(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 109977274 A
(43)申请公布日 2019.07.05

(21)申请号 201910255596.2

(22)申请日 2019.03.31

(71)申请人 杭州复杂美科技有限公司
地址 310000 浙江省杭州市西湖区文三路90号东部软件园6号楼7层702室

(72)发明人 张振华 吴思进 王志文

(51)Int.Cl.
G06F 16/901(2019.01)
G06F 16/13(2019.01)
H04L 29/08(2006.01)

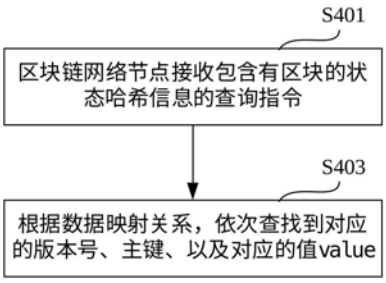
权利要求书2页 说明书7页 附图5页

(54)发明名称

一种数据查询和验证方法、系统、设备及存储介质

(57)摘要

本发明公开了一种数据查询和验证方法、系统、设备及存储介质，属于区块链技术领域。区块链网络节点接收包含有区块的状态哈希值的查询指令，根据数据映射关系，依次查找到对应的版本号、主键、以及对应的值value；其中，所述的版本号为区块的高度，所述区块的状态哈希值是指将包含有前一区块的状态哈希值，区块状态数据信息，区块的版本号进行哈希计算得到；所述的数据映射关系，以及区块的状态哈希值存储在区块链网络节点上。本发明不需要构建树形结构来读取历史值和改变当前值，这对于数据读写的效率提升会有比较大的帮助，可以有效用于节点状态数据的索引及数据验证。



CN 109977274 A

1.一种数据查询方法，其特征在于，区块链网络节点接收包含有区块的状态哈希值的查询指令，根据数据映射关系，依次查找到对应的版本号、主键、以及对应的值value；其中，所述的版本号为区块的高度，所述的数据对应关系包括区块的状态哈希值和区块的版本号之间的对应关系，主键和区块的版本号拼成新的主键，与值value之间的对应关系；所述区块的状态哈希值是指将包含有前一区块的状态哈希值，区块状态数据信息，区块的版本号进行哈希计算得到；所述的数据映射关系，以及区块的状态哈希值存储在区块链网络节点上。

2.根据权利要求1所述的一种数据查询方法，其特征在于，所述的数据映射关系，以及区块的状态哈希值存储在区块链网络节点的KV数据库中。

3.根据权利要求1所述的一种数据查询方法，其特征在于，将本区块的版本号与主键拼成新的主键，与值value一起构成所述的区块状态数据信息。

4.根据权利要求1所述的一种数据查询方法，其特征在于，所述数据映射关系还包括本区块的版本号和本区块的状态哈希值之间的对应关系。

5.根据权利要求4所述的一种数据查询方法，其特征在于，所述数据映射关系还包括最新版本号的主键，与值value的对应关系。

6.一种数据验证方法，其特征在于，在区块链网络节点上，根据权利要求1所述的一种数据查询方法，查找到数据的版本号、主键、对应的值value，计算出区块状态数据信息，根据前一区块的状态哈希值，区块状态数据信息，版本号计算区块的状态哈希值，进行比较，如果哈希值相符，则数据未被篡改，验证通过；否则，数据有误或被改动。

7.根据权利要求6所述的一种数据验证方法，其特征在于，将区块的版本号与主键拼成新的主键，与值value一起构成所述的区块状态数据信息。

8.根据权利要求6所述的一种数据验证方法，其特征在于，所述数据映射关系还包括本区块的版本号和本区块的状态哈希值之间的对应关系。

9.根据权利要求6所述的一种数据验证方法，其特征在于，所述数据映射关系还包括最新版本号的主键，与值value的对应关系。

10.一种数据查询系统，其特征在于，根据权利要求1所述的一种数据查询方法，包括：

用于接收包含有区块的状态哈希值的查询指令的接收单元；

根据数据映射关系，用于依次查找到对应的版本号、主键、以及对应的值value的查询单元；

用于存储各区块的状态哈希值的存储单元一；

用于存储数据映射关系的存储单元二。

11.根据权利要求1所述的一种数据查询系统，其特征在于，所述的第二存储单元为KV数据库。

12.一种数据验证系统，其特征在于，根据权利要求6所述的一种数据验证方法，以及权利要求10或11所述的一种数据查询系统，包括：

用于接收包含有区块的状态哈希值的查询指令的接收单元；

根据数据映射关系，用于依次查找到对应的版本号、主键、以及对应的值value的查询单元；

用于存储各区块的状态哈希值的存储单元一；

用于存储数据映射关系的存储单元二；

用于验证数据是否有误的验证单元。

13.一种设备，其特征在于，所述设备包括：

一个或多个处理器；

存储器，用于存储一个或多个程序，

当所述一个或多个程序被所述一个或多个处理器执行时，使得所述一个或多个处理器执行如权利要求1-9中任一项所述的方法。

14.一种存储有计算机程序的存储介质，其特征在于，该程序被处理器执行时实现如权利要求1-9中任一项所述的方法。

一种数据查询和验证方法、系统、设备及存储介质

技术领域

[0001] 本发明涉及区块链技术领域，尤其涉及一种数据查询和验证方法、系统、设备及存储介质。

背景技术

[0002] 当前区块链系统的数据读写的解决方案中，通常会采用默克尔树(Merkle Tree)的结构，例如比特币的系统中通过默克尔树进行SPV验证、以太坊的系统中通过默克尔前缀树(Merkle Patricia Tree，简称MPT)进行数据的读写等。而现有的默克尔树结构的缺陷在于：所存储的数据限制了系统的读取性能，查询一笔交易的数据需要通过多次读操作来完成。例如，对于一颗20层的默克尔树，查询一个叶子节点的数据需要进行20次读操作来完成，导致数据查询的效率仅为普通数据库的查询效率的1/20，对于每秒能完成10万次读操作的系统，每秒仅能读取5000笔交易的数据。现有方案在节点本地数据库中写默克尔树的数据时若发生崩溃，会导致无法生成区块，对于想快速存储和读取数据信息的用户而言，无疑会带来较大的数据风险。

发明内容

[0003] 1.发明要解决的技术问题

[0004] 为了克服上述技术问题，本发明提供了一种数据查询和验证方法、系统、设备及存储介质。读写操作可以并发互不影响的执行，并且这些数据可以平铺存储在普通的KVDB中，不需要构建树形结构来读取历史值和改变当前值，这对于数据读写的效率提升会有比较大的帮助。

[0005] 2.技术方案

[0006] 为解决上述问题，本发明提供的技术方案为：

[0007] 一种数据查询方法，区块链网络节点接收包含有区块的状态哈希值的查询指令，根据数据映射关系，依次查找到对应的版本号、主键、以及对应的值value；其中，所述的版本号为区块的高度，所述的数据对应关系包括区块的状态哈希值和区块的版本号之间的对应关系，主键和区块的版本号拼成新的主键，与值value之间的对应关系；所述区块的状态哈希值是指将包含有前一区块的状态哈希值，区块状态数据信息，区块的版本号进行哈希计算得到；所述的数据映射关系，以及区块的状态哈希值存储在区块链网络节点上。

[0008] 进一步地，所述的数据映射关系，以及区块的状态哈希值存储在区块链网络节点的KV数据库中。进一步地，将本区块的版本号与主键拼成新的主键，与值value一起构成所述的区块状态数据信息。进一步地，所述数据映射关系还包括本区块的版本号和本区块的状态哈希值之间的对应关系。进一步地，所述数据映射关系还包括最新版本号的主键，与值value的对应关系。

[0009] 本发明还提出了一种数据验证方法，在区块链网络节点上，根据以上所述的一种数据查询方法，查找到数据的版本号、主键、对应的值value，计算出区块状态数据信息，根

据前一区块的状态哈希值，区块状态数据信息，版本号计算区块的状态哈希值，进行比较，如果哈希值相符，则数据未被篡改，验证通过；否则，数据有误或被改动。

[0010] 进一步地，将区块的版本号与主键拼成新的主键，与值value一起构成所述的区块状态数据信息。进一步地，所述数据映射关系还包括本区块的版本号和本区块的状态哈希值之间的对应关系。进一步地，所述数据映射关系还包括最新版本号的主键，与值value的对应关系。

[0011] 相应地，本发明提出了一种数据查询系统，根据以上所述的一种数据查询方法，包括：

[0012] 用于接收包含有区块的状态哈希值的查询指令的接收单元；

[0013] 根据数据映射关系，用于依次查找到对应的版本号、主键、以及对应的值value的查询单元；

[0014] 用于存储各区块的状态哈希值的存储单元一；

[0015] 用于存储数据映射关系的存储单元二。

[0016] 进一步地，所述的存储单元二为KV数据库。

[0017] 一种数据验证系统，根据以上所述的一种数据验证方法，以及以上所述的一种数据查询系统，包括：

[0018] 用于接收包含有区块的状态哈希值的查询指令的接收单元；

[0019] 根据数据映射关系，用于依次查找到对应的版本号、主键、以及对应的值value的查询单元；

[0020] 用于存储各区块的状态哈希值的存储单元一；

[0021] 用于存储数据映射关系的存储单元二；

[0022] 用于验证数据是否有误的验证单元。

[0023] 一种设备，所述设备包括：

[0024] 一个或多个处理器；

[0025] 存储器，用于存储一个或多个程序，

[0026] 当所述一个或多个程序被所述一个或多个处理器执行时，使得所述一个或多个处理器执行如以上任一项所述的方法。

[0027] 一种存储有计算机程序的存储介质，该程序被处理器执行时实现如以上任一项所述的方法。

[0028] 3.有益效果

[0029] 采用本发明提供的技术方案，与现有技术相比，具有如下有益效果：

[0030] 本发明不需要构建树形结构来读取历史值和改变当前值，这对于数据读写的效率提升会有比较大的帮助，可以有效用于节点状态数据的索引及数据验证。可以方便的对状态数据进行存储及查询、验证。存储方便，查询效率高，验证简单有效。

附图说明

[0031] 图1为本发明一实施例中数据查询方法的流程图。

[0032] 图2为本发明一实施例中数据验证方法的流程图。

[0033] 图3为本发明一实施例提供的一种数据查询系统的结构示意图。

[0034] 图4为本发明一实施例提供的一种数据验证系统的结构示意图。

[0035] 图5为本发明一优选实施例提供的一种数据存储方法的流程图。

[0036] 图6为本发明一实施例提供的一种数据存储系统的结构示意图。

[0037] 图7为数据存取过程示意图。

[0038] 图8为图6一优选实施例提供的一种数据存储系统的结构示意图。

[0039] 图9为本发明实施例提供的一种设备的结构示意图。

具体实施方式

[0040] 为进一步了解本发明的内容，结合附图及实施例对本发明作详细描述。

[0041] 下面结合附图和实施例对本申请作进一步的详细说明。可以理解的是，此处所描述的具体实施例仅仅用于解释相关发明，而非对该发明的限定。另外还需要说明的是，为了便于描述，附图中仅示出了与发明相关的部分。

[0042] 本发明中所述的第一、第二等词语，是为了描述本发明的技术方案方便而设置，并没有特定的限定作用，均为泛指，对本发明的技术方案不构成限定作用。

[0043] 需要说明的是，在不冲突的情况下，本申请中的实施例及实施例中的特征可以相互组合。下面将参考附图并结合实施例来详细说明本申请。

[0044] 实施例1

[0045] 如图1所示，一种数据查询方法，步骤为：

[0046] S401、区块链网络节点接收包含有区块的状态哈希值的查询指令，S403、根据数据映射关系，依次查找到对应的版本号、主键、以及对应的值value；其中，所述的版本号为区块的高度，所述的数据对应关系包括区块的状态哈希值和区块的版本号之间的对应关系，主键和区块的版本号拼成新的主键，与值value之间的对应关系；所述区块的状态哈希值是指将包含有前一区块的状态哈希值，区块状态数据信息，区块的版本号进行哈希计算得到；所述的数据映射关系，以及区块的状态哈希值存储在区块链网络节点上。

[0047] 进一步地，所述的数据映射关系，以及区块的状态哈希值存储在区块链网络节点的KV数据库中。进一步地，将本区块的版本号与主键拼成新的主键，与值value一起构成所述的区块状态数据信息。进一步地，所述数据映射关系还包括本区块的版本号和本区块的状态哈希值之间的对应关系。进一步地，所述数据映射关系还包括最新版本号的主键，与值value的对应关系。

[0048] 本发明还提出了一种数据验证方法，如图2所示，在区块链网络节点20上，根据以上所述的一种数据查询方法，步骤为：S501、查找到数据的版本号、主键、对应的值value，计算出区块状态数据信息，S503、根据前一区块的状态哈希值，区块状态数据信息，版本号计算区块的状态哈希值，进行比较，如果哈希值相符，则数据未被篡改，验证通过；否则，数据有误或被改动。

[0049] 进一步地，将区块的版本号与主键拼成新的主键，与值value一起构成所述的区块状态数据信息。进一步地，所述数据映射关系还包括本区块的版本号和本区块的状态哈希值之间的对应关系。进一步地，所述数据映射关系还包括最新版本号的主键，与值value的对应关系。

[0050] 相应地，本发明提出了一种数据查询系统，根据以上所述的一种数据查询方法，如

图3所示，包括：

[0051] 用于接收包含有区块的状态哈希值的查询指令的接收单元601；

[0052] 根据数据映射关系，用于依次查找到对应的版本号、主键、以及对应的值value的查询单元603；

[0053] 用于存储各区块的状态哈希值的存储单元一605；

[0054] 用于存储数据映射关系的存储单元二607。

[0055] 进一步地，所述的存储单元二607为KV数据库。

[0056] 一种数据验证系统，根据以上所述的一种数据验证方法，以及以上所述的一种数据查询系统，如图4所示，包括：

[0057] 用于接收包含有区块的状态哈希值的查询指令的接收单元601；

[0058] 根据数据映射关系，用于依次查找到对应的版本号、主键、以及对应的值value的查询单元603；

[0059] 用于存储各区块的状态哈希值的存储单元一605；

[0060] 用于存储数据映射关系的存储单元二607；

[0061] 用于验证数据是否有误的验证单元609。

[0062] 与本发明的查询方法、验证方法相对应的是一种数据存储方法，如图5所示：

[0063] S101、将前一区块的状态哈希值，区块状态数据信息，版本号一起进行哈希计算，得到区块的状态哈希值；

[0064] S103、区块的状态哈希值和数据映射关系一起存储在区块链网络节点本地的数据库内。

[0065] 其中，所述版本号为区块的高度，所述区块链网络节点本地的数据库是指KV数据库；将版本号与主键拼成新的主键，与值value一起构成所述的区块状态数据信息。所述数据映射关系包括区块的状态哈希值和版本号之间的对应关系，主键和版本号拼成新的主键，与值value之间的对应关系，也就是区块状态数据信息。所述主键也可能包含其他标识信息，比如用于区分该主键的用途的信息等。

[0066] 所述数据映射关系还包括版本号和区块的状态哈希值之间的对应关系。所述数据映射关系还包括最新版本号的主键，与值value的对应关系。

[0067] 区块链中历史数据不能修改，修改只发生在当前正在生成的区块上；因此，如图7所示，读历史数据的比例高(对历史区块数据的读取)，写数据的比例低(只在新生成区块中修改数据)。读取区块101、102和103中的一项，与写入区块104可以同时进行，互不影响。

[0068] 主键为Key的变量值在区块高度101、10状态2中均发生变化，对应的在KVDB中有Key:101->value101和Key:102->value102两个历史版本数据，在当前区块104中正在写入新值Key:104->value104；

[0069] 那么对于历史数据Key:101、Key:102可以有很多并发的读，对于Key:104有唯一的当前写，这些读写操作可以并发互不影响的执行，并且这些数据可以平铺存储在普通的KVDB中，也就是按简单的kv对应关系进行存储不需要保存额外的树型结构的信息。读写只要根据区块高度作为版本号与Key构成唯一主键即可(Key:version或者Key:height)，不需要构建树形结构来读取历史值和改变当前值，这对于数据读写的效率提升会有比较大的帮助，可以有效用于节点状态数据的索引。

[0070] 如图8所示，区块高度(版本号)为0、1、2和3，版本号为0的区块的状态哈希值(statehash)h0＝hash(nil,kvset,height)，区块的状态数据KVSet为K1:V1,K2:V2...;数据映射关系包括区块的状态哈希值和版本号之间的对应关系(如图8所示hash->version)，主键和版本号拼成新的主键，与值value之间的对应关系，也就是区块状态数据信息(如图8所示key:version->value)。所述数据映射关系还包括版本号和区块的状态哈希值之间的对应关系(如图8所示version->hash)。所述数据映射关系还包括最新版本号的主键，与值value的对应关系(如图8所示latest:key->value)。这一类数据的存储，为按列表查询最新版本的数据创造了条件。由于kv数据库是按照key值有序存放的，那么latest开头的数据相邻存放，如果key值有某些共同前缀的数据是可以被连续取到多条而不消耗性能的。如果没有latest:key数据，那么每个key对应多版本数据，就无法直接进行高效的列表操作。

[0071] 版本号为0的区块的状态哈希值以及数据映射关系均写入到本地key-value数据库中存储。版本号为1的区块的状态哈希值(statehash)h1＝hash(h0,kvset,height)，区块的状态数据KVSet为K1:V1′,K3:V3...;数据映射状态关系包括区块的状态哈希值和版本号之间的对应关系(如图8所示hash->version)，主键和版本号拼成新的主键，与值value之间的对应关系，也就是区块状态数据信息(如图8所示key:version->value)。所述数据映射关系还包括版本号和区块的状态哈希值之间的对应关系(如图8所示version->hash)。所述数据映射关系还包括最新版本号的主键，与值value的对应关系(如图8所示latest:key->value)。版本号为1的区块的状态哈希值以及数据映射关系均写入到本地key-value数据库中存储。

[0072] 版本号为2的区块的状态哈希值(statehash)h2＝hash(h1,kvset,height)，区块的状态数据KVSet为K4:V4,K5:V5...;数据映射关系包括区块的状态哈希值和版本号之间的对应关系(如图8所示hash->version)，主键和版本号拼成新的主键，与值value之间的对应关系，也就是区块状态数据信息(如图8所示key:version->value)。所述数据映射关系还包括版本号和区块的状态哈希值之间的对应关系(如图8所示version->hash)。所述数据映射关系还包括最新版本号的主键，与值value的对应关系(如图8所示latest:key->value)。版本号为1的区块的状态哈希值以及数据映射关系均写入到本地key-value数据库中存储。

[0073] 版本号为3的区块的状态哈希值(statehash)h3＝hash(h2,kvset,height)，区块的状态数据KVSet，数据映射关系包括区块的状态哈希值和版本号之间的对应关系(如图8所示hash->version)，主键和版本号拼成新的主键，与值value之间的对应关系，也就是区块状态数据信息(如图8所示key:version->value)。所述数据映射关系还包括版本号和区块的状态哈希值之间的对应关系(如图8所示version->hash)。所述数据映射关系还包括最新版本号的主键，与值value的对应关系(如图8所示latest:key->value)。版本号为1的区块的状态哈希值以及数据映射关系均写入到本地key-value数据库中存储。

[0074] 如图6所示，相应地，本实施例提出了一种数据存储系统，分布在区块链网络各节点20上，根据以上所述的一种数据存储方法，包括：

[0075] 用于存储各区块的状态哈希值的第一存储单元305；

[0076] 用于存储数据映射关系的第二存储单元307；

[0077] 用于计算各区块的状态哈希值的哈希运算单元303。

[0078] 所述的第二存储单元307为KV数据库。

[0079] 将版本号与主键拼成新的主键，与值value一起构成所述的区块状态数据信息。

[0080] 所述数据映射关系包括区块的状态哈希状态值和版本号之间的对应关系，主键和版本号拼成新的主键，与值value之间的对应关系。所述数据映射关系还包括版本号和区块的状态哈希值之间的对应关系。所述数据映射关系还包括最新版本号的主键，与值value的对应关系。

[0081] 实施例2

[0082] 一种设备，所述设备包括：

[0083] 一个或多个处理器；

[0084] 存储器，用于存储一个或多个程序，

[0085] 当所述一个或多个程序被所述一个或多个处理器执行时，使得所述一个或多个处理器执行如以上任一项所述的方法。

[0086] 一种存储有计算机程序的存储介质，该程序被处理器执行时实现如以上任一项所述的方法。

[0087] 图9为本发明一实施例提供的一种设备的结构示意图。

[0088] 如图9所示，作为另一方面，本申请还提供了一种设备500，包括一个或多个中央处理单元(CPU)501，其可以根据存储在只读存储器(ROM)502中的程序或者从存储部分508加载到随机访问存储器(RAM)503中的程序而执行各种适当的动作和处理。在RAM503中，还存储有设备500操作所需的各种程序和数据。CPU501、ROM502以及RAM503通过总线504彼此相连。输入/输出(I/O)接口505也连接至总线504。

[0089] 以下部件连接至I/O接口505：包括键盘、鼠标等的输入部分506；包括诸如阴极射线管(CRT)、液晶显示器(LCD)等以及扬声器等的输出部分507；包括硬盘等的存储部分508；以及包括诸如LAN卡、调制解调器等的网络接口卡的通信部分509。通信部分509经由诸如因特网的网络执行通信处理驱动器510也根据需要连接至I/O接口505。可拆卸介质511，诸如磁盘、光盘、磁光盘、半导体存储器等等，根据需要安装在驱动器510上，以便于从其上读出的计算机程序根据需要被安装入存储部分508。

[0090] 特别地，根据本申请公开的实施例，上述任一实施例描述的方法可以被实现为计算机软件程序。例如，本申请公开的实施例包括一种计算机程序产品，其包括有形地包含在机器可读介质上的计算机程序，所述计算机程序包含用于执行上述任一实施例描述的方法的程序代码。在这样的实施例中，该计算机程序可以通过通信部分509从网络上被下载和安装，和/或从可拆卸介质511被安装。

[0091] 作为又一方面，本申请还提供了一种计算机可读存储介质，该计算机可读存储介质可以是上述实施例的装置中所包含的计算机可读存储介质；也可以是单独存在，未装配入设备中的计算机可读存储介质。计算机可读存储介质存储有一个或者一个以上程序，该程序被一个或者一个以上的处理器用来执行描述于本申请的方法。

[0092] 附图中的流程图和框图，图示了按照本发明各种实施例的系统、方法和计算机程序产品的可能实现的体系架构、功能和操作。在这点上，流程图或框图中的每个方框可以代表一个模块、程序段、或代码的一部分，该模块、程序段、或代码的一部分包含一个或多个用于实现规定的逻辑功能的可执行指令。也应当注意，在有些作为替换的实现中，方框中所标注的功能也可以以不同于附图中所标注的顺序发生。例如，两个接连地表示的方框实际上

可以基本并行地执行，它们有时也可以按相反的顺序执行，这根据所涉及的功能而定。也要注意的是，框图和/或流程图中的每个方框、以及框图和/或流程图中的方框的组合，可以通过执行规定的功能或操作的专用的基于硬件的系统来实现，或者可以通过专用硬件与计算机指令的组合来实现。

[0093] 描述于本申请实施例中所涉及到的单元或模块可以通过软件的方式实现，也可以通过硬件的方式来实现。所描述的单元或模块也可以设置在处理器中，例如，各所述单元可以是设置在计算机或移动智能设备中的软件程序，也可以是单独配置的硬件装置。其中，这些单元或模块的名称在某种情况下并不构成对该单元或模块本身的限定。

[0094] 以上描述仅为本申请的较佳实施例以及对所运用技术原理的说明。本领域技术人员应当理解，本申请中所涉及的发明范围，并不限于上述技术特征的特定组合而成的技术方案，同时也应涵盖在不脱离本申请构思的情况下，由上述技术特征或其等同特征进行任意组合而形成的其它技术方案。例如上述特征与本申请中公开的(但不限于)具有类似功能的技术特征进行互相替换而形成的技术方案。

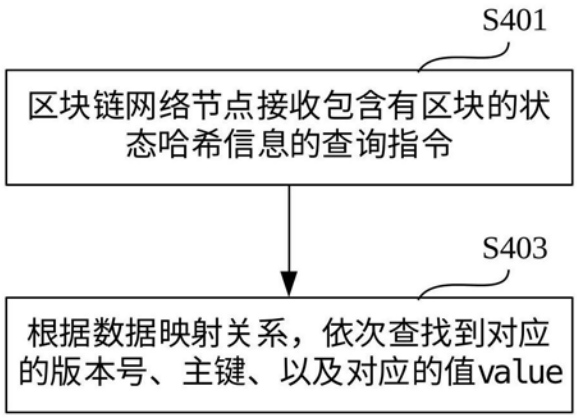

S401
区块链网络节点接收包含有区块的状态哈希信息的查询指令
S403
根据数据映射关系，依次查找到对应的版本号、主键、以及对应的值value

图1

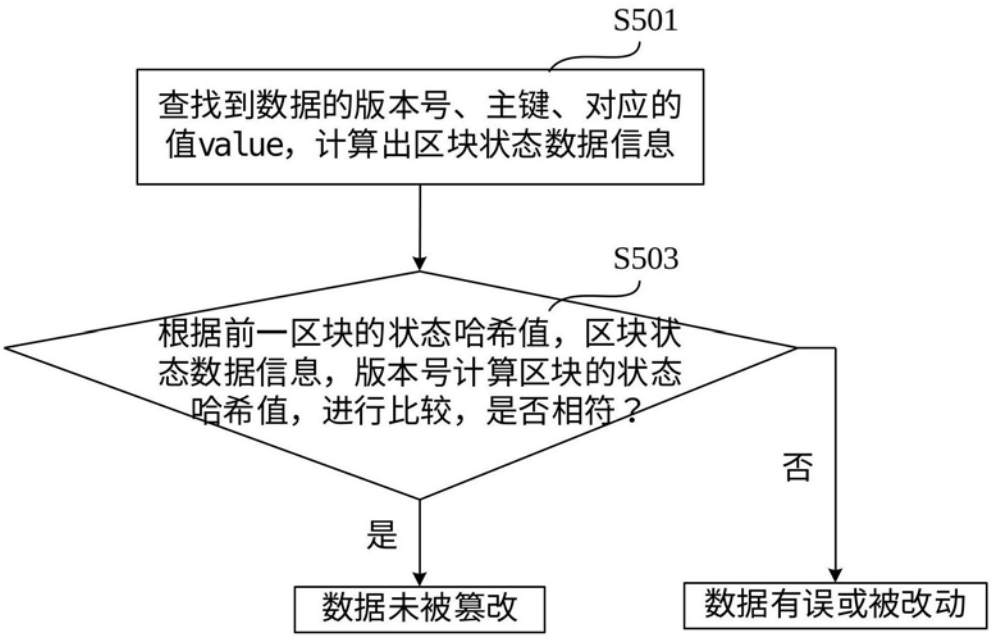

S501
查找到数据的版本号、主键、对应的值value，计算出区块状态数据信息
S503
根据前一区块的状态哈希值，区块状态数据信息，版本号计算区块的状态哈希值，进行比较，是否相符？
否
是
数据未被篡改
数据有误或被改动

图2

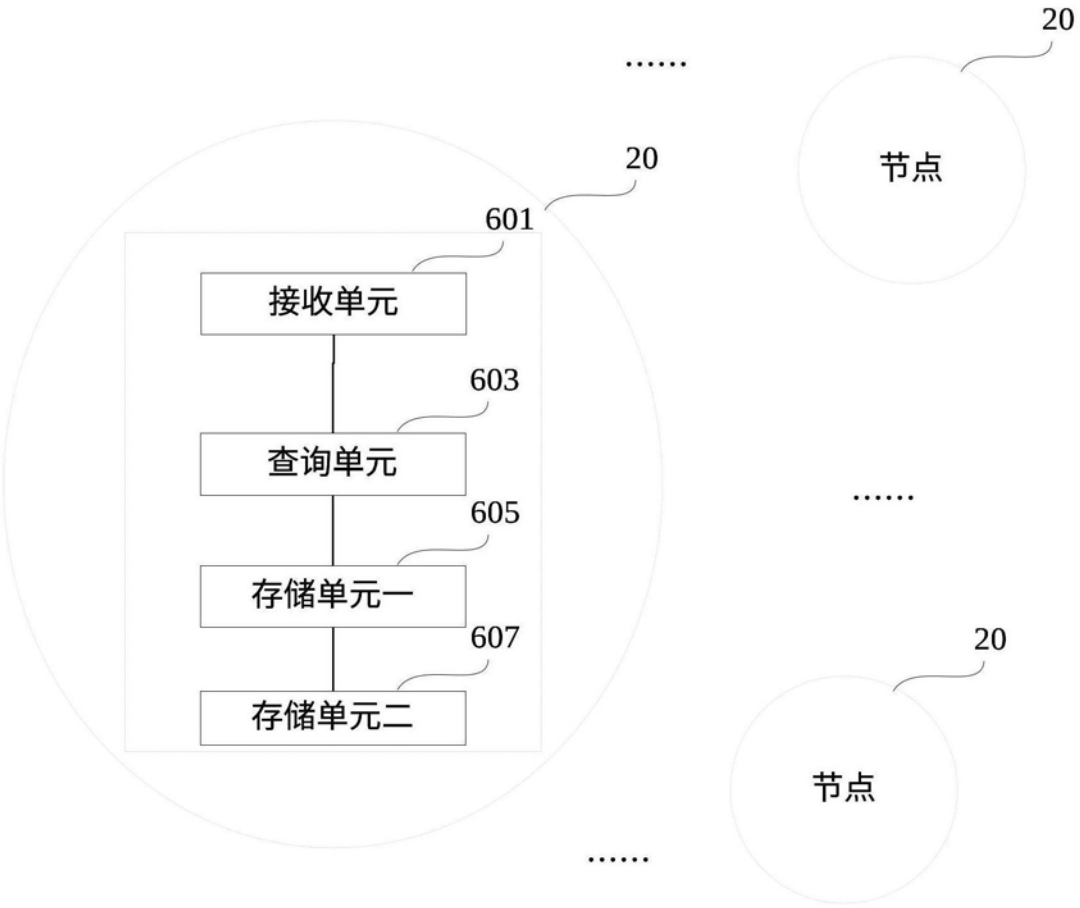

20
.....
节点
20
601
接收单元
603
查询单元
605
存储单元一
607
存储单元二
.....
20
节点
.....

图3

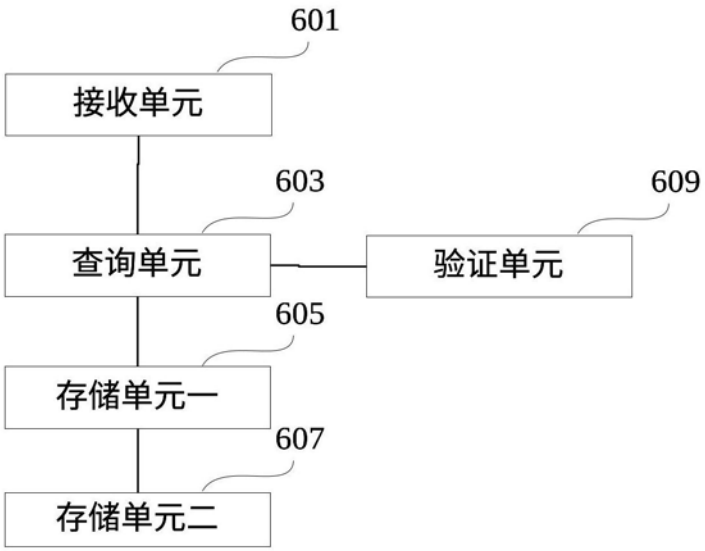

601
接收单元
603
609
查询单元
验证单元
605
存储单元一
607
存储单元二

图4

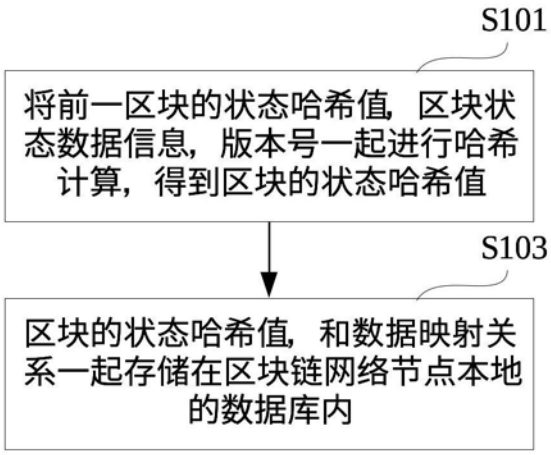

S101
将前一区块的状态哈希值，区块状态数据信息，版本号一起进行哈希计算，得到区块的状态哈希值
S103
区块的状态哈希值，和数据映射关系一起存储在区块链网络节点本地的数据库内

图5

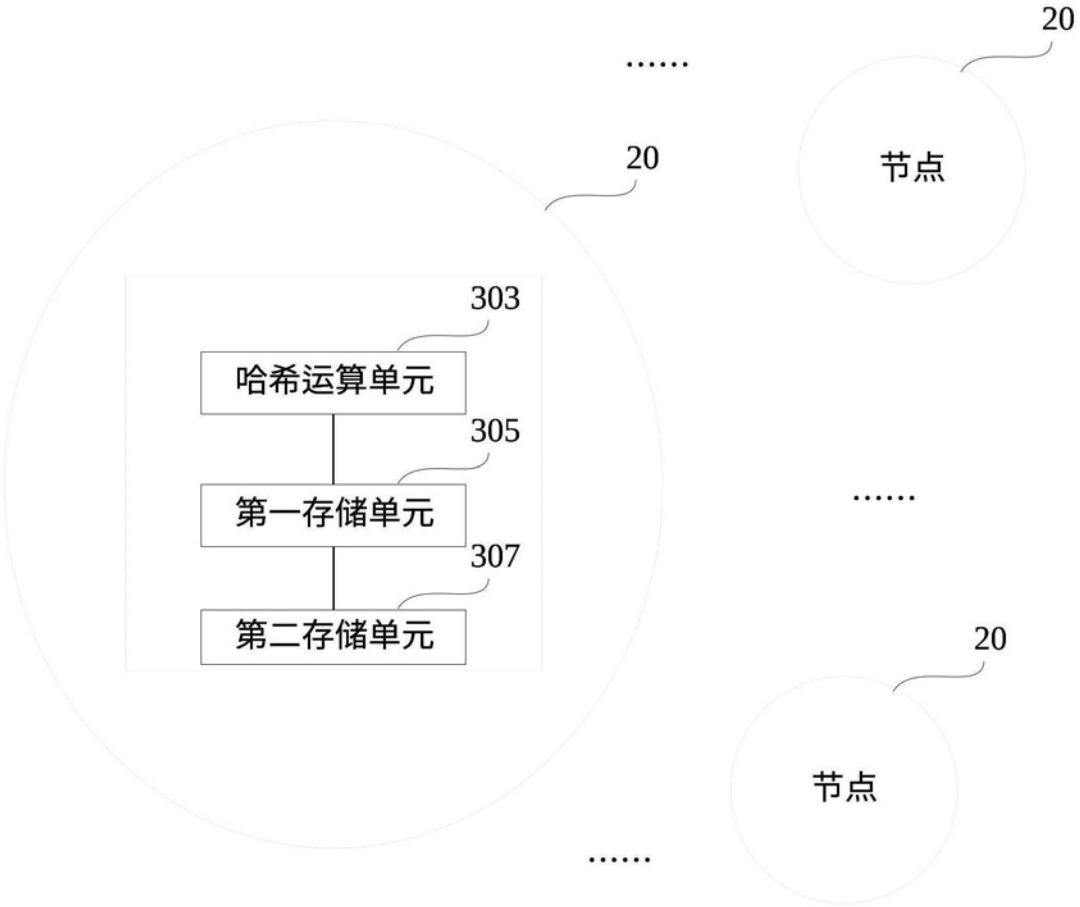

20
.....
节点
20
303
哈希运算单元
305
第一存储单元
307
第二存储单元
.....
20
节点
.....

图6

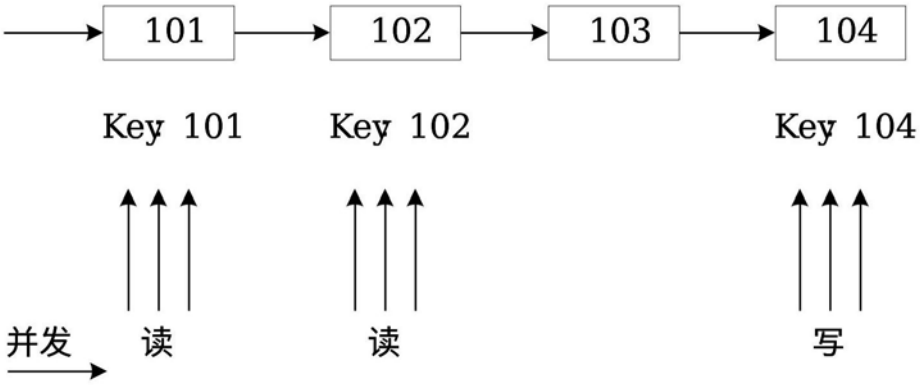

101
102
103
104
Key 101
Key 102
Key 104
并发
读
读
写

图7

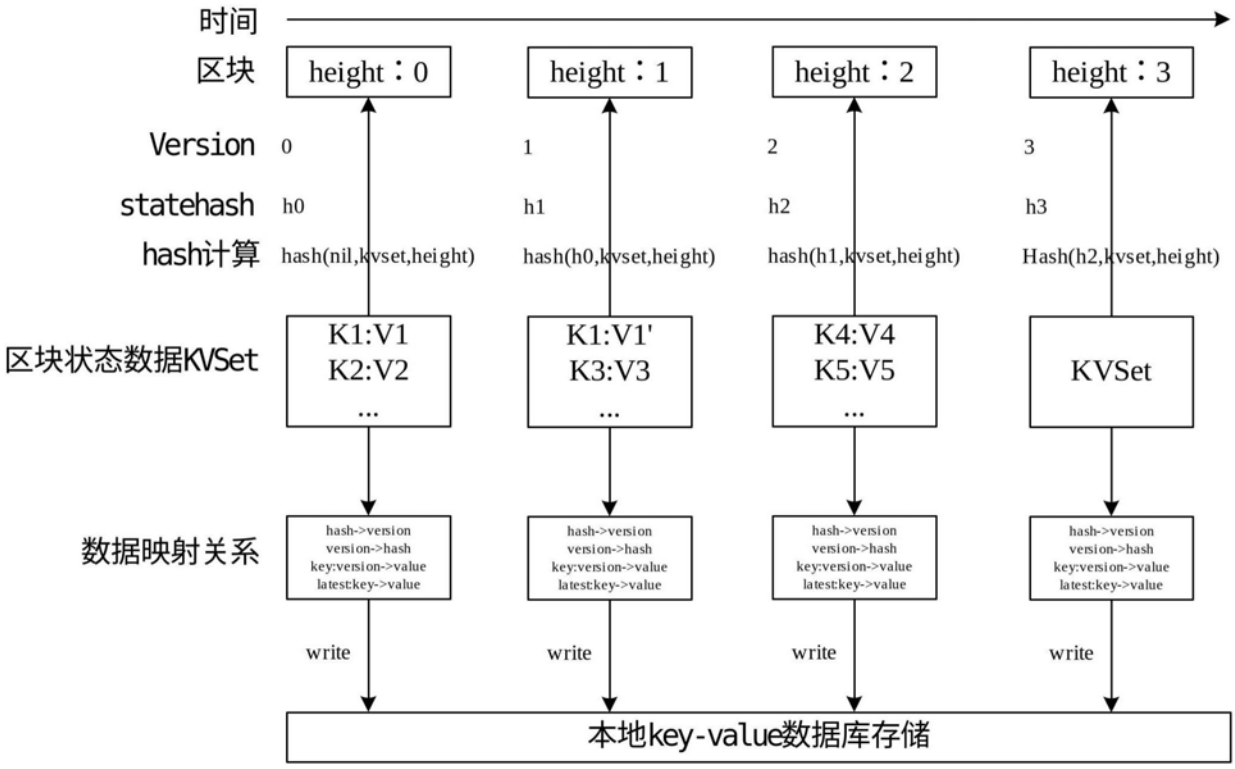

时间
区块
height：0
height：1
height：2
height：3
Version
0
1
2
3
statehash
h0
h1
h2
h3
hash计算
hash(nil,kvset,height)
hash(h0,kvset,height)
hash(h1,kvset,height)
Hash(h2,kvset,height)
区块状态数据KVSet
K1:V1
K2:V2
...
K1:V1'
K3:V3
...
K4:V4
K5:V5
...
KVSet
数据映射关系
hash->version
version->hash
key:version->value
latest:key->value
write
本地key-value数据库存储

图8

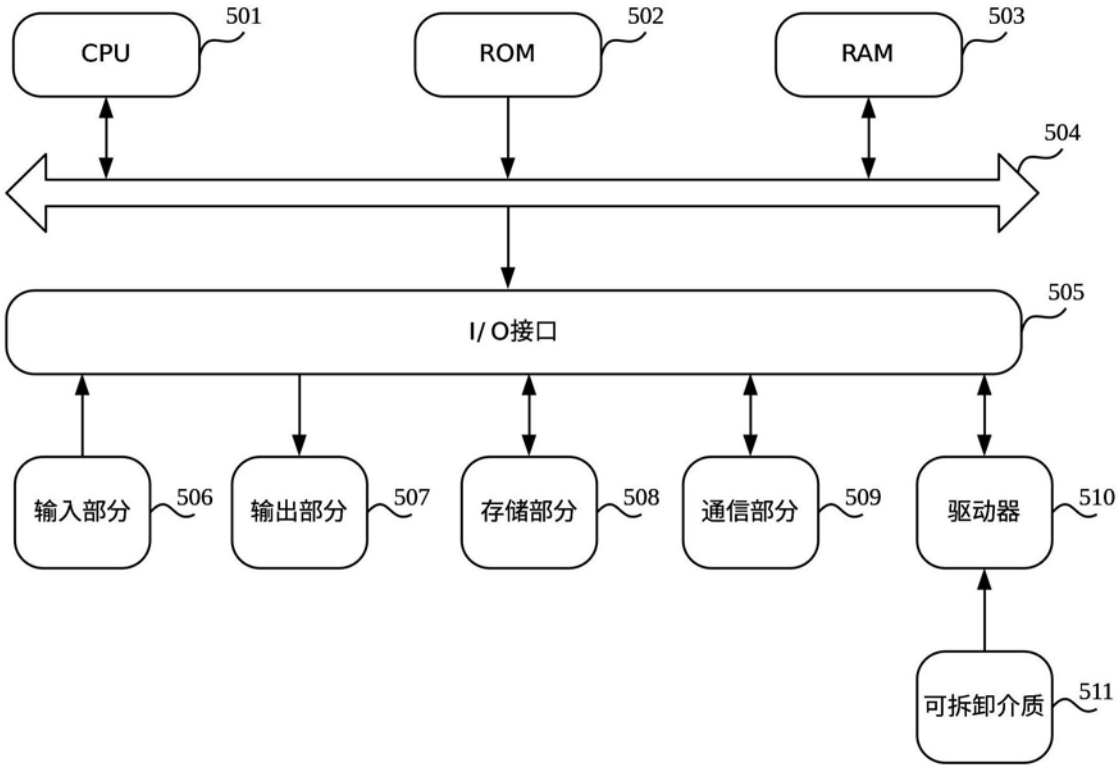

CPU
501
ROM
502
RAM
503
504
I/O接口
505
输入部分
506
输出部分
507
存储部分
508
通信部分
509
驱动器
510
可拆卸介质
511

图9