链系统中可以理解为是一种区块链各节点（包括共识节点）可以理解并执行的代码，可以执行任意逻辑并得到结果。用户可以通过客户端发起一个交易业务请求的方式，调用区块链上已经部署的智能合约，随后，区块链上的数据节点或轻节点可以将该交易业务请求发送至共识节点，以调用各个共识节点中运行的智能合约执行该用户所请求的交易业务。应当理解，区块链中可以包括一个或多个智能合约，这些智能合约可以通过合约调用地址、合约标识号（Identity document, ID）或合约名称来进行区分，而客户端发起的交易业务请求中，也可以携带智能合约的合约调用地址或者合约标识号或合约名称，以此指定需要运行的智能合约。而若客户端所指定的智能合约为需要读取数据的合约（即数据读取合约），则各个共识节点会访问本地账本（例如，各节点可以快速访问到访问节点内存中通过块链结构所构建的多级缓存）来进行数据的读取，最后各个共识节点会互相验证各交易执行结果是否一致（也就是进行共识），若一致则可以将交易执行结果存入各自的本地账本的写缓存中，并可以将上述交易业务的交易执行结果返回至客户端。

应当理解，如图1所示的网络架构可以包含核心节点（即共识节点）集群，轻节点集群以及用户终端集群。其中，该核心节点集群可以包含多个核心节点，这里的多个核心节点具体可以包括图1所示的节点10a、节点10b、节点10c、节点10d。如图1所示，节点10a、节点10b、节点10d可以分别与节点10c进行网络连接，以构成图1所示的共识网络100a。可以理解的是，在该共识网络100a中，节点10a、节点10b、节点10d均能够通过与该节点10c之间的网络连接进行数据交互。此外，用户终端集群可以包含多个用户终端，这里的多个用户终端具体可以包含图1所示的用户终端3000a、用户终端3000b、...、用户终端3000c。如图1所示，用户终端3000a、用户终端3000b、...、用户终端3000c可以分别与节点10c进行网络连接，以便于能够通过该节点10c之间的网络连接进行数据交互。其中，该轻节点集群可以包含多个数据节点。这里的多个数据节点具体可以包括图1所示的数据节点4000a、数据节点4000b，该数据节点4000a、数据节点4000b可以作为轻量节点，分别与其他用户终端进行网络连接，以便于能够通过该数据节点4000a、数据节点4000b，使得其他用户终端与该节点10c进行数据交互。

其中，本申请实施例可以将共识网络 100a 中的每个核心节点（例如，节点 10a、节点 10b、节点 10c、节点 10d）统称为区块链节点。应当理解，这些区块链节点均可以用于维护同一区块链（比如，图 1 所示的区块链 10e 可以为上述目标区块链），该共识网络 100a 中的任意两个区块链节点之间可以形成点对点（P2P, Peer To Peer）网络，该点对点网络可以采用 P2P 协议，其中，该 P2P 协议是一个运行在传输控制协议（TCP, Transmission Control Protocol）协议之上的应用层协议。在分布式系统中，任何设备如服务器、终端等都可以加入而成为区块链节点，其中，每个区块链节点均可以包括硬件层、中间层、操作系统层和应用层。

其中，可以理解的是，本申请实施例可以为接入该区块链网络的任意一个角色（例如，任意一个个人用户、任意一个企业、任意一个机构等实体对象）绑定一个区块链节点，以将这些区块链节点所构成的区块链网络统称为联盟链网络。所以，如图 1 所示的节点 10a、节点 10b、节点 10c、节点 10d 可以分别与需要接入该联盟链网络中的相应角色（即相应业务场景下的实体对象）之间存在一一对应关系。这里的业务场景可以包含电子票据场景、社交场景、赊购场景、信贷场景等。此时，相应业务场景下的目标业务具体可以包含电子票据业务、社交业务、赊购业务、信贷业务等，这里将不对相应业务场景下的具体业务进行一一列举。

其中，可以理解的是，由于每个实体对象均可以对应一个区块链节点，所以，本申请实施例可以以实体对象为上述企业用户（即前述企业）为例，此时，与每个企业用户相关联的区块链节点可以为同一区块链节点（例如，上述图 1 所示的节点 10c 可以与多个企业用户所对应的用户终端进行数据交互）。比如，在区块链电子票据系统中，可以将每个开票企业所对应的电子票据业务（比如，注册业务、开票业务、票据转移业务等）统称为一笔交易业务。其中，开票企业 A 可以通过图 1 所示的用户终端 3000a 与图 1 所示的节点 10c 进行数据交互，以完成相应的交易；以此类推，开票企业 B 可以通过图 1 所示的用户终端 3000b 与图 1 所示的节点 10c 进行数据交互，以完成相应的交易；开票企业 C 可以通过图 1 所示的用户终端 3000c 与图 1 所示的节点 10c 进行数据交互，以完成相应的交易。

其中，可以理解的是，本申请实施例可以将针对上述电子票据业务发送交易业务请求的实体对象（例如，开票企业 A、开票企业 B、...、开票企业 C）统称为第一用户，并可以将接收第一用户（例如，开票企业 A、开票企业 B、...、开票企业 C）所发送的交易业务请求的区块链节点统称为上述数据节点，还可以将参与对该交易业务请求进行共识的区块链节点统称为上述核心节点。

在共识网络 100a 中，由于上述节点 10c 可以与其存在网络连接（也可以称之为会话连接）的其他区块链节点之间进行数据同步，即上述节点 10c 可以从其他区块链节点上同步相应的业务数据信息（例如，可以从其他区块链节点上同步其他企业用户所发送的交易业务请求等），此时，与每个企业用户相关联的数据节点可以为不同的区块链节点。比如，开票企业 A 也可以通过图 1 所示的用户终端 3000a 与图 1 所示的节点 10c 进行数据交互；开票企业 B 也可以通过图 1 所示的用户终端 3000b 与图 1 所示的节点 10b 进行数据交互；

-6-

开票企业 C 也可以通过图 1 所示的用户终端 3000c 与图 1 所示的节点 10a 进行数据交互。由于区块链网络中的这些节点均会维护同一区块链，所以，通过将不同用户终端所发送的交易业务请求随机分配给与上述区块链 10e 相关联的区块链节点，可以有效地均衡区块链网络中的网络负载，从而可以提高相应业务所对应的业务数据的处理效率。

5 又比如，在信贷系统中，可以将第一用户（例如，上述用户终端 3000a 对应的用户）所请求的信贷业务理解为另一笔交易业务。其中，第一用户可以通过图 1 所示的用户终端 3000a 与图 1 所示的节点 10c 进行数据交互，以向图 1 所示的节点 10c 发送针对相应业务的交易业务请求。即在该信贷系统中，可以为该接入该信贷系统中的每个用户配置一个区块链节点，以接收相应用户所发送的交易业务请求。应当理解，本申请实施例中的每个用户
10 也可以统称为实体对象，比如，请求前述信贷业务的实体对象可以为前述个人用户、企业用户等。

可以理解的是，当数据节点接收到交易业务请求时，可以将该第一用户所发起的交易业务请求转发给核心节点，以通过核心节点对该第一用户所发起的交易业务请求进行合法性验证。这样，核心节点可以在合法性验证通过时，将该第一用户所请求的交易业务添加
15 至交易池，以便于后续可以将与该交易业务相关联的交易数据打包成区块，以便于后续可以与其他共识节点之间进行共识，从而可以在共识通过后，将携带该交易业务的交易数据的区块写入上述本地账本，以便于后续可以将携带该交易数据的区块写入至上述区块链数据库（也可以简称为数据库）。

为便于理解，请参见图 2，图 2 是本申请实施例提供的一种在节点内存中执行第一区块以及提交目标区块的场景示意图。其中，如图 2 所示的第一区块可以是在当前时刻为 T1
20 时刻由上述共识网络 100a 中的某个区块链节点（例如，节点 10c）对交易池中的多个交易业务的交易数据进行打包后所得到的、待写入分叉区块链的区块。可以理解的是，本申请实施例可以将分叉链 200a 和分叉链 200b 统称为分叉链区块链。其中，分叉链 200a 可以为尚未写入第一区块之前的分叉区块链。分叉链 200b 可以为成功写入该第一区块之后的分叉
25 区块链。如图 2 所示的第一区块中可以包含多个交易业务；这里的多个交易业务具体可以包括：图 2 所示的交易 1、交易 2、交易 3、...、交易 n。

其中，可以理解的是，为避免链上存储资源的浪费，本申请实施例提出可以在将打包生成的区块在进行共识处理后，将当前共识成功的区块临时写入图 2 所示的节点内存的分叉区块链上（例如，能够支持分叉的分叉链 200a），以得到图 2 所示的分叉链 200b。可以
30 理解的是，该分叉链 200b 可以为成功将当前区块写入分叉链 200a 后所形成的新的分叉区块链。

如图 2 所示，区块链节点（例如，节点 10c）可以在对第一区块进行共识处理（即执行第一区块中的交易业务）之后，可以将该第一区块中的交易业务的交易执行结果保存到该第一区块的写缓存中。如图 2 所示，该第一区块的区块哈希值可以为图 2 所示的写缓存

映射表 300a 中的区块哈希值 11, 另外, 该第一区块所对应的写缓存可以为图 2 所示的写缓存映射表 300a 中的块缓存 11。

比如, 当上述节点 10c 打包得到待写入分叉区块链中的第一区块时, 可以将该第一区块广播给该共识网络 100a 中的其他区块链节点 (例如, 上述图 1 所示的节点 10a、节点 10b 和节点 10d), 以使得该共识网络 100a 中的其他区块链节点 (即节点 10a、节点 10b 和节点 10d), 可以基于共识机制对该需要写入分叉区块链 (例如, 支持分叉的分叉链 200a) 的第一区块进行共识处理, 即其他区块链节点可以在各自节点内存的链读缓存中获取与该第一区块中的每个交易业务相关联的数据读取合约, 进而可以调用获取到的数据读取合约, 快速得到与该第一区块相关联的写缓存映射表, 比如, 这些区块链节点所获取到的写缓存映射表均可以为图 2 所示的写缓存映射表 300a。

其中, 可以理解的是, 当区块链节点获取到第一区块时, 可以在节点内存中为该第一区块创建对应的写缓存, 并可以将为第一区块创建的写缓存统称为第一块缓存, 该第一块缓存可以为图 2 所示的区块哈希值 11 所对应的块缓存 11。该第一块缓存可以用于存储执行上述交易 1 后的交易执行结果 (例如, 图 2 所示的结果 1), 同理, 该第一块缓存还可以用于存储执行上述交易 2 后的交易执行结果 (例如, 图 2 所示的结果 2)。以此类推, 该第一块缓存可以用于存储执行第一区块中的每个交易业务之后所得到的交易执行结果。

其中, 可以理解的是, 本申请实施例提供了一种支持链分叉的缓存管理机制, 进而可以在执行第一区块之后, 将该第一区块写入支持分叉的分叉链 200a, 以得到图 2 所示的分叉链 200b。其中, 本申请实施例可以将当前最新写入该分叉链 200b 中的区块 (例如, 图 2 所示的当前具有最大区块生成时间戳的区块 11) 统称为上述第一区块。与此同时, 如图 2 所示, 区块链节点可以基于共识机制中的最长链选取策略, 预先在节点内存中完成对支持分叉的区块链 (例如, 已成功写入第一区块的分叉链 200b) 的最长链的选择。这样, 区块链节点可以在节点内存中所存储的分叉区块链 (例如, 图 2 的分叉链 200b) 上获取与该第一区块相关联的目标区块, 该目标区块可以为图 2 所示的区块 5。这样, 当区块链节点确定该目标区块属于分叉链 200b 的最长链时, 则可以将该目标区块写入图 2 的目标区块链 400a, 以得到图 2 所示的包含目标区块的目标区块链 400b。

其中, 可以理解的是, 该写缓存映射表 300a 可以用于维护每个区块的区块哈希值与每个区块的块缓存 (也可以称之为上述写缓存) 之间的一一映射关系, 这些映射关系均以 KV 键值对的形式呈现。比如, 如图 2 所示, 每个区块的区块哈希值可以作为 KV 键值对中的 key (即键), 而对应区块的块缓存可以作为 KV 键值对中的 value (即值)。具体的, 对于图 2 所示的分叉链 200b 中的区块 1 而言, 该区块 1 的区块哈希值与该区块 1 的块缓存之间的映射关系, 可以记为 (区块哈希值 1, 块缓存 1); 同理, 对于图 2 所示的分叉链 200b 中的区块 2 而言, 该区块 2 的区块哈希值与该区块 2 的块缓存之间的映射关系, 可以记为 (区块哈希值 2, 块缓存 2); 同理, 对于图 2 所示的分叉链 200b 中的区块 3 而言, 该区块 3 的区块哈希值与该区块 3 的块缓存之间的映射关系, 可以记为 (区块哈希值 3, 块缓存 3); 以此类推, 对于图 2 所示的分叉链 200b 中的区块 11 (即第一区块) 而言, 该区块 11 的区

块哈希值与该区块 11 的块缓存之间的映射关系，可以记为（区块哈希值 11，块缓存 11）。其中，该区块哈希值 11 可以为图 2 所示的包含 n 个交易业务的第一区块的区块哈希值，块缓存 11 可以为图 2 所示的存储有 n 个交易执行结果的第一块缓存。

5 考虑到图 2 所示的分叉链 200b 中的每个区块均保存有前一区块的区块哈希值。于是，本申请实施例可以根据分叉链 200b 中的这些区块的哈希索引关系（例如，后向哈希索引关系），将这些区块的写缓存组合成一个逻辑的多级缓存，即当前区块的上一级缓存即为前一区块的写缓存。比如，在分叉链 200b 中，在当前执行的区块为区块 11 时，该区块 11 的上一级缓存就是图 2 的区块 10 的写缓存。同理，区块 10 的上一级缓存就是图 2 所示的区块 9 的写缓存。以此类推，区块 2 的上一级缓存就是图 2 所示的区块 1 的写缓存。

10 这意味着在该分叉链 200b 所对应的写缓存映射表 300a 中，区块 11 所映射的块缓存 11 的逻辑查询等级高于区块 10 所映射的块缓存 10 的逻辑查询等级。同理，区块 10 所映射的块缓存 10 的逻辑查询等级高于区块 9 所映射的块缓存 9。以此类推，区块 2 所映射的块缓存 2 的逻辑查询等级高于区块 1 所映射的块缓存 1 的逻辑查询等级。

15 为便于理解，本申请实施例以该共识网络 100a 中的节点 10c 作为具有缓存管理功能的区块链节点为例，以阐述在该节点 10c 的节点内存中执行数据处理方法的具体过程。如图 2 所示，该第一区块中可以包含多个交易业务，这里的每个交易业务可以为图 2 所示的交易 1、交易 2、交易 3、...、交易 n。如图 2 所示，当该节点 10c 对该第一区块进行共识时，可以通过获取到的数据读取合约执行该第一区块中的交易业务（例如，可以执行图 2 所示的交易 1）。其中，可以理解的是，该数据读取合约可以为该节点 10c 从该节点内存的链读缓存中所读取到的热点合约。

其中，可以理解的是，本申请实施例可以预先在节点内存中创建一个链读缓存，以通过该链读缓存来负责暂存一些常用的链数据，比如，一些频繁被调用的业务合约、或者经常发起合约调用请求的同一合约调用账户等。其中，本申请可以将目标时长内频繁发起合约调用请求的同一合约调用账户统称为热点调用账户。

25 其中，可以理解的是，由于区块中打包的 n 个交易业务之间存在一定的交易执行顺序。所以，当该节点 10c 对该第一区块进行共识时，则可以在该节点内存中将该第一区块中的这些交易业务遍历执行一遍，以验证这些交易业务的可靠性和真实性。比如，区块链节点（例如，节点 10c）可以将该第一区块中的交易 1 的交易执行结果（即图 2 所示的结果 1）暂时保持在对应区块的写缓存中，以便于后续在需要执行同一区块中的其他交易业务（例如，交易 2）时，可以自适应地将当前区块的写缓存（即图 2 所示的第一块缓存）作为最高逻辑查询等级的缓存。这样，该区块链节点（例如，节点 10c）在执行其他交易业务时，可以快速从当前区块的写缓存中查询用于执行其他交易业务（例如，交易 2）所需要的目标读取数据，进而可以基于查询到的目标读取数据执行其他交易业务，以将其他交易业务的交易执行结果分别写入当前区块的写缓存进行保存。其中，交易 2 的交易执行结果即图

30

2 所示的结果 2、交易 3 的交易执行结果即图 2 所示的结果 3、……、交易 n 的交易执行结果即图 2 所示的结果 n。

由此可见，本申请实施例通过将这些交易执行结果（例如，交易结果数据）缓存在节点内存中，可以大大减少对区块链数据库的访问，从而可以减少网络请求，这样，在执行某个区块时，可以提高相应交易业务所需要的目标读取数据的读取效率，进而可以提升交易执行的效率以及出块的速率，从而可以进一步提升目标区块链的吞吐量，比如，可以允许有更多属于最长链上的目标区块写入到目标区块链上。

其中，可以理解的是，当区块链节点（例如，节点 10c）将目标区块写入目标区块链时，可以一并将该目标区块的目标块缓存中的数据一并写入目标区块链对应的区块链数据库。然后，区块链节点（例如，节点 10c）可以将该目标块缓存中的数据作为目标区块链上的目标链数据，以将该目标链数据写入链读缓存，以便于能够为后续打包生成的新的区块提供一种快速访问热点调用账户或者热点合约的访问通道。

可以理解的是，本申请实施例还维护了一张哈希关系表，该哈希关系表中记录的是分叉区块链（例如，上述图 2 所示的分叉链 200b）中的这些区块的区块高度和区块哈希值之间的映射关系。由于该分叉链 200b 为在节点内存的本地支持链分叉的分叉区块链，所以，该分叉链 200b 中可能在同一区块高度上存在多个区块的现象，这意味着在该分叉链 200b 中可能存在两个或者更多数量的区块共同竞争在目标区块链上的同一位置的现象，以至于该分叉链 200b 出现分叉，本申请实施例可以将该分叉链 200b 中的分叉统称为分支链（或者备用链），例如，除区块 1、区块 2、区块 3 构成的主链之外，还包括分支链，这里的分支链至少可以包含分支链 1a 和分支链 1b。其中，分支链 1a 中的区块可以包含区块 4a、区块 5、区块 6、分支链 2a（区块 7b、区块 8b、区块 9、区块 10 和区块 11）和分支链 2b（区块 7a 和区块 8a）。其中，分支链 1b 中的区块可以包含区块 4b。

其中，本申请实施例可以将新添加的区块所在的分支链统称为第一分支链，并可以将分叉区块链中的其他分支链统称为第二分支链。比如，如图 2 所示的第一分支链上的可以为上述分支链 1a，第二分支链可以为上述分支链 1b。如图 2 所示，该第一分支链中的分支链 2a 可以为第一子分支链（即新的第一分支链），第一分支链中的分支链 2b 可以为第二子分支链（即新的第二分支链）。

其中，第一子分支链中的区块可以包含图 2 所示的区块 7b、区块 8b、区块 9、区块 10 和区块 11；其中，第二子分支链上的区块可以包含图 2 所示的区块 7a 和区块 8a。所以，如图 2 所示，如果新添加的区块（例如，图 2 所示的第一区块）可以使某条分支链（例如，上述分支链 1a）积累更多的工作量，那么，这条分支链（例如，上述分支链 1a）将有可能作为新的主链。那么，为了避免链上存储资源的浪费现象。本申请实施例提出可以在该节点内存的本地预先选择分叉区块链上的最长链，进而可以在确定目标区块属于分叉区块链上的最长链时，直接将目标区块的目标块缓存中的数据写入目标区块链对应的区块链数据库，进而可以确保所有写入区块链数据库中的数据均属于最长链上的数据，这样，对于分

又区块链上的不属于主链的其他分支链上的数据，则将不会写入区块链数据库，进而可以有效地避免链上存储资源的浪费。

其中，区块链节点（例如，节点 10c）基于数据读取合约确定与交易业务相关联的目标读取数据以及获取与第一区块相关联的目标区块的具体过程，可以参见如下图 3 至图 12 所对应的实施例。

请参见图 3，图 3 是本申请提供的一种基于节点内存的数据处理方法的流程示意图，如图 3 所示，方法可以由上述区块链网络中的区块链节点执行，该区块链节点中可以运行有基于节点内存的数据处理装置，该区块链节点可以为上述图 1 所示的共识网络 100a 中任意一个节点。方法具体可以包括以下 S101-S104。

S101，获取第一区块中的交易业务，在节点内存的链读缓存中获取与交易业务相关联的数据读取合约。

其中，第一区块为在节点内存中待写入分叉区块链中的区块；这里的分叉区块链可以为支持分叉的本地区块链。区块链节点可以在获取到包含交易业务的第一区块时，对第一区块进行共识处理，以生成用于对第一区块中的交易业务进行验证的业务处理指令；其中，业务处理指令中包含与交易业务相关联的热点读取辅助参数；热点读取辅助参数包括：用于执行交易业务的业务合约的合约调用地址；区块链节点可以从业务处理指令中提取合约调用地址，基于合约调用地址从节点内存的链读缓存中查找与合约调用地址所映射的合约名称相匹配的热点合约；热点合约可以为在链读缓存中所确定的累计访问频次大于访问阈值的业务合约；区块链节点可以将查找到的与合约名称相匹配的热点合约作为与交易业务相关联的数据读取合约。

为便于理解，本申请实施例所涉及的区块链节点可以为上述图 1 所对应实施中的核心节点，例如：节点 10c。可以理解的是，在本申请实施例中，每个节点（例如，共识网络 100a 中的节点）在进行正常工作时均可以用于接收外界发送的数据，并可以基于接收到的数据进行区块上链处理，还可以向外界发送数据。这里的区块链节点可以包括：手机、平板电脑、笔记本电脑、掌上电脑、智能音响、移动互联网设备（MID, mobile internet device）、POS（Point Of Sales，销售点）机、可穿戴设备（例如智能手表、智能手环等）、服务器等计算机设备。可以理解的是，这里的服务器可以是独立的物理服务器，也可以是多个物理服务器构成的服务器集群或者分布式系统，还可以是提供云服务、云数据库、云计算、云函数、云存储、网络服务、云通信、中间件服务、域名服务、安全服务、CDN、以及大数据和人工智能平台等基础云计算服务的云服务器。

为便于理解，请参见图 4，图 4 是本申请实施例提供的一种获取第一区块的场景示意图。其中，如图 4 所示的区块链节点可以接收用户 A 通过图 4 所示的用户终端 20a 发送的交易业务请求。如图 4 所示，当用户 A 需要向用户 B 转移一笔资产（例如，用户 A 需要向用户 B 转移虚拟电子资源）时，可以在图 4 所示的交易业务 K 的交易详情页面中针对“确认业务”控件执行触发操作，以执行图 4 所示的 S1，进而可以向图 4 所示的区块链节点发

送针对该交易业务 K 的交易业务请求。其中，可以理解的是，这里的虚拟电子资源可以包括但不限于上述电子票据、信贷资源、游戏币等虚拟资产。该交易业务请求可以用于指示该区块链节点执行图 4 所示的 S2，以通过图 4 所示的共识网络 500a 中的其他区块链节点验证该交易业务请求的合法性。例如，其他区块链节点可以对该交易业务请求中所携带的用户 A 的签名信息进行签名验证，进而可以在签名验证成功时，确定发起该交易业务 K 的用户 A 具备转移上述资产的权限。

这样，当区块链节点在确定交易业务请求为合法请求时，可以执行图 4 所示的 S3 在确定交易业务请求为合法请求时，将交易业务 K 的交易数据添加至交易池，即将该用户 A 所请求执行的交易业务 K（例如，用户 A 请求向用户 B 转移虚拟电子资源的业务）作为目标业务，进而可以将执行该目标业务后的交易数据（例如，该交易业务 K 的交易详情数据和该交易业务 K 的交易执行结果）添加至交易池，以便能于后续该区块链节点在被选举作为记账节点时，可以执行图 4 所示的 S4，即将交易池中的交易业务所对应的交易数据进行打包，以打包生成待写入分叉区块链的第一区块。其中，交易池中可以包括 n 个交易业务。

可以理解的是，这里的 n 可以为正整数，第一区块中的 n 个交易业务可以包括但不限于图 4 所示的交易业务 K。如图 4 所示，该区块链节点还可以用于将该包含交易业务 K 的第一区块广播至图 4 所示的共识网络 500a，以便于该共识网络 500a 中的其他区块链节点（即其他核心节点）可以对该第一区块进行共识，以验证该第一区块中的交易业务所对应的交易执行结果的有效性。

可以理解的是，区块链节点在获取到第一区块时，可以在该区块链节点的节点内存中预先为该第一区块创建对应的写缓存。应当理解，本申请实施例可以将为第一区块创建的写缓存统称为第一块缓存。这样，当图 4 所示的区块链节点通过共识层执行第一区块时，可以直接获取到该第一区块中的一个或者多个交易业务，进而可以生成用于对第一区块中的这些交易业务进行验证的业务处理指令；业务处理指令中可以包含与这些交易业务相关联的热点读取辅助参数；其中，这里的热点读取辅助参数可以包括：用于执行交易业务的业务合约的合约调用地址。这样，当区块链节点在获取到该合约调用地址时，可以基于该合约调用地址在节点内存的链读缓存中获取与交易业务相关联的数据读取合约，进而可以继续执行下述 S102。

其中，可以理解的是，这里的链读缓存可以用于暂存一些常用的链数据。可以理解的是，本申请实施例所涉及的缓存管理机制可以包括对链读缓存的管理以及对写缓存的管理。本申请实施例可以将这里的链读缓存和写缓存统称为本地缓存。可以理解的是，本申请实施例通过引入缓存管理机制，可以在执行某个区块（例如，上述第一区块）的过程中，将后续待写入区块链数据库中的数据（例如，执行第一区块中的交易业务的交易执行结果）作为缓存数据，临时写入当前执行的区块（即上述第一区块）的写缓存中，以便于后续可以在执行新的第一区块的过程中，可以在本地缓存中通过哈希链式结构实现全新的多级缓存。此外，本申请实施例还可以在执行完第一区块之后，将与第一区块相关联的目标区块

的写缓存中的数据作为可以写入区块链数据库中的链数据，并可以将最新得到的链数据暂存在链读缓存中，以便于后续在执行新的第一区块的过程中，可以快速访问到链读缓存中的热点数据。

其中，对链读缓存的管理具体可以包括：增加链读缓存中的链数据、使用链读缓存中的链数据以及删除链读缓存中的链数据。

比如，区块链节点可以在节点内存中预先为区块链（例如，上述目标区块链）创建一个链读缓存。为便于理解，请参见图 5，图 5 是本申请实施例提供的一种在节点内存中创建链读缓存的场景示意图。如图 5 所示，区块链节点可以在节点内存中预留一个缓存空间，以在该缓存空间创建得到图 5 所示的链读缓存 S1，可以理解的是，该链读缓存 S1 可以为上述链读缓存。应该理解，在节点内存的链读缓存（即图 5 所示的链读缓存 S1）可以用于暂存一些将会频繁被访问或者近期会再次访问的链数据，这里的链数据主要是指可以历史写入上述目标区块链所对应的区块链数据库中的数据。这样，当执行区块中的某个交易业务时，则可以快速且直接地从节点内存的链读缓存（即图 5 所示的链读缓存 S1）中找到相应的热点数据（例如，上述需要被频繁访问的热点合约等），而不需要直接通过网络访问到区块链数据库（即不需要直接通过网络访问到图 5 所示的数据库 S2），也就无需在区块链数据库中进行相关链数据的查询，进而可以有效地提升数据访问的效率。可以理解的是，这里的热点合约可以为在链读缓存（即链读缓存 S1）的单位累计周期内所统计到的累计访问频次大于访问阈值的业务合约，例如，该业务合约可以为上述用于从节点内存的写缓存（或者区块链数据库）中读取数据的数据读取合约。

反之，可以理解的是，对于节点内存中的链读缓存（即图 5 所示的链读缓存 S1）而言，为了不让上述链读缓存中的链数据占用大量的内存空间，本申请实施例还需要将一些访问不再频繁的链数据保存到图 5 所示的数据库 S2 中，并可以同步在节点内存的链读缓存中，删除这些累计访问频次小于或者等于访问阈值的链数据，以避免节点内存的内存资源的浪费。

其中，可以理解的是，写缓存可以用于暂存每个交易改动中的数据（即可以暂存每个交易业务的交易执行结果），例如，在上述用户 A 向用户 B 转移虚拟电子资源（例如，转移游戏币）的过程中，可以将写缓存用于暂存该用户 A 的用户账户地址中的账户资产余额等。应当理解，这里所涉及的对写缓存的管理具体可以包括：增加写缓存映射表中的写缓存、使用写缓存映射表中的写缓存以及删除写缓存映射表中的写缓存中所存储的数据。

比如，区块链节点可以在节点内存中预先为每个区块创建一个写缓存。为便于理解，请参见图 6，图 6 是本申请实施例提供的一种在节点内存中创建写缓存的场景示意图。如图 6 所示的区块链节点可以为分叉区块链中的每个区块创建一个写缓存。其中，为便于理解，如图 6 所示的写缓存映射表可以为上述图 2 所对应实施例中的写缓存映射表 300a。这意味着在图 6 所示的写缓存映射表中，可以存在多个 KV 键值对（简称多个键值对）。比如，图 6 所示的哈希 1 可以为上述图 2 所示的区块哈希值 1，写缓存 1 可以为上述图 2 所

示的块缓存 1。同理，图 6 所示的哈希 2 可以为上述图 2 所示的区块哈希值 2，写缓存 2 可以为上述图 2 所示的块缓存 2。以此类推，图 6 所示的哈希 N 可以为上述图 2 所示的区块哈希值 11，写缓存 N 可以为上述图 2 所示的块缓存 11。由此可见，本申请实施例可以在节点内存中动态的维护一张写缓存映射表。比如，对于图 6 所示的哈希 N 对应的区块（即上述第一区块）而言，可以将该第一区块中的交易业务的交易执行结果作为缓存数据，写入图 6 所示的写缓存 N。

S102，调用数据读取合约获取与第一区块相关联的写缓存映射表，基于第一区块在写缓存映射表中的第一块缓存，确定与交易业务相关联的目标读取数据。

其中，可以理解的是，在本申请实施例中，该节点内存的每个写缓存的逻辑查询等级均高于区块链数据库的逻辑查询等级。即该区块链节点在调用数据读取合约时，可以优先从节点内存的这些写缓存中按照逻辑查询等级遍历进行目标读取数据的查询，进而可以在遍历完这些写缓存，且未查找到目标读取数据的情况下，进一步访问该区块链数据库，以在区块链数据库中查询得到该目标读取数据。

其中，可以理解的是，对于上述支持分叉的分叉区块链中的区块而言，可以在块与块之间通过父区块哈希连接起来，以构成该分叉区块链的区块链数据结构。为便于理解，请参见图 7，图 7 是本申请实施例提供的一种区块链数据结构的场景示意图。其中，如图 7 所示的 3 个区块可以为上述节点内存中所存储的分叉区块链中的部分区块，这 3 个区块的区块数据结构均可以包括但不限于区块高度、前一区块的区块哈希值（即父区块哈希值）以及区块哈希值等数据。可以理解的是，对于图 7 所示的区块高度为 3 的区块（例如，上述图 2 所示的区块 3）而言，该区块 3 的区块哈希值（即图 7 所示的哈希字符串 0000...123）可以作为下一区块的父区块哈希值。同理，对于图 7 所示的区块高度为 4 的区块（例如，上述图 2 所示的区块 4a）而言，该区块 4a 的区块哈希值（即图 7 所示的哈希字符串 0000...456）也可以作为下一区块的父区块哈希值。以此类推，对于图 7 所示的区块高度为 5 的区块（例如，上述图 2 所示的区块 5）而言，该区块 5 的区块哈希值（即图 7 所示的哈希字符串 0000...789）也可以作为下一区块（例如，上述图 2 所示的区块 6）的父区块哈希值。基于此，这里将不对该节点内存中支持分叉的每个区块之间的连接关系进行一一列举。

可以理解的是，由于一个区块的区块哈希值可以用于唯一地标识一个特定的区块，且每个区块都保存有前一区块的区块哈希值，因此，本申请实施例可以通过上述图 7 所示的区块链数据结构中的哈希关系所构成的块链结构和上述图 6 所示的写缓存映射表，快速找到上一区块的写缓存。换言之，本申请实施例可以通过块链结构中的哈希关系，将各区块的写缓存组合成一个全新的多级缓存。为便于理解，请参见图 8，图 8 是本申请实施例提供的一种基于块链结构实现多级缓存的场景示意图。如图 8 所示的块链结构是将节点内存中的分叉区块链中的区块，按照哈希关系进行组合后所得到的。

其中，如图 8 所示的区块 1 的区块哈希值可以为图 8 所示的哈希 1，该哈希 1 可以作为该区块 1 的唯一标识符，此外，该区块 1 的前一区块的区块哈希值可以为图 8 所示的区

块 1 中的哈希 0。同理，图 8 所示的区块 2 的区块哈希值可以为图 8 所示的哈希 2，该哈希 2 可以作为该区块 2 的唯一标识符，此外，区块 2 中存储有前一区块（即图 8 所示的区块 1）的区块哈希值，例如，图 8 所示的区块 2 中的哈希 1。同理，图 8 所示的区块 3 的区块哈希值可以为图 8 所示的哈希 3，该哈希 3 可以作为该区块 3 的唯一标识符，此外，区块 3 中存储有前一区块（即图 8 所示的区块 2）的区块哈希值，例如，图 8 所示的区块 3 中的哈希 2。以此类推，图 8 所示的区块 N 的区块哈希值可以为图 8 所示的哈希 N，该哈希 N 可以作为该区块 N 的唯一标识符，此外，区块 N 中存储有前一区块（即区块 N-1，未在图 8 示出）的区块哈希值，例如，图 8 所示的区块 N 中的哈希 N-1。

如图 8 所示，当区块链节点为分叉区块链中的每个区块创建写缓存之后，还可以基于该块链结构中的哈希关系（比如，哈希索引关系），将各个区块的写缓存组合成一个逻辑的多层缓存，进而使得当前块的上一级缓存就是前一块的写缓存。比如，对于第一区块为图 8 所示的区块 N 而言，可以在该区块链节点中找到该区块 N 的唯一标识符（即图 8 所示的区块 N 中的哈希 N），进而可以在图 8 所示的写缓存映射表中快速找到该哈希 N 所对应的区块 N 的写缓存（例如，图 8 所示的写缓存映射表中的写缓存 N），进而可以在该写缓存 N 中对目标读取数据进行查询。可以理解的是，若该区块链节点在该区块 N 的写缓存 N（即上述第一块缓存）中查找到目标读取数据，则可以进一步执行下述 S103。

若在该写缓存 N（即第一块缓存）中未查询到该目标读取数据，则可以基于该块链结构的区块 N 中所存储的前一区块的区块哈希值（例如，图 8 所示的区块 N 的哈希 N-1），由于该哈希 N-1 可以用于唯一的标识该区块 N 的前一区块（例如，区块 N-1），所以，该区块链节点可以快速在该写缓存映射表中找到该哈希 N-1 所对应的区块 N-1 的写缓存（例如，写缓存 N-1，未在图上示出），进而可以在写缓存 N-1（即第二块缓存）中对目标读取数据进行查询。以此类推，该区块链节点可以通过该块链结构，向后遍历找到该分叉区块链中的每个区块的唯一标识符，进而可以基于每个区块的唯一标识符，对写缓存映射表中的每个区块哈希值所映射的写缓存进行遍历访问。这意味着，本申请实施例可以通过图 8 所示的块链结构的哈希关系，构建得到图 8 所示的基于哈希关系的多级缓存。其中，区块 N 的写缓存（例如，图 8 所示的写缓存 N）的逻辑查询等级会高于区块 N-1 的写缓存（例如，写缓存 N-1）的逻辑查询等级；以此类推，图 8 所示的区块 3 的写缓存（例如，图 8 所示的写缓存 3）的逻辑查询等级会高于区块 2 的写缓存（例如，图 8 所示的写缓存 2）的逻辑查询等级；同理，图 8 所示的区块 2 的写缓存（例如，图 8 所示的写缓存 2）的逻辑查询等级也会高于区块 1 的写缓存（例如，图 8 所示的写缓存 1）的逻辑查询等级。

其中，可以理解的是，一个区块可以由多个交易业务组成，每个交易业务中均可以包含需要调用的业务合约，这些业务合约中可以包含更改用户账户数据的方法以及读取用户账户数据的方法等，一个方法可以对应一个合约函数。为便于理解，本申请实施例以上述区块中的交易业务包括第一用户（例如，小红）向第二用户（例如，小明）发起的转移虚拟电子资源的目标业务为例，以阐述在节点内存中对该目标业务的目标读取数据（例如，

用户账户地址中的用户账户数据) 进行查询的具体过程。此时, 上述 S101 所获取到的与目标业务相关联的数据读取合约中的合约函数, 可以用于执行前述读取用户账户数据的方法。

请参见图 9, 图 9 是本申请实施例提供的一种调用数据读取合约进行数据读取的场景示意图。其中, 本申请实施例可以将当前执行的目标业务所属的区块统称为第一区块, 该第一区块的区块哈希值可以为图 9 所示的哈希 H1。本申请实施例可以将该第一区块的区块哈希值统称为第一区块哈希值。如图 9 所示, 当区块链节点在图 9 所示的虚拟机上运行上述数据读取合约时, 可以按照上述图 8 所维护的多级缓存, 预先在图 9 所示的写缓存映射表中获取到与哈希 H1 (即第一区块哈希值) 相关联的 KV 键值对 (例如, 键值对 1), 进而可以根据该键值对 1 快速定位到该哈希 H1 (即第一区块哈希值) 所映射块缓存 (即第一块缓存), 从而可以在该第一块缓存中获取历史写入该第一块缓存的历史业务的历史执行结果。可以理解的是, 位于该第一区块中的历史业务的交易执行顺序早于该目标业务的交易执行顺序。比如, 若第一用户 (例如, 小红) 曾经向第二用户 (例如, 小明) 进行过资产转移 (即转移过虚拟电子资源), 则可以将该第一用户调用业务合约向第二用户发起的历史资产转移记录统称为历史业务。其中, 数据读取合约中的合约函数可以包括第一执行函数和第二执行函数; 这里的第一执行函数可以用于读取第一用户的用户账户地址中的第一资产数据; 这里的第二执行函数可以用于读取第二用户的用户账户地址, 这里的第二执行函数还可以用于读取第二用户的用户账户地址中的第二资产数据, 这里将不对第二执行函数所起到的功能进行限定。

该区块链节点可以根据数据读取合约中的第一执行函数在第一块缓存的历史执行结果中对第一用户 (例如, 小红) 的用户账户地址进行查找, 并可以根据数据读取合约中的第二执行函数在第一块缓存的历史执行结果中对第二用户的用户账户地址进行查找。若该区块链节点在第一块缓存中查找到第一用户的用户账户地址, 则可以从第一用户 (例如, 小红) 的用户账户地址中读取第一用户的第一资产数据, 例如, 小红在执行完上一笔交易业务 (例如, 前述历史业务) 后所剩余的资产余额 (即可以读取第一用户在最新变动后的用户账本数据)。

可以理解的是, 为了验证该第一区块中的目标业务 (即该第一用户调用业务合约向第二用户发起的资产转移业务) 的有效性, 本申请实施例还需要根据数据读取合约中的第二执行函数执行图 9 中①所标识的步骤, 即可以在第一块缓存的历史执行结果中对第二用户的用户账户地址进行查找, 若在第一块缓存中查找到第二用户的用户账户地址, 则可以将读取到的第一资产数据和第二用户的用户账户地址, 确定为目标业务的资产读取数据, 进而可以基于资产读取数据确定与交易业务相关联的目标读取数据。

如图 9 所示, 若在图 9 所示的哈希 H1 所映射的块缓存 (即第一块缓存) 中未查找到第一用户的用户账户地址, 则区块链节点还可以执行图 9 中②所标识的步骤, 即可以基于第一块缓存所映射的第一区块哈希值 (即图 9 所示的哈希 H1), 在分叉区块链中确定第一区块的区块头, 进而可以在第一区块的区块头中, 将第一区块的父区块哈希值作为第二区块的第二区块哈希值 (例如, 图 9 所示的哈希 H2); 然后, 该区块链节点可以进一步在图

9 所示的写缓存映射关系表中获取到与哈希 H2（即第二区块哈希值）相关联的 KV 键值对（例如，键值对 2），进而可以根据该键值对 2 快速定位到该哈希 H2（即第二区块哈希值）所映射块缓存。基于此，本申请实施例可以将第二区块哈希值所映射的写缓存作为第二块缓存，以在该第二块缓存中对第一用户的用户账户地址进行查询；可以理解的是，若在第二块缓存中查找到第一用户的用户账户地址，则该区块链节点可以从第一用户的用户账户地址中读取第一用户的第一资产数据；应当理解，图 9 所示的哈希 H2 所映射的第二块缓存的逻辑查询等级低于图 9 所示的哈希 H1 所映射的第一块缓存的逻辑查询等级。

其中，可以理解的是，若在该区块链节点的节点内存中所缓存的分叉区块链为支持分叉的本地区块链，该本地区块链可以为上述图 2 所对应实施例中的分叉链 200a，此时，该第一区块可以为上述图 2 所对应实施例中的区块 11，本申请实施例可以在分叉区块链上将待写入第一区块的分支链，统称为第一分支链（也可以叫做第一备用链），并可以在该分叉区块链上除第一分支链之外的其它分支链统称为第二分支链（也可以叫做第二备用链）。所以，如图 9 所示，该第一分支链上可以包含图 9 所示的多个区块（例如，第一区块，第二区块、第三区块、第四区块）。其中，可以理解的是，本申请实施例可以将第二区块的上一区块称之为第三区块（即图 9 所示的哈希 H3 对应的区块），并可以将该第一分支链上区块生成时间戳早于该第三区块的区块生成时间戳的区块统称为第四区块。其中，图 9 所示的写缓存映射表中可以包含第三区块的第三区块哈希值和第四区块的第四区块哈希值；可以理解的是，按照上述关于多级缓存的定义，这里的第三区块哈希值所映射的第三块缓存的逻辑查询等级会低于第二块缓存的逻辑查询等级，且第四区块哈希值所映射的第四块缓存的逻辑查询等级会低于第三块缓存的逻辑查询等级。

如图 9 所示，若区块链节点在上述第二块缓存中未查找到第一用户的用户账户地址，则可以遍历执行图 9 中③所标识的步骤、④所标识的步骤以及⑤所标识的步骤，以便于可以在该第一分支链上的其他区块的块缓存中对该第一用户的用户账户地址进行遍历查询。可以理解的是，若该区块链节点在上述第二块缓存中未查找到第一用户的用户账户地址，且在图 9 所示的第三区块的块缓存（例如，第三块缓存可以为图 9 所示的哈希 H3 所映射的块缓存）和图 9 所示的第四区块的块缓存（例如，第四块缓存可以包括但不限于图 9 所示的哈希 Hn 所映射的块缓存）中均未查找到第一用户的用户账户地址，则该区块链节点可以执行图 9 中⑦所标识的步骤，以在目标区块链对应的区块链数据库中对第一用户的用户账户地址进行查询。若在区块链数据库中查找到第一用户的用户账户地址，则可以从第一用户的用户账户地址中读取第一用户的第一资产数据。

其中，可以理解的是，由于一个区块是由多笔交易业务所构成的，所以，当区块链节点对上述第一区块进行共识处理时，可以对该第一区块中的每笔交易业务进行交易验证，进而可以在上述共识网络中的区块链节点（或者共识节点）之间达到交易共识时，即可以承认当前执行的某个交易业务（例如，上述目标业务）的合法性，因此，当该第一区块中的这些交易业务均频繁调用同一业务合约发起上述资产转移业务时，则可以将该被频繁调用的业务合约作为热点合约存储至图 9 所示的链读缓存。基于此，本申请实施例还可以在

执行这些区块的交易业务中，累计某个合约调用账户（例如，上述小红所对应的合约调用账户）所频繁发起的针对于进行资产转移的合约调用请求，进而可以将累计调用频次大于调用阈值的合约调用账户作为热点调用账户，以存储在链读缓存中，以便于后续可以执行图 9 中⑦所标识的步骤之前，预先在节点内存中执行图 9 中⑥所标识的步骤，以使该区块链节点可以在节点内存的链读缓存中对上述第一用户的用户账户地址进行查询，这样，可以大大地减少对图 9 所示的数据库的访问，可以理解的是，这里的数据库可以为上述目标区块链对应的区块数据库。由此可见，本申请实施例可以在交易业务执行的过程中，对链读缓存中的热点数据进行动态更新。

S103，基于目标读取数据执行交易业务，将交易业务对应的目标执行结果作为目标缓存数据，将目标缓存数据存储至第一块缓存，将第一区块写入分叉区块链。

可以理解的是，当区块链节点在执行完上述 S103 之后，可以进一步根据获取到目标读取数据执行当前的交易业务（例如，上述目标业务），进而可以将该当前执行的交易业务的交易执行结果统称为目标执行结果。可以理解的是，本申请实施例还可以进一步将目标执行结果（例如，资产转移方在发起目标业务之后的资产余额，以及上述第一用户在交易详情页面中的交易详情数据所构成的资产转移记录等）作为目标缓存数据，以便于可以将目标缓存数据存储至该第一区块的块缓存（即第一块缓存），进而可以在执行完该第一区块中的所有交易业务之后，将该第一区块写入上述分叉区块链。

由于传统的区块链系统会将存在分叉的每个区块中的数据无差别地写入至区块链数据库，进而导致这些矿工节点所共同维护的区块链上存在需要进行版本回滚的区块，比如，若需要进行版本回滚的区块为前述区块 1B，则会导致该区块 1B 中的数据为无效数据，这意味现有的记账方式势必会造成链上的存储空间浪费。

为此，本申请实施例还可以获取与第一区块相关联的目标区块，当在节点内存中确定目标区块属于分叉区块链上的最长链时，将目标区块的目标块缓存中的数据写入目标区块链对应的区块链数据库。

通常情况下，区块链节点可以获取与第一区块相关联的最长链选取策略，进而可以基于第一区块的区块高度在分叉区块链中筛选满足最长链选取策略的区块，并将满足最长链选取策略的区块作为与第一区块相关联的目标区块；其中，目标区块为在节点内存中所存储的历史写入分叉区块链中的区块；比如，该目标区块的区块生成时间戳可以早于第一区块的区块生成时间戳。当在节点内存中确定目标区块属于分叉区块链上的最长链时，该区块链节点可以正式将目标区块写入目标区块链，且可以正式将目标区块的目标块缓存中的数据写入目标区块链对应的区块链数据库，进而可以避免链上存储资源的浪费。

可见在本申请实施例中，区块链节点还可以按照最长链选取策略在分叉区块链中获取与该第一区块相关联的目标区块（即区块生成时间戳早于第一区块的区块），从而可以在节点内存中预先判断该目标区块是否属于分叉区块链上的最长链。可以理解的是，如果区块链节点确定该目标区块属于分叉区块链上的最长链，则该区块链节点可以将目标区块所

对应的写缓存（即目标块缓存）中的数据写入目标区块链对应的区块链数据库，进而可以确保所有写入区块链数据库中的数据均属于最长链上的数据，从而可以确保录入区块链数据库中的数据的正确性，以尽可能地避免链上的存储资源的浪费。

5 其中，可以理解的是，在本申请实施例中，通过引入缓存管理机制，使得该区块链节点通过共识层所执行的区块（即第一区块）不同于上述按照最长链选取策略所筛选出的写入目标区块链的目标区块。比如，这里的第一区块可以为暂存在上述图 2 所对应实施例中的分叉链 200b 上的区块 11。另外，这里的目标区块可以为上述图 2 所对应实施例中的已成功写入目标区块链 400b 中的区块 5。

10 其中，这里的最长链选取策略具体包括第一选择策略和第二选择策略。其中，第一选择策略是指区块链节点可以对分叉区块链的各分支链上的所存在的区块数量进行比较，进而可以将具有最大区块数量的分支链作为主链。比如，区块链节点可以在分叉区块链中，将最新添加了第一区块的分支链作为上述第一分支链，进而可以统计第一分支链上的区块的第一区块数量，并可以将统计到的分叉区块链中其他分支链（即上述第二分叉区块链）上的区块数量作为第二区块数量，进而可以在第一分支链对应的第一区块数量和第二分支链对应的第二区块数量中，将具有最大区块数量的分支链作为该节点内存中的主链（即上述最长链）。这样，区块链节点可以在获取到与该第一区块相关联的目标区块（例如，该目标区块可以为与最长链上具有最大区块生成时间戳的区块（例如，上述区块 11）之间的区块距离达到预设距离阈值时的区块，例如，以预设距离阈值为 6 个区块为例，该目标区块可以为上述图 2 所对应实施例中的区块 5）时，进一步判断目标区块是否在当前确定的主链上，若该目标区块确实在确定的主链上，则该区块链节点可以允许将暂存在该节点内存中的目标区块写入目标区块链，进而可以确保该目标区块链对应的区块链数据库中可以用于存储当前写入目标区块链中的目标区块的目标块缓存中的数据。可以理解的是，此时，该目标块缓存中的数据即为该目标区块链上的链数据。

25 其中，第二选择策略是指区块链节点可以对分叉区块链的各分支链上具有最大区块生成时间戳的区块的区块高度进行比较，进而可以将具有最大区块高度的分支链作为主链。比如，当节点内存的分叉区块链上存在区块分叉现象时，区块链节点还可以按照上述第二选择策略在第一分支链中确定具有最大区块生成时间戳的区块（例如，上述图 2 所示的区块 11）的区块高度，并可以在第二分支链确定具有最大区块生成时间戳的区块（例如，上述图 2 所示的区块 4b）的区块高度，基于此，可以将该第一分支链作为初始主链，同理，由于该第一分支链中存在第一子分支链和第二子分支链，所以，该区块链节点也会将第一子分支链上的具有最大区块生成时间戳的区块（例如，图 2 所示的区块 11）的区块高度，与第二子分支链上具有最大区块生成时间戳的区块（例如，图 2 所示的区块 8a）的区块高度进行比较，进而可以将该初始主链上的区块和第一子分支链上区块所构成的哈希链统称为主链（即上述最长链），以在最长链上找到与第一区块的区块高度满足高度阈值（例如，30 6）的区块作为目标区块，以判断该目标区块是否在该最长链上。

本申请实施例通过引入写缓存映射表（即一种支持分叉的缓存管理机制），可以在对该第一区块进行共识确认的过程中，将暂未确定属于最长链的第一区块中的交易业务的交易执行结果，作为目标缓存数据暂存至第一块缓存，这样，该区块链节点可以在完成对该第一区块的共识确认时，允许将该第一区块写入节点内存中的分叉区块链（这里指能够在节点内存中支持分叉的本地区块链），从而可以在节点内存中通过存储的这些区块的哈希值构成全新的多级缓存，进而可以在这些区块的写缓存中提高数据读取的效率。此外，区块链节点还可以按照最长链选取策略在分叉区块链中获取与该第一区块相关联的目标区块（即区块生成时间戳早于第一区块的区块），从而可以在节点内存中预先判断该目标区块是否属于分叉区块链上的最长链。可以理解的是，如果区块链节点确定该目标区块属于分叉区块链上的最长链，则该区块链节点可以将目标区块所对应的写缓存（即目标块缓存）中的数据接入目标区块链对应的区块链数据库，进而可以确保所有写入区块链数据库中的数据均属于最长链上的数据，从而可以确保录入区块链数据库中的数据的正确性，以尽可能地避免链上的存储资源的浪费。

请参见图 10，图 10 是本申请实施例提供的一种基于节点内存的数据处理方法的流程示意图。如图 10 所示，方法可以由区块链节点执行，该区块链节点可以为上述图 1 所示的共识网络 100a 中的任一个区块链节点（例如，上述节点 10c）。该方法具体包括以下 S201-S208：

S201，获取第一区块中的交易业务，在节点内存的链读缓存中获取与交易业务相关联的数据读取合约。

其中，第一区块为在节点内存中待写入分叉区块链中的区块。

S202，调用数据读取合约获取与第一区块相关联的写缓存映射表，基于第一区块在写缓存映射表中的第一块缓存，确定与交易业务相关联的目标读取数据。

S203，基于目标读取数据执行交易业务，将交易业务对应的目标执行结果作为目标缓存数据，将目标缓存数据存储至第一块缓存，将第一区块写入分叉区块链。

S204，获取与第一区块相关联的目标区块，当在节点内存中确定目标区块属于分叉区块链上的最长链时，将目标区块的目标块缓存中的数据写入目标区块链对应的区块链数据库。

其中，目标区块的区块生成时间戳早于第一区块的区块生成时间戳。区块链节点可以获取与第一区块相关联的最长链选取策略，基于第一区块的区块高度在分叉区块链中筛选满足最长链选取策略的区块，进而可以将满足最长链选取策略的区块作为与第一区块相关联的目标区块；目标区块为在节点内存中所缓存的历史写入分叉区块链中的区块；当在节点内存中确定目标区块属于分叉区块链上的最长链时，区块链节点可以将目标区块写入目标区块链，且将目标区块的目标块缓存中的数据写入目标区块链对应的区块链数据库。

其中，链读缓存中可以包含：上述热点数据和历史链数据；这里的历史链数据可以包括在历史时刻成功写入目标区块链的历史块缓存中的数据；历史块缓存为分叉区块链中的历史区块的写缓存，且历史区块的区块生成时间戳早于目标区块的区块生成时间戳，所以，当区块链节点执行完上述 S204 之后，还可以执行下述 S205。

- 5 S205，将目标块缓存中的数据作为目标区块链上的目标链数据，将目标链数据写入链读缓存，在链读缓存中用目标链数据更新历史链数据。

例如，区块链节点可以将目标块缓存中的数据作为目标区块链上的目标链数据，进而可以在当前时刻为 T2 时刻（即该 T2 时刻为上述 T1 时刻之后的时刻）时将目标链数据写入链读缓存，可以理解的是，由于该链读缓存中包含在上述历史时刻（例如，T1 时刻之前的时刻）写入链读缓存的其他块缓存（即历史块缓存）中的数据（即上述历史链数据）。所以，当将该目标链数据写入该链读缓存时，可以在该链读缓存中动态更新历史链数据（即可以将该目标链数据作为新的历史链数据），以便于后续在打包生成新的第一区块的时候，可以在链读缓存中为最新生成的新的第一区块提供快速的历史链数据的访问。又比如，该链读缓存中还可以用于存储该区块链节点在执行某个区块的这些交易业务的过程中，经常调用的业务合约（即上述热点合约），以及经常调用发起合约调用请求的热点调用账户等热点数据。

为便于理解，请参见图 11，图 11 是本申请实施例提供的一种提交目标区块的场景示意图。如图 11 所示，区块链节点通过图 11 所示的共识层执行步骤 1 时，可以对当前打包生成的区块（即第一区块，该第一区块可以为图 11 所示的）进行共识，进而可以对该第一区块中的交易业务（简称为交易）进行交易验证。此时，该区块链节点通过图 11 所示的执行层执行步骤 2 时，执行该第一区块中的各交易，进而可以在各交易验证通过时将各自的交易执行结果写入第一块缓存，以维护该节点内存的第一块缓存中的缓存数据。可以理解的是，当区块链节点完成对该第一区块中的交易业务的交易共识之后，可以完成对该第一区块的共识，进而可以将该第一区块写入到节点内存的分叉区块链（即本地支持分叉的分叉链）上，以得到上述图 2 所示的分叉链 200a。

可以理解的是，该区块链节点在执行第一区块的交易业务的过程中，可以执行图 11 所示的步骤 3，即该区块链节点可以获取与该交易业务相关联的热点读取辅助参数，并可以将包含该热点读取辅助参数的业务处理指令发送给图 11 所示的账本层，以从该账本层的链读缓存中基于热点读取辅助参数中的合约调用地址，获取用于执行交易业务的业务合约，进而可以将链读缓存中所获取到的业务合约统称为上述数据读取合约。如图 11 所示，如果该区块链节点未在该链读缓存中获取到与该交易业务相关联的数据读取合约，则可以执行图 11 所示的步骤 4，进而可以从与该区块链节点相关联的数据库（例如，图 11 所示的目标区块链对应的主数据库）中，获取到与该交易业务相关联的数据读取合约，以便于可以通过获取到的数据读取合约读取用于验证该交易业务的目标读取数据，进而可以基于该目标读取数据执行该交易业务，从而可以将该交易业务的交易执行结果写入至该第一区块对应的第一块缓存，以在图 11 所示的写缓存映射表中动态维护该第一块缓存中的缓存数据。

其中，可以理解的是，由于一个区块中可以包含大量的交易业务，这些交易业务的执行可以按照交易执行顺序进行执行。所以，该区块链节点在执行第一区块中的这些交易业务的过程中，还可以累计执行这些交易业务的过程中所频繁调用的业务合约或者频繁发起合约调用的合约调用账户，进而可以将这些频繁调用的业务合约和合约调用账户作为热点数据，以在图 11 所示的账本层中动态更新链读缓存中的热点数据。

可以理解的是，当图 11 所示的共识层将第一区块（例如，图 11 所示的区块高度为 10 的区块）写入上述分叉区块链，得到上述分叉链 200a 时，该区块链节点中的共识层还可以基于上述最长链选取策略，在该分叉链 200a 上选取历史执行过的满足最长链选取策略的区块，作为与第一区块相关联的目标区块，进而可以执行图 11 所示的步骤 a，即可以在将该目标区块提交至目标区块链的同时，进一步执行图 11 所示的步骤 c，即将目标区块所对应的目标块缓存中的数据写入图 11 所示的主数据库。

此外，可以理解的是，该区块链节点还可以将该目标区块的目标块缓存中的数据写入图 11 所示的账本层，以便于对该链读缓存中的历史链数据进行动态更新。

由此可见，采用本申请实施例，可以提供一种支持分叉的缓存管理机制，进而可以在执行完当前打包生成的区块（即第一区块）时，进一步在节点内存中完成对支持分叉的分叉区块链上的最长链的选择，从而可以确定出当前能够提交至目标区块链的区块（即目标区块）。然后，区块链节点可以在将当前提交的目标区块的目标块缓存中的数据作为目标链数据，写入图 11 所示的链读缓存之后，还可以执行图 11 所示的步骤 b，即可以获取目标区块的目标块缓存，并可以进一步执行图 11 所示的步骤 d，即可以在节点内存中对删除该目标区块的目标块缓存中的数据。

其中，可以理解的是，如图 11 所示，该区块链节点可以在节点内存中维护的一张哈希关系表，进而可以通过该哈希关系表来确定当前提交的目标区块的区块高度，进而可以在该哈希关系表中找出与该目标区块具有同一区块高度的所有区块（即所有的分叉区块）。换言之，可以理解的是，若该分叉区块链上存在与目标区块具有相同区块高度的分叉区块，则该区块链节点可以进一步执行下述 S206。

S206，在分叉区块链中确定目标区块的目标区块高度，从节点内存中获取哈希关系表，在哈希关系表中确定目标区块高度所在的目标列表栏。

S207，从目标列表栏中获取分叉区块的分叉区块哈希值，在写缓存映射关系表中确定分叉区块哈希值对应的分叉块缓存。

S208，清除目标块缓存中的目标链数据，且清除分叉块缓存中的数据。

为便于理解，请参见图 12，图 12 是本申请实施例提供的一种进行数据删除的场景示意图。如图 12 所示，考虑到节点内存中的分叉区块链（即图 12 所示的分叉链）可以具备区块分叉的特性，所以，区块链节点可以在节点内存的分叉链支持分叉特性的同时，维护一张哈希映射表（即图 12 所示的哈希关系表）。比如，如图 12 所示的哈希映射表是由分

叉链中的所有区块按照区块高度和区块哈希值所构成的。在该哈希映射表中，具有同一区块高度的区块的数量可以是一个，也可以是多个。比如，在图 12 所示的分叉链中，该区块链节点可以获取与第一区块相关联的目标区块（例如，可以基于上述最长链选取策略将图 12 所示的与第一区块的区块高度差为 6 的区块作为目标区块），进而可以在确定图 12 所示的区块 4a（即目标区块）属于该分叉区块链的最长链时，将目标区块（例如，图 12 所示的区块 4a）的目标块缓存中的数据写入目标区块链对应的区块链数据库，该区块链数据库可以为上述图 11 所示的主数据库。

此外，本申请实施例在将目标块缓存中的数据写入目标区块链对应的区块链数据库之后，为了提高数据读取的效率，还可以将该目标块缓存中的数据作为目标链数据，写入链读缓存，这样，就可以直接在节点内存中快速查询到上述目标读取数据，而无需通过网络查询前述区块链数据库。

最后，由于该节点内存的载体容量是有限的，所以，本申请实施例还可以在节点内存中删除该目标块缓存中的目标链数据。具体的，区块链节点可以在图 12 所示的分叉链中确定当前提交的目标区块的区块位置，即目标区块的区块位置可以为该目标区块的目标区块高度。比如，图 12 所示的区块 4a（即目标区块）的目标区块高度可以为图 12 所示的哈希关系表中的“4”，进而可以在该哈希关系表中将目标区块高度为“4”的列表栏作为目标列表栏。如图 12 所示，该区块链节点可以在该目标列表栏确定与目标区块高度相关联的区块哈希值。比如，与目标区块高度为“4”的区块哈希值具体可以包括图 12 的区块 4a 的区块哈希值（例如，哈希 4a）和区块 4b 的区块哈希值（例如，区块 4b）。可以理解的是，这里的区块 4a 即为在分叉区块链（即图 12 所示的分叉链）上所存在的与目标区块具有相同区块高度的分叉区块。所以，由于分叉区块中的数据并不会记录到区块链数据库，所以，为了避免节点内存中的内存资源的浪费，该区块链节点还可以在目标列表栏中获取分叉区块的分叉区块哈希值（即图 12 所示的哈希 4b），这样，该区块链节点可以快速在写缓存映射关系表中确定分叉区块哈希值对应的分叉块缓存。所以，当区块链节点需要在上述写缓存映射表中删除目标列表栏中的区块哈希值所对应的写缓存时，可以在清除目标块缓存（即区块 4a 对应的写缓存）中的目标链数据的同时，一并清除分叉块缓存（即区块 4b 对应的写缓存）中的数据，进而可以改善节点内存的有效利用率。

可以理解的是，在本申请实施例中，当在节点内存中确定第一区块属于分叉区块链上的最长链时，区块链节点还可以将第一块缓存中的目标缓存数据写入目标区块链对应的区块链数据库。

比如，具体的，若上述第一节点中的交易业务的数量为多个，比如，该第一节点中的任意两个具有相邻关系的交易业务可以包括第一交易业务和第二交易业务，则上述目标缓存数据可以包括第一交易业务对应的第一缓存数据和第二交易业务对应的第二缓存数据；其中，第二交易业务可以为第一交易业务的上一业务，比如，这里的第一交易业务可以为上述目标业务，这里的第二交易业务可以为上述历史业务；此时，区块链节点可以在将第

一块缓存中的第一缓存数据写入目标区块链对应的区块链数据库的同时，一并将第一块缓存中的第二缓存数据写入区块链数据库。

换言之，在本申请实施例中，区块链节点还可以基于上述最长链选取策略，在后续确定上述第一区块也属于分叉区块链上的最长链时，允许将该第一区块的第一块缓存中的所有数据（即前述第一缓存数据和第二缓存数据）均写入上述区块链数据库。

可以理解的是，由于区块链节点的本地内存的内存资源有限，为确保区块链节点可以提供适应的内存计算资源，本申请实施例还可以动态地对节点内存的缓存容量进行监测，进而可以在监测到节点内存的缓存容量达到缓存阈值时，在分叉区块链中将满足缓存时长条件的区块（即在分叉区块链中所缓存的一些生成时间戳比较早的区块）确定为待销毁区块；这样，该区块链节点可以在分叉区块链中对待销毁区块进行区块销毁处理，以有效地管理和利用节点内存的内存资源。

由此可见，本申请实施例通过提供的一种支持分叉的缓存管理机制，可以将当前执行的某个区块（即第一区块）的交易业务的交易执行结果写入第一块缓存之后，可以将该第一区块暂存在节点内存的分叉区块链上，比如，本申请实施例可以根据该第一区块的区块头中的父区块哈希值，在分叉区块链上定位到上一区块（例如，上述第二区块）的区块哈希值，进而可以将该第一区块添加至该分叉区块链上的第二区块所在的分支链。应当理解，这里的分叉区块链可以为在节点内存中支持分叉特性的本地区块链。基于此，本申请实施例还可以在节点内存中实时动态地维护一张将区块高度与区块哈希值进行关联的哈希关系表，通过该哈希关系表可以快速找出与当前提交的目标区块具有相同区块高度的其他区块，并可以在节点内存中将找出的与当前提交的目标区块具有相同区块高度的其他区块统称为分叉区块。这样，区块链节点在节点内存中清除上链后的目标区块的写缓存（即上述目标区块缓存）中的数据时，还可以一并在节点内存中清除所有的分叉区块的写缓存（即上述分叉区块缓存）中的数据，进而可以确保最终能够写入目标区块链的数据均属于最长链的数据，进而可以避免链上的存储资源的浪费。显然，在本申请实施例中，通过引入上述支持分叉的缓存管理机制，可以有效地利用节点内存的内存资源，进而可以减轻对区块链数据库的访问压力，从而可以在节点内存中提高数据的访问的效率。此外，应当理解，本申请实施例所涉及的目标区块链上不会存在区块的分叉现象，这意味着该区块链节点无需对目标区块链进行版本回滚，进而无需在目标区块链上维护复杂的区块链分叉状态。

请参见图 13，图 13 是本申请提供的一种基于节点内存的数据处理装置的结构示意图。如图 13 所示，基于节点内存的数据处理装置 1 可应用于上述共识网络中任意一个区块链节点，例如，数据处理装置 1 可应用于上述图 1 所对应实施例中的节点 10c。如图 13 所示，基于节点内存的数据处理装置 1 可以包括：业务获取模块 11、合约调用模块 12 和业务执行模块 13；还可以包括最长链确定模块 14；基于节点内存的数据处理装置 1 还可以包括：链读数据存储模块 15，哈希表获取模块 16，分叉缓存确定模块 17，数据清除模块 18，缓存容量监测模块 19，销毁处理模块 20；

业务获取模块 11, 用于获取第一区块中的交易业务, 在节点内存的链读缓存中获取与交易业务相关联的数据读取合约; 第一区块为在节点内存中待写入分叉区块链中的区块;

业务获取模块 11 包括: 区块共识单元 111, 热点合约查找单元 112 和合约确定单元 113;

5 区块共识单元 111, 用于获取包含交易业务的第一区块, 在对第一区块进行共识处理时, 生成用于对第一区块中的交易业务进行验证的业务处理指令; 业务处理指令中包含与交易业务相关联的热点读取辅助参数; 热点读取辅助参数包括: 用于执行交易业务的业务合约的合约调用地址;

10 热点合约查找单元 112, 用于从业务处理指令中提取合约调用地址, 基于合约调用地址从节点内存的链读缓存中查找与合约调用地址所映射的合约名称相匹配的热点合约; 热点合约为在链读缓存中所确定的累计访问频次大于访问阈值的业务合约;

合约确定单元 113, 用于将查找到的与合约名称相匹配的热点合约作为与交易业务相关联的数据读取合约。

其中, 区块共识单元 111, 热点合约查找单元 112 和合约确定单元 113 的具体实现方式, 可以参见上述图 3 所对应实施例中对 S101 的描述, 这里将不再继续进行赘述。

15 其中, 业务获取模块 11, 还用于若在链读缓存中未获取到与交易业务相关联的数据读取合约, 则从与节点内存相关联的区块链数据库中获取数据读取合约。

合约调用模块 12, 用于调用数据读取合约获取与第一区块相关联的写缓存映射表, 基于第一区块在写缓存映射表中的第一块缓存, 确定与交易业务相关联的目标读取数据;

20 其中, 交易业务包括: 第一用户向第二用户发起的用于进行资产转移的目标业务; 数据读取合约中的合约函数包括第一执行函数和第二执行函数; 第一执行函数用于读取第一用户的用户账户地址中的第一资产数据; 第二执行函数用于读取第二用户的用户账户地址;

合约调用模块 12 包括: 合约运行单元 121, 第一缓存查找单元 122, 和第一确定单元 123; 合约调用模块 12 还包括: 哈希确定单元 124, 第二缓存确定单元 125, 第二查找单元 126, 第二确定单元 127, 数据库查找单元 128 和数据读取单元 129;

25 合约运行单元 121, 用于当在虚拟机中运行数据读取合约时, 获取与第一区块相关联的写缓存映射表, 在写缓存映射表中, 将第一区块的第一区块哈希值所映射的写缓存作为第一块缓存, 获取第一块缓存中所存储的历史业务的历史执行结果; 历史业务的交易执行顺序早于目标业务的交易执行顺序;

30 第一缓存查找单元 122, 用于根据数据读取合约中的第一执行函数在第一块缓存的历史执行结果中对第一用户的用户账户地址进行查找, 根据数据读取合约中的第二执行函数在第一块缓存的历史执行结果中对第二用户的用户账户地址进行查找;

第一确定单元 123, 用于若在第一块缓存中查找到第一用户的用户账户地址, 则从第一用户的用户账户地址中读取第一用户的第一资产数据;

- 5 第一确定单元 123, 还用于若在第一块缓存中查找到第二用户的用户账户地址, 则将读取到的第一资产数据和第二用户的用户账户地址, 确定为目标业务的资产读取数据, 基于资产读取数据确定与交易业务相关联的目标读取数据。

其中, 合约运行单元 121, 第一缓存查找单元 122, 第一确定单元 123 的具体实现方式, 可以参见上述图 3 所对应实施例中对 S102 的描述, 这里将不再继续进行赘述。

其中, 分叉区块链中包含第二区块; 第二区块为第一区块的上一区块; 写缓存映射表中包含第二区块的第二区块哈希值;

- 10 哈希确定单元 124, 用于若在第一块缓存中未查找到第一用户的用户账户地址, 则基于第一块缓存所映射的第一区块哈希值, 在分叉区块链中确定第一区块的区块头, 在第一区块的区块头中, 将第一区块的父区块哈希值作为第二区块的第二区块哈希值;

第二缓存确定单元 125, 用于在写缓存映射关系表中将第二区块哈希值所映射的写缓存作为第二块缓存; 第二块缓存的逻辑查询等级低于第一块缓存的逻辑查询等级;

- 15 第二查找单元 126, 用于在第二块缓存中对第一用户的用户账户地址进行查询;

第二确定单元 127, 用于若在第二块缓存中查询到第一用户的用户账户地址, 则从第一用户的用户账户地址中读取第一用户的第一资产数据。

- 20 其中, 分叉区块链包含第一分支链; 若第一分支链中包含第三区块和第四区块; 第三区块为第二区块的上一区块, 且第四区块的区块生成时间戳早于第三区块的区块生成时间戳; 写缓存映射表中包含第三区块的第三区块哈希值和第四区块的第四区块哈希值; 第三区块哈希值所映射的第三块缓存的逻辑查询等级低于第二块缓存的逻辑查询等级, 且第四区块哈希值所映射的第四块缓存的逻辑查询等级低于第三块缓存的逻辑查询等级;

- 25 数据库查找单元 128, 用于若在第二块缓存中未查找到第一用户的用户账户地址, 且在第三块缓存和第四块缓存中均未查找到第一用户的用户账户地址, 则在目标区块链对应的区块链数据库中对第一用户的用户账户地址进行查询;

数据读取单元 129, 用于若在区块链数据库中查询到第一用户的用户账户地址, 则从第一用户的用户账户地址中读取第一用户的第一资产数据。

- 30 其中, 哈希确定单元 124, 第二缓存确定单元 125, 第二查找单元 126, 第二确定单元 127, 数据库查找单元 128 和数据读取单元 129 的具体实现方式, 可以参见上述图 3 所对应实施例中对基于哈希关系进行多级缓存的查询的具体过程的描述, 这里将不再继续进行赘述。

业务执行模块 13, 用于基于目标读取数据执行交易业务, 将交易业务对应的目标执行结果作为目标缓存数据, 将目标缓存数据存储至第一块缓存, 将第一区块写入分叉区块链。

基于节点内存的数据处理装置 1 还可以包括最长链确定模块 14:

5 最长链确定模块 14, 用于获取与第一区块相关联的目标区块, 当在节点内存中确定目标区块属于分叉区块链上的最长链时, 将目标区块的目标块缓存中的数据写入目标区块链对应的区块链数据库; 目标区块的区块生成时间戳大于第一区块的区块生成时间戳。

其中, 最长链确定模块 14 包括: 区块筛选单元 141 和数据写入单元 142;

10 区块筛选单元 141, 用于获取与第一区块相关联的最长链选取策略, 基于第一区块的区块高度在分叉区块链中筛选满足最长链选取策略的区块, 将满足最长链选取策略的区块作为与第一区块相关联的目标区块; 目标区块为在节点内存中所缓存的历史写入分叉区块链中的区块;

数据写入单元 142, 用于当在节点内存中确定目标区块属于分叉区块链上的最长链时, 将目标区块写入目标区块链, 且将目标区块的目标块缓存中的数据写入目标区块链对应的区块链数据库。

15 其中, 区块筛选单元 141 和数据写入单元 142 的具体实现方式, 可以参见上述图 3 所对应实施例中对 S104 的描述, 这里将不再继续进行赘述。

其中, 最长链确定模块 14, 还用于当在节点内存中确定第一区块属于分叉区块链上的最长链时, 将第一块缓存中的目标缓存数据写入目标区块链对应的区块链数据库。

20 具体的, 其中, 若交易业务包括第一交易业务和第二交易业务, 则目标缓存数据包括第一交易业务对应的第一缓存数据和第二交易业务对应的第二缓存数据; 第二交易业务为第一交易业务的上一业务;

最长链确定模块 14, 具体用于将第一块缓存中的第一缓存数据写入目标区块链对应的区块链数据库, 且将第一块缓存中的第二缓存数据写入区块链数据库。

25 其中, 链读缓存中包含: 历史链数据; 历史链数据包括在历史时刻成功写入目标区块链的历史块缓存中的数据; 历史块缓存为分叉区块链中的历史区块的写缓存, 且历史区块的区块生成时间戳早于目标区块的区块生成时间戳;

链读数据存储模块 15, 用于将目标块缓存中的数据作为目标区块链上的目标链数据, 将目标链数据写入链读缓存, 在链读缓存中用目标链数据更新历史链数据。

其中, 分叉区块链上存在与目标区块具有相同区块高度的分叉区块;

30 哈希表获取模块 16, 用于在分叉区块链中确定目标区块的目标区块高度, 从节点内存中获取哈希关系表, 在哈希关系表中确定目标区块高度所在的目标列表栏;

分叉缓存确定模块 17, 用于从目标列表栏中获取分叉区块的分叉区块哈希值, 在写缓存映射关系表中确定分叉区块哈希值对应的分叉块缓存;

数据清除模块 18, 用于清除目标块缓存中的目标链数据, 且清除分叉块缓存中的数据。

缓存容量监测模块 19, 用于在监测到节点内存的缓存容量达到缓存阈值时, 在分叉区块链中将满足缓存时长条件的区块确定为待销毁区块;

销毁处理模块 20, 用于在分叉区块链中对待销毁区块进行区块销毁处理。

其中, 业务获取模块 11、合约调用模块 12、业务执行模块 13 和最长链确定模块 14 的具体实现方式, 可以参见上述图 3 所对应实施例中对 S101-S104 的描述, 这里将不再继续进行赘述。链读数据存储模块 15, 哈希表获取模块 16, 分叉缓存确定模块 17, 数据清除模块 18, 缓存容量监测模块 19, 销毁处理模块 20 的具体实现方式, 可以参见上述图 10 所对应实施例中对 S201-S208 的描述, 这里将不再继续进行赘述。可以理解的是, 对采用相同方法的有益效果描述, 也不再进行赘述。

请参见图 14, 图 14 是本申请实施例提供的一种计算机设备的结构示意图。如图 14 所示, 计算机设备 1000 可以应用于上述图 1 对应实施例中的区块链节点, 计算机设备 1000 可以包括: 处理器 1001, 网络接口 1004 和存储器 1005, 此外, 计算机设备 1000 还可以包括: 用户接口 1003, 和至少一个通信总线 1002。其中, 通信总线 1002 用于实现这些组件之间的连接通信。其中, 用户接口 1003 可以包括显示屏 (Display)、键盘 (Keyboard), 可选用户接口 1003 还可以包括标准的有线接口、无线接口。网络接口 1004 可选的可以包括标准的有线接口、无线接口 (如 WI-FI 接口)。存储器 1004 可以是高速 RAM 存储器, 也可以是非不稳定的存储器 (non-volatile memory), 例如至少一个磁盘存储器。存储器 1005 可选的还可以是至少一个位于远离前述处理器 1001 的存储装置。如图 14 所示, 作为一种计算机存储介质的存储器 1005 中可以包括操作系统、网络通信模块、用户接口模块以及设备控制应用程序。

该计算机设备 1000 中的网络接口 1004 还可以提供网络通讯功能, 且可选用户接口 1003 还可以包括显示屏 (Display)、键盘 (Keyboard)。在图 14 所示的计算机设备 1000 中, 网络接口 1004 可提供网络通讯功能; 而用户接口 1003 主要用于为用户提供输入的接口; 而处理器 1001 可以用于调用存储器 1005 中存储的设备控制应用程序, 以实现:

获取第一区块中的交易业务, 在节点内存的链读缓存中获取与交易业务相关联的数据读取合约; 第一区块为在节点内存中待写入分叉区块链中的区块;

调用数据读取合约获取与第一区块相关联的写缓存映射表, 基于第一区块在写缓存映射表中的第一块缓存, 确定与交易业务相关联的目标读取数据;

基于目标读取数据执行交易业务, 将交易业务对应的目标执行结果作为目标缓存数据, 将目标缓存数据存储至第一块缓存, 将第一区块写入分叉区块链。

应当理解，本申请实施例中所描述的计算机设备 1000 可执行前文图 3 或图 10 所对应实施例中对基于节点内存的数据处理方法的描述，也可执行前文图 13 所对应实施例中对基于节点内存的数据处理装置 1 的描述，在此不再赘述。另外，对采用相同方法的有益效果描述，也不再进行赘述。

- 5 此外，这里需要指出的是：本申请实施例还提供了一种计算机存储介质，且计算机存储介质中存储有前文提及的基于节点内存的数据处理装置 1 所执行的计算机程序，且计算机程序包括程序指令，当处理器执行程序指令时，能够执行前文图 3 或图 10 所对应实施例中对基于节点内存的数据处理方法的描述，因此，这里将不再进行赘述。另外，对采用相同方法的有益效果描述，也不再进行赘述。对于本申请所涉及的计算机存储介质实施例中未披露的技术细节，请参照本申请方法实施例的描述。
- 10

- 本领域普通技术人员可以理解实现上述实施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计算机可读取存储介质中，该程序在执行时，可包括如上述各方法的实施例的流程。其中，所述的存储介质可为磁碟、光盘、只读存储记忆体（Read-Only Memory, ROM）或随机存储记忆体（Random Access
- 15 Memory, RAM）等。

以上所揭露的仅为本申请较佳实施例而已，当然不能以此来限定本申请之权利范围，因此依本申请权利要求所作的等同变化，仍属本申请所涵盖的范围。

权 利 要 求

1、一种基于节点内存的数据处理方法，所述方法由区块链节点执行，包括：

获取第一区块中的交易业务，在所述节点内存的链读缓存中获取与所述交易业务相关的数据读取合约；所述第一区块为在所述节点内存中待写入分叉区块链中的区块；

5 调用所述数据读取合约获取与所述第一区块相关联的写缓存映射表，基于所述第一区块在所述写缓存映射表中的第一块缓存，确定与所述交易业务相关联的目标读取数据；

基于所述目标读取数据执行所述交易业务，将所述交易业务对应的目标执行结果作为目标缓存数据，将所述目标缓存数据存储至所述第一块缓存，将所述第一区块写入所述分叉区块链。

10 2、根据权利要求1所述的方法，所述链读缓存是在所述节点内存的缓存空间中为所述目标区块链创建的，且所述链读缓存用于暂存被频繁访问的热点数据；所述热点数据包括所述数据读取合约。

3、根据权利要求1所述的方法，所述方法还包括：

15 获取与所述第一区块相关联的目标区块，当在所述节点内存中确定所述目标区块属于所述分叉区块链上的最长链时，将所述目标区块的目标块缓存中的数据写入目标区块链对应的区块链数据库；所述目标区块的区块生成时间戳早于所述第一区块的区块生成时间戳。

4、根据权利要求1所述的方法，所述获取第一区块中的交易业务，在所述节点内存的链读缓存中获取与所述交易业务相关的数据读取合约，包括：

20 获取包含所述交易业务的第一区块，在对所述第一区块进行共识处理时，生成用于对所述第一区块中的所述交易业务进行验证的业务处理指令；所述业务处理指令中包含与所述交易业务相关联的热点读取辅助参数；所述热点读取辅助参数包括：用于执行所述交易业务的业务合约的合约调用地址；

25 从所述业务处理指令中提取所述合约调用地址，基于所述合约调用地址从所述节点内存的链读缓存中查找与所述合约调用地址所映射的合约名称相匹配的热点合约；所述热点合约为在所述链读缓存中所确定的累计访问频次大于访问阈值的业务合约；

将查找到的与所述合约名称相匹配的热点合约作为与所述交易业务相关的数据读取合约。

30 5、根据权利要求1所述的方法，所述交易业务包括：第一用户向第二用户发起的用于进行资产转移的目标业务；所述数据读取合约中的合约函数包括第一执行函数和第二执行函数；所述第一执行函数用于读取所述第一用户的用户账户地址中的第一资产数据；所述第二执行函数用于读取所述第二用户的用户账户地址；

所述调用所述数据读取合约获取与所述第一区块相关联的写缓存映射表，基于所述第一区块在所述写缓存映射表中的第一块缓存，确定与所述交易业务相关联的目标读取数据，包括：

5 当在虚拟机中运行所述数据读取合约时，获取与所述第一区块相关联的写缓存映射表，在所述写缓存映射表中，将所述第一区块的第一区块哈希值所映射的写缓存作为第一块缓存，获取所述第一块缓存中所存储的历史业务的历史执行结果；所述历史业务的交易执行顺序早于所述目标业务的交易执行顺序；

10 根据所述数据读取合约中的所述第一执行函数在所述第一块缓存的历史执行结果中对所述第一用户的用户账户地址进行查找，根据所述数据读取合约中的所述第二执行函数在所述第一块缓存的历史执行结果中对所述第二用户的用户账户地址进行查找；

若在所述第一块缓存中查找到所述第一用户的用户账户地址，则从所述第一用户的用户账户地址中读取所述第一用户的所述第一资产数据；

15 若在所述第一块缓存中查找到所述第二用户的用户账户地址，则将读取到的所述第一资产数据和所述第二用户的用户账户地址，确定为所述目标业务的资产读取数据，基于所述资产读取数据确定与所述交易业务相关联的目标读取数据。

6、根据权利要求5所述的方法，所述分叉区块链中包含第二区块；所述第二区块为所述第一区块的上一区块；所述写缓存映射表中包含所述第二区块的第二区块哈希值；

所述方法还包括：

20 若在所述第一块缓存中未查找到所述第一用户的用户账户地址，则基于所述第一块缓存所映射的所述第一区块哈希值，在所述分叉区块链中确定所述第一区块的区块头，在所述第一区块的区块头中，将所述第一区块的父区块哈希值作为所述第二区块的所述第二区块哈希值；

在所述写缓存映射关系表中将所述第二区块哈希值所映射的写缓存作为第二块缓存；所述第二块缓存的逻辑查询等级低于所述第一块缓存的逻辑查询等级；

25 在所述第二块缓存中对所述第一用户的用户账户地址进行查询；

若在所述第二块缓存中查询到所述第一用户的用户账户地址，则从所述第一用户的用户账户地址中读取所述第一用户的所述第一资产数据。

30 7、根据权利要求6所述的方法，所述分叉区块链包含第一分支链；若所述第一分支链中包含第三区块和第四区块；所述第三区块为所述第二区块的上一区块，且所述第四区块的区块生成时间戳早于所述第三区块的区块生成时间戳；所述写缓存映射表中包含所述第三区块的第三区块哈希值和所述第四区块的第四区块哈希值；所述第三区块哈希值所映射的第三块缓存的逻辑查询等级低于所述第二块缓存的逻辑查询等级，且所述第四区块哈希值所映射的第四块缓存的逻辑查询等级低于所述第三块缓存的逻辑查询等级；

所述方法还包括：

若在所述第二块缓存中未查找到所述第一用户的用户账户地址，且在所述第三块缓存和所述第四块缓存中均未查找到所述第一用户的用户账户地址，则在目标区块链对应的区块链数据库中对所述第一用户的用户账户地址进行查询；

- 5 若在所述区块链数据库中查询到所述第一用户的用户账户地址，则从所述第一用户的用户账户地址中读取所述第一用户的所述第一资产数据。

8、根据权利要求3所述的方法，所述获取与所述第一区块相关联的目标区块，当在所述节点内存中确定所述目标区块属于所述分叉区块链上的最长链时，将所述目标区块的目标块缓存中的数据写入目标区块链对应的区块链数据库，包括：

- 10 获取与所述第一区块相关联的最长链选取策略，基于所述第一区块的区块高度在所述分叉区块链中筛选满足所述最长链选取策略的区块，将满足所述最长链选取策略的区块作为与所述第一区块相关联的目标区块；所述目标区块为在所述节点内存中所缓存的历史写入所述分叉区块链中的区块；

- 15 当在所述节点内存中确定所述目标区块属于所述分叉区块链上的最长链时，将所述目标区块写入目标区块链，并将所述目标区块的目标块缓存中的数据写入所述目标区块链对应的区块链数据库。

- 20 9、根据权利要求8所述的方法，所述链读缓存中包含：历史链数据；所述历史链数据包括在历史时刻成功写入所述目标区块链的历史块缓存中的数据；所述历史块缓存为所述分叉区块链中的历史区块的写缓存，且所述历史区块的区块生成时间戳早于所述目标区块的区块生成时间戳；

所述方法还包括：

将所述目标块缓存中的数据作为所述目标区块链上的目标链数据，将所述目标链数据写入所述链读缓存，在所述链读缓存中用所述目标链数据更新所述历史链数据。

- 25 10、根据权利要求9所述的方法，所述分叉区块链上存在与所述目标区块具有相同区块高度的分叉区块；

所述方法还包括：

在所述分叉区块链中确定所述目标区块的目标区块高度，从所述节点内存中获取哈希关系表，在所述哈希关系表中确定所述目标区块高度所在的目标列表栏；

- 30 从所述目标列表栏中获取所述分叉区块的分叉区块哈希值，在所述写缓存映射关系表中确定所述分叉区块哈希值对应的分叉块缓存；

清除所述目标块缓存中的所述目标链数据，且清除所述分叉块缓存中的数据。

11、根据权利要求 1 所述的方法，所述方法还包括：

若在链读缓存中未获取到与所述交易业务相关联的数据读取合约，则从与所述节点内存相关联的区块链数据库中获取所述数据读取合约。

12、根据权利要求 3 所述的方法，所述方法还包括：

5 当在所述节点内存中确定所述第一区块属于所述分叉区块链上的最长链时，将所述第一块缓存中的所述目标缓存数据写入所述目标区块链对应的所述区块链数据库。

13、根据权利要求 12 所述的方法，若所述交易业务包括第一交易业务和第二交易业务，则所述目标缓存数据包括所述第一交易业务对应的第一缓存数据和所述第二交易业务对应的第二缓存数据；所述第二交易业务为所述第一交易业务的上一业务；

10 所述将所述第一块缓存中的所述目标缓存数据写入所述目标区块链对应的所述区块链数据库，包括：

将所述第一块缓存中的所述第一缓存数据写入所述目标区块链对应的所述区块链数据库，且将所述第一块缓存中的所述第二缓存数据写入所述区块链数据库。

14、根据权利要求 1 所述的方法，所述方法还包括：

15 在监测到所述节点内存的缓存容量达到缓存阈值时，在所述分叉区块链中将满足缓存时长条件的区块确定为待销毁区块；

在所述分叉区块链中对所述待销毁区块进行区块销毁处理。

15、一种基于节点内存的数据处理装置，所述装置部署在区块链节点上，包括：

20 业务获取模块，用于获取第一区块中的交易业务，在所述节点内存的链读缓存中获取与所述交易业务相关联的数据读取合约；所述第一区块为在所述节点内存中待写入分叉区块链中的区块；

合约调用模块，用于调用所述数据读取合约获取与所述第一区块相关联的写缓存映射表，基于所述第一区块在所述写缓存映射表中的第一块缓存，确定与所述交易业务相关联的目标读取数据；

25 业务执行模块，用于基于所述目标读取数据执行所述交易业务，将所述交易业务对应的目标执行结果作为目标缓存数据，将所述目标缓存数据存储至所述第一块缓存，将所述第一区块写入所述分叉区块链。

16、根据权利要求 15 所述的装置，所述装置还包括：

30 最长链确定模块，用于获取与所述第一区块相关联的目标区块，当在所述节点内存中确定所述目标区块属于所述分叉区块链上的最长链时，将所述目标区块的目标块缓存中的

— 33 —

数据写入目标区块链对应的区块链数据库；所述目标区块的区块生成时间戳大于所述第一区块的区块生成时间戳。

17、一种计算机设备，包括：处理器和存储器；

5 所述处理器与存储器相连，其中，所述存储器用于存储计算机程序，所述处理器用于调用所述计算机程序，以使得所述计算机设备执行权利要求 1-14 任一项所述的方法。

18、一种计算机可读存储介质，所述计算机可读存储介质中存储有计算机程序，该计算机程序适于由处理器加载并执行，以使得具有所述处理器的计算机设备执行权利要求 1-14 任一项所述的方法。

10 19、一种计算机程序产品，当所述计算机程序产品被执行时，用于执行权利要求 1-14 任一项所述的方法。