

数据处理方法、装置、计算机设备及存储介质

本申请要求于 2019 年 01 月 23 日提交的申请号为 201910063326.1、发明名称为“数据处理方法、装置、计算机设备及存储介质”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。

技术领域

本申请涉及计算机技术领域，特别是涉及一种数据处理方法、装置、计算机设备及存储介质。

背景技术

随着科学技术的飞速发展，各种技术突飞猛进，大量先进技术不断涌现，区块链技术即属于当今时代一个非常重要的技术，在非常多的场景中都得到了应用。

传统方法中，区块链主要用于记账，即用于存储交易记录。交易记录在区块链上一经存储，则无法删除和篡改，保证了所存储的交易记录真实有效。然而，有些情况下不需要一直对区块链中的所有交易记录都保留，只需要保留一部分交易记录，比如，对账完毕后的交易记录则可以不再保留，仅保留没有对账的交易记录即可。但是，基于区块链上的数据无法删除和篡改的特性，传统方法并不能将不需保留的交易记录删除，使得这些不必要的数据占据大量空间，从而浪费大量存储资源。

发明内容

基于此，本申请提供了一种数据处理方法、装置、计算机设备及存储介质，可以节约区块链中的存储资源。

一种数据处理方法，所述方法由区块链节点执行，所述方法包括：
针对第一区块文件中的每个区块，检测所述区块中需保留的交易记录；
从所述区块的哈希树中，筛选用于证明所述交易记录属于所述区块的节点；
根据筛选出的所述节点构建节点保留树；
存储所述交易记录至所述节点保留树的相应叶节点中；

将所述区块的区块头和存储了交易记录的节点保留树关联更新至第二区块文件中；所述区块的区块头在所述第二区块文件中相邻于前一区块的区块头；及

在对每个所述区块更新处理完毕后，删除所述第一区块文件。

一种数据处理装置，所述装置用于区块链节点中，所述装置包括：

节点保留模块，用于针对第一区块文件中的每个区块，检测所述区块中需保留的交易记录；从所述区块的哈希树中，筛选用于证明所述交易记录属于所述区块的节点；根据筛选出的所述节点构建节点保留树；

交易存储模块，用于存储所述交易记录至所述节点保留树的相应叶节点中；

区块更新模块，用于将所述区块的区块头和存储了交易记录的节点保留树关联更新至第二区块文件中；所述区块的区块头在所述第二区块文件中相邻于前一区块的区块头；及

删除模块，用于在针对每个所述区块的更新处理完毕后，删除所述第一区块文件。

一种计算机设备，包括存储器和处理器，所述存储器中存储有计算机程序，所述计算机程序被所述处理器执行时，使得所述处理器执行如下步骤：

针对第一区块文件中的每个区块，检测所述区块中需保留的交易记录；

从所述区块的哈希树中，筛选用于证明所述交易记录属于所述区块的节点；

根据筛选出的所述节点构建节点保留树；

存储所述交易记录至所述节点保留树的相应叶节点中；

将所述区块的区块头和存储了交易记录的节点保留树关联更新至第二区块文件中；所述区块的区块头在所述第二区块文件中相邻于前一区块的区块头；及

在对每个所述区块更新处理完毕后，删除所述第一区块文件。

一种计算机可读存储介质，所述计算机可读存储介质上存储有计算机程序，所述计算机程序被处理器执行时，使得所述处理器执行如下步骤：

针对第一区块文件中的每个区块，检测所述区块中需保留的交易记录；

从所述区块的哈希树中，筛选用于证明所述交易记录属于所述区块的节点；

根据筛选出的所述节点构建节点保留树；

存储所述交易记录至所述节点保留树的相应叶节点中；

将所述区块的区块头和存储了交易记录的节点保留树关联更新至第二区块文件中；所述区块的区块头在所述第二区块文件中相邻于前一区块的区块头；
及

在对每个所述区块更新处理完毕后，删除所述第一区块文件。

上述数据处理方法、装置、计算机设备及存储介质，针对第一区块文件中的每个区块，检测区块中需保留的交易记录；从该区块的哈希树中，筛选用于证明该需保留的交易记录属于该区块的节点；根据筛选出的节点构建节点保留树；存储交易记录至节点保留树的相应叶节点中。将区块的区块头和存储了交易记录的节点保留树关联更新至第二区块文件中；区块头在第二区块文件中相邻于前一区块的区块头。在第二区块文件中，由于区块头并未发生改变而且区块头仍然依次相邻，所以维持了区块链上通过哈希值首尾相连的特性，这样一来，根据保存在节点保留树的叶节点的交易记录，以及节点保留树中的节点，就可以证明该交易记录属于原来所对应的区块，相当于仍然保留了原始哈希树校验数据真实性的功能，第二区块文件中保留的交易记录也就仍然无法伪造和篡改。在针对每个区块的更新处理完毕后，删除第一区块文件。由于第二区块文件中并未存储原有区块中不需要保留的交易记录，而仅保存了需保留的交易记录和用于校验该交易记录真实性的节点保留树，所以相较于第一区块文件大大节省了存储资源。

附图说明

图 1 为一个实施例中数据处理方法的应用场景图；

图 2 为一个实施例中数据处理方法的流程示意图；

图 3 为一个实施例中构建节点保留树的原理示意图；

图 4 为一个实施例中对账步骤的流程示意图；

图 5 为另一个实施例中数据处理方法的流程示意图；

图 6 为一个实施例中数据处理装置的框图；

图 7 为另一个实施例中数据处理装置的框图；

图 8 为一个实施例中计算机设备的内部结构示意图。

具体实施方式

为了使本申请的目的、技术方案及优点更加清楚明白，以下结合附图及实施例，对本申请进行进一步详细说明。应当理解，此处所描述的具体实施例仅仅用以解释本申请，并不用于限定本申请。

图 1 为一个实施例中数据处理方法的应用场景图。参照图 1，该应用场景中包括通过网络连接的第一对账设备 110、第二对账设备 120 和区块链节点 130。区块链节点 130，用于存放区块链中的区块。区块链节点 130 由一台或多台服务器构成。多个区块链节点 130 构成区块链平台。区块链平台可以是私有区块链平台也可以是开发的区块链平台。区块链平台可以是云区块链平台。云区块链平台，是构建于云基础之上，让用户在开放的云平台上能够快速构建自身所需的基础设施和区块链服务的平台。

第一对账设备 110 和第二对账设备 120，是对账双方用于实现对账的设备。第一对账设备 110 和第二对账设备 120 可以是终端或服务器。终端可以是智能电视机、台式计算机或移动终端，移动终端可以包括手机、平板电脑、笔记本电脑、个人数字助理和穿戴式设备等中的至少一种。服务器可以用独立的服务器或者是多个物理服务器组成的服务器集群来实现。

第一对账设备 110 和第二对账设备 120 可以将需要对账的交易记录存储至区块链节点 130 中的区块链上。区块链上的各区块存储于第一区块文件中。区块链节点 130 可以针对第一区块文件中的每个区块，检测该区块中需保留的交易记录；从该区块的哈希树中，筛选用于证明该交易记录属于该区块的节点；根据筛选出的该节点构建节点保留树；存储该交易记录至该节点保留树的相应叶节点中；将该区块的区块头和存储了交易记录的节点保留树关联更新至第二区块文件中；该区块的区块头在该第二区块文件中相邻于前一区块的区块头；在对每个该区块更新处理完毕后，删除该第一区块文件。第二区块文件中存储了每个区块更新后的需要保留的交易记录而去掉了不需要保留的交易记录，从而节省了存储空间。此外，第二区块文件中每个区块的区块头依然按照在第一区块文件中的顺序相邻，即仍然维持了区块链上通过哈希值首尾相连的特性，那么，第二区块文件中保留的交易记录也就仍然无法伪造和篡改。

需要说明的是，图 1 只是本申请中数据处理方法的一个应用场景。本申请中的数据处理方法并不限定于对图 1 的对账场景中的交易记录进行合理清除，

还可以是对其他场景中所存储的交易记录进行清除。

图 2 为一个实施例中数据处理方法的流程示意图。本实施例主要以该数据处理方法应用于图 1 中的区块链节点 130 进行举例说明。参照图 2，该方法具体包括如下步骤：

S202，针对第一区块文件中的每个区块，检测区块中需保留的交易记录。

其中，需保留的交易记录，是不需要清除的交易记录。第一区块文件是存储于区块链节点上的原始区块链的文件。第一区块文件中包括构成区块链的多个区块。每个区块中可以包括一条或多条交易记录。需保留的交易记录可以为一条或多条。

在一个实施例中，交易记录可以是用于对账的交易记录。在其他实施例中，交易记录还可以是纯粹用于存储备份的交易记录。可以理解，交易记录还可以是其他使用场景下需要使用到的交易记录。对此不作限定。

在一个实施例中，针对第一区块文件中的每个区块，区块链节点可以检测该区块中直接指定的需保留的交易记录。

在一个实施例中，区块链节点也可以对各区块中的交易记录进行分析，从中确定出需保留的交易记录。可选地，区块链节点可以获取区块中各交易记录的状态，将处于有效状态的交易记录确定为需保留的交易记录。区块链节点也可以筛选区块中位于未成功处理集合中的交易记录，得到需保留的交易记录。

可以理解，区块链节点可以从第一区块文件中的首个区块开始，逐个区块执行步骤 S202~S210。区块链节点也可以针对第一区块文件中的各个区块并发地执行步骤 S202~S210。这里不做限定。

需要说明的是，区块链节点从区块中可能检测出需要保留的交易记录，也可能检测得到不存在需保留的交易记录。那么，响应于检测出区块中需保留的交易记录，则执行步骤 S204~S210，响应于检测到区块中不存在需保留的交易记录，则可以只将该区块的区块头相邻于前一区块的区块头更新至第二区块文件中。比如，针对不存在需保留的交易记录的区块，区块链节点可以在第二区块文件中确定其前一区块的区块头，并于第二区块文件中在该前一区块的区块头之后更新存储该区块的区块头。即针对不存在需保留的交易记录的区块，在第二区块文件中可以仅保留区块头而不需要保留任何交易记录。

S204，从区块的哈希树中，筛选用于证明需保留的交易记录属于区块的节点。

其中，区块的哈希树，是根据该区块中的各交易记录生成的哈希树。哈希树的叶节点与区块中各交易记录的哈希值一一对应。哈希树的根节点的哈希值，根据各叶节点的哈希值计算得到。哈希树的根节点的哈希值与该区块唯一对应。

用于证明需保留的交易记录属于区块的节点，是指从该需保留的交易记录所对应叶节点的哈希值计算得到根节点的哈希值时所需要的哈希值对应的节点。可以理解，因为要从该需保留的交易记录所对应叶节点的哈希值计算得到根节点的哈希值，所以筛选的节点中包括该需保留的交易记录所对应叶节点和根节点。

在一个实施例中，哈希树可以为双层哈希树。双层哈希树，是指包括根节点和叶节点的两层级的哈希树。双层哈希树的根节点的哈希值，可以由每个叶节点的哈希值串联合并后进行哈希运算得到。可以理解，当哈希树为双层哈希树时，筛选的用于证明交易记录属于区块的节点，即为哈希树的全部节点。

在另一个实施例中，哈希树也可以是多层哈希树。多层哈希树，是指层级大于两层的哈希树。这种情况下，用于证明所述交易记录属于所述区块的节点，就仅为多层哈希树的一部分节点。在一个实施例中，多层哈希树可以为默克尔树。其中，默克尔树（Merkle Tree），是一种多层级的树形数据结构。默克尔树的各叶节点以各交易记录的哈希值作为标签，而叶节点以其子节点标签串联后进行哈希运算得到的哈希值作为标签。可以理解，默克尔树的根节点以最接近该根节点的这一层级的节点的标签串联后进行哈希运行得到的哈希值为标签。

需要说明的是，当需保留的交易记录为多条时，则针对每个需保留的交易记录，从区块的哈希树中，筛选用于证明该需保留的交易记录属于这个区块的节点。可以理解，针对多条需保留的交易记录筛选的节点可能存在重复情况。

S206，根据筛选出的节点构建节点保留树。

筛选出的节点，即为原始的哈希树中需保留的节点。节点保留树，是根据筛选出的需保留的节点重新构建的哈希树。

当只有一条需保留的交易记录时，则可以仅根据针对该条交易记录筛选出的节点构建节点保留树。当有多条需保留的交易记录时，则可以将针对多条需保留的交易记录筛选出的节点进行汇总并去重，然后根据去重后的节点重新构建节点保留树。需要说明的是，构建的节点保留树中保留了筛选出的各节点在原始的哈希树中所存在的层级关系。

为了便于理解，现结合图3对节点保留树进行举例说明。参照图3，第n-1

个区块的区块头中包括前一区块的哈希值和第 $n-1$ 个区块自身的根哈希，其中，图 3 中的根哈希，即指默克尔树的根节点，第 n 个区块的区块头中包括第 $n-1$ 个区块的哈希值和第 n 个区块自身的根哈希，第 $n+1$ 个区块的区块头中包括第 n 个区块的哈希值和第 $n+1$ 个区块自身的根哈希。以第 n 个区块为例，假设该第 n 个区块中包括 4 条交易记录 $Tx(1) \sim Tx(4)$ ，分别对 $Tx(1) \sim Tx(4)$ 进行哈希计算，得到相应的哈希值 $H(1) \sim H(4)$ 。那么，以这 4 条交易记录，即可生成包括叶节点 $H(1) \sim H(4)$ 、 $H(1+2)$ 、 $H(3+4)$ 、以及根节点的哈希树。其中， $H(1+2)$ 为 $H(1)$ 和 $H(2)$ 的父节点，其所对应的哈希值是根据 $H(1)$ 和 $H(2)$ 串联后进行哈希计算得到。 $H(3+4)$ 为 $H(3)$ 和 $H(4)$ 的父节点，其所对应的哈希值是根据 $H(3)$ 和 $H(4)$ 串联后进行哈希计算得到。根节点为 $H(1+2)$ 和 $H(3+4)$ 的父节点，其所对应的哈希值是根据 $H(1+2)$ 和 $H(3+4)$ 串联后进行哈希计算得到。

针对第 n 个区块，假设检测出交易记录 $Tx(3)$ 为需保留的交易记录，其余的交易记录为不需要保留的交易记录，则可以选取用于证明 $Tx(3)$ 属于该区块的节点。如图 3 所示，框选的节点即为除根节点以外的需筛选出的节点，即除了根节点以外，还需要筛选节点 $H(1+2)$ 、 $H(3+4)$ 、 $H(3)$ 和 $H(4)$ ，来构建节点保留树。又比如，假设检测出交易记录 $Tx(2)$ 和 $Tx(3)$ 为需保留的交易记录，那么就可以选取分别用于证明 $Tx(2)$ 和 $Tx(3)$ 属于该区块的节点。可以理解，针对 $Tx(2)$ 选取的节点必然也包括 $H(1+2)$ 和 $H(3+4)$ ，那么，就需要将针对 $Tx(2)$ 和 $Tx(3)$ 选取的节点进行汇总去重，去重后相当于筛选整个哈希树中所有节点构建节点保留树。然后，仅将 $Tx(2)$ 和 $Tx(3)$ 存储于各自相应的叶节点中。

S208，存储需保留的交易记录至节点保留树的相应叶节点中。

可选的，针对需保留的交易记录，区块链节点可以将其存储于节点保留树中相应的叶节点所对应的数据块中。

同样以图 3 为例，在节点保留树中，区块链节点可以将交易记录 $Tx(3)$ 存储于相应叶节点 $H(3)$ 所对应的数据块中。

S210，将区块的区块头和存储了交易记录的节点保留树关联更新至第二区块文件中；区块头在第二区块文件中相邻于前一区块的区块头。

可以理解，第二区块文件不同于第一区块文件，是在区块链节点上新建的区块文件。

可选的，区块链节点可以对区块的区块头和存储了交易记录的节点保留树

进行序列化处理，将区块的区块头和存储了交易记录的节点保留树转化为可存储的形式。区块链节点可以在第二区块文件中，确定该区块的前一区块的区块头。将序列化后的该区块的区块头相邻于前一区块的区块头更新至第二区块文件中，并在第二区块文件中，将序列化后的存储有需保留的交易记录的节点保留树与该区块的区块头关联存储。

在一个实施例中，该方法还包括：获取第二区块文件的第二存储位置；将所述区块的位置索引由指向所述第一区块文件的第一存储位置修改为指向所述第二存储位置。

其中，区块的位置索引，是指用于查找该区块的存储位置的索引。

可以理解，区块的位置索引初始是指向第一区块文件的第一存储位置，区块链节点在将区块按照步骤 S204~210 更新至第二区块文件后，则可以将该区块的位置索引修改为指向第二区块的第二存储位置。这样一来，后续就可以根据该指向第二存储位置的位置索引，从第二区块文件中查找到该区块。

需要说明的是，由于区块链节点之间具有共识机制，所以在达成共识后，各区块链节点都会将区块的区块头和存储了交易记录的节点保留树关联更新至第二区块文件中。

S212，在对每个区块更新处理完毕后，删除第一区块文件。

可以理解，针对第一区块文件中存在需保留的交易记录的区块，可以通过执行步骤 S204~S210，将该区块的区块头相邻于前一区块的区块头，与存储有需保留的交易记录的节点保留树关联更新至第二区块文件中。针对不存在需保留的交易记录的区块，可以于第二区块文件中在该前一区块的区块头之后更新存储该区块的区块头。第二区块文件中存储的每个区块的内容，即相当于该区块更新处理后的内容，那么，在针对第一区块文件中的每个区块都执行更新处理完毕之后，则可以得到包括更新后的区块的第二区块文件，然后就可以删除第一区块文件。

需要说明的是，由于区块链节点之间具有共识机制，所以在达成共识后，各区块链节点都会删除第一区块文件。

上述实施例中，由于区块头并未发生改变而且区块头仍然依次相邻，所以维持了区块链上通过哈希值首尾相连的特性，这样一来，根据保存在节点保留树的叶节点的交易记录，以及节点保留树中的节点，就可以证明该交易记录属于原来所对应的区块，相当于仍然保留了原始哈希树校验数据真实性的功能，

第二区块文件中保留的交易记录也就仍然无法伪造和篡改。在针对每个区块的更新处理完毕后，删除第一区块文件。由于第二区块文件中并未存储原有区块中不需要保留的交易记录，而仅保存了需保留的交易记录和用于校验该交易记录真实性的节点保留树，所以相较于第一区块文件大大节省了存储资源。

在一个实施例中，交易记录是用于对账的交易记录。步骤 S202 包括：针对第一区块文件中的每个区块，将区块中的交易记录与未成功对账的集合中的交易记录进行比对；响应于区块中的交易记录包括于未成功对账的集合，则判定交易记录为需保留的交易记录。

其中，未成功对账的集合，是由没有对账成功的各个交易记录形成的集合。

在一个实施例中，未成功对账的集合，可以是一个既包括未对账的交易记录又包括对账失败的交易记录的集合。

在一个实施例中，未成功对账的集合可以包括未对账集合或对账差错集合。

在一个实施例中，未成功对账集合可以包括未对账集合和对账差错集合这两个子集合。

其中，未对账集合，是尚未进行对账处理的交易记录所形成的集合。对账差错集合，是对账不一致（即对账出现错误）的交易记录所形成的集合。

可选的，区块链节点可以将区块中的每条交易记录，分别与未成功对账的集合中的交易记录进行比对，以判断在未成功对账的集合中是否包括该区块中的交易记录。响应于区块中的交易记录与未成功对账的集合中的交易记录比对一致时，说明该交易记录包括于未成功对账的集合，则可以将该区块中的包括于未成功对账的集合的交易记录判定为需保留的交易记录。响应于区块中的交易记录与未成功对账的集合中的交易记录比对不一致时，说明该交易记录不存在于未成功对账的集合，则可以判定该交易记录为不需保留的交易记录。

上述实施例中，在区块链上设置了未成功对账的集合，将区块中的交易记录与未成功对账的集合中交易记录进行比对，能够更加快速地确定出需保留的交易记录。

在一个实施例中，未成功对账的集合包括未对账集合。该方法还包括对账步骤，该对账步骤可以包括以下步骤：

S402，响应于接收到第一对账设备发送的对账调用请求，通过智能合约将

对账调用请求中携带的待对账的第一交易记录存储至第一区块文件的区块中。

其中，未对账集合，是尚未进行对账处理的交易记录所形成的集合。第一对账设备，是发起对账处理的一方所使用的设备。对账调用请求，是用于请求调用区块链上的智能合约，以实现了对账处理。

智能合约，是对由各区块链节点共同参与制定的、且用于明确权利和义务的协议进行编程得到的一段设置于区块链上的代码。该代码中包含会触发合约自动执行的条件。可以理解，在达到对某一事件的触发条件时，等各区块链节点对该事件达成共识后，智能合约可以自动地执行该事件。

可以理解，区块链节点预先建立了未对账集合，用于保存还未对账的交易记录。初始创建的未对账集合可以是空集合，里面并不包括交易记录。

可选的，对账双方在进行对账时，可以通过各自的对账设备向区块链节点发送对账调用请求，以请求调用区块链节点上设置的智能合约。响应于区块链节点接收到第一对账设备发送的对账调用请求，则可以通过智能合约解析该对账调用请求，从中获取待对账的第一交易记录。区块链节点可以通过智能合约将该第一交易记录存储至第一区块文件的区块中。

可以理解，在接收到对账双方任一方所使用的对账设备发送的对账调用请求时，区块链节点都可以通过智能合约将该对账调用请求中携带的交易记录存储至第一区块文件的区块中。

S404，通过智能合约获取第一交易记录对应的对账关联标识。

其中，对账关联标识，用于将由对账双方所使用的对账设备分别发送的、且表示同一笔交易的交易记录进行关联标示。

在一个实施例中，对账关联标识可以包括交易流水号。那么，对账双方发送的对应于同一交易流水号的交易记录之间即需要进行对账处理。可以理解，在其他实施例中，对账关联标识也可以是其他能够唯一表示同一笔交易的标识。

可以理解，区块链节点可以通过智能合约从第一交易记录中提取对账关联标识。区块链节点也可以通过智能合约从对账调用请求中提取第一交易记录对应的对账关联标识。

S406，从区块链上创建的未对账集合中，查找与对账关联标识对应的第二对账设备的第二交易记录。

其中，第二对账设备，是发起对账处理的另一方所使用的设备。

具体地，区块链节点可以通过智能合约，从未对账集合的交易记录中，查

找是否存在与该对账关联标识对应的第二交易记录。该第二交易记录即为由发起对账处理的另一方所使用的第二对账设备发送对账调用请求时携带的交易记录。

S408, 响应于未找到第二交易记录, 将第一交易记录添加至未对账集合中。

可以理解, 由于对账双方在调用智能合约进行对账时, 并非完全一致, 必然会有快有慢, 而且选取对账的交易记录的顺序也可能会有不同, 当一个对账方发送需要对账的交易记录至智能合约时, 另一方可能尚未发送对应于同一对账关联标识的交易记录至智能合约, 所以就可能存在未查找到与对账关联标识对应的第二对账设备的第二交易记录的情况。同样地, 如果另一方先发送需要对账的交易记录至智能合约, 该先发送的交易记录也无法查找到对应于同一对账关联标识的交易记录, 因而也无法对账, 因此, 先发送的交易记录就会被添加至未对账集合。

因此, 从未对账集合中, 区块链节点可能查找到与对账关联标识对应的第二对账设备的第二交易记录, 也可能未查找到与对账关联标识对应的第二对账设备的第二交易记录。

响应于未找到第二交易记录, 区块链节点可以将第一交易记录添加至未对账集合中。

在一个实施例中, 响应于查找到与对账关联标识对应的第二对账设备的第二交易记录, 区块链节点则可以将该第二交易记录从未对账集合中删除, 并将第一交易记录和第二交易记录进行比对。响应于比对不一致时, 则可以将第一交易记录和第二交易记录与对账成功的交易记录进行区别处理。区别处理, 是指将第一交易记录和第二交易记录与对账成功的交易记录区分开的处理。

在一个实施例中, 区块链节点可以将比对不一致的第一交易记录和第二交易记录添加对账失败标记。可以理解, 区块链节点也可以采用其他方式将比对不一致的第一交易记录和第二交易记录与对账成功的交易记录进行区别处理。

上述实施例中, 在区块链上设置了未对账集合, 在接收到对账调用请求时, 结合未对账集合来实现实时对账处理, 使得对账更加的及时、便捷。

如图 4 所示, 在一个实施例中, 未成功对账的集合还包括对账差错集合。该对账步骤除了上述步骤 S402~408 以外, 还包括以下步骤:

S410, 响应于找到第二交易记录时, 从未对账集合中移除第二交易记录。

S412, 比对第一交易记录和第二交易记录。

可选的, 区块链节点响应于在未对账集合中查找到第二交易记录, 则从未对账集合中删除第二交易记录, 并将第二交易记录和第一交易记录进行比对, 以比对二者是否一致。

S414, 响应于比对不一致, 将第一交易记录和第二交易记录添加至对账差错集合。

其中, 对账差错集合, 是对账不一致的交易记录所形成的集合。

可以理解, 将第二交易记录和第一交易记录进行比对的结果, 包括比对一致和比对不一致两种情况。响应于比对一致, 则判定第一交易记录和第二交易记录属于对账成功的交易记录。响应于比对不一致, 说明第一交易记录和第二交易记录属于对账失败的交易记录, 则可以将第一交易记录和第二交易记录添加至对账差错集合中。

在一个实施例中, 区块链节点可以接收针对对账差错集合中的交易记录的修改处理, 并将修改后的交易记录再次进行对账比对处理。

上述实施例中, 在区块链上设置两种类型的未成功对账的集合, 即未对账集合和对账差错集合, 通过这两种集合来实现实时对账非常的便捷, 而且, 能够更加准确地管理对账结果。

在一个实施例中, 步骤 S204 包括: 在区块的哈希树中, 确定与交易记录相应的叶节点; 从哈希树中, 筛选叶节点的第一祖先节点和兄弟叶节点; 针对叶节点的非兄弟叶节点, 筛选非兄弟叶节点的最接近根节点的第二祖先节点; 根据叶节点、第一祖先节点、兄弟叶节点和第二祖先节点, 构建节点保留树。

其中, 与交易记录相应的叶节点, 是指以需保留的交易记录的哈希值作为标签的叶子节点。祖先节点, 是指与叶子节点具有祖先关系的节点。可以理解, 父节点、祖父节点、曾祖父节点等这些都属于叶节点的祖先节点。兄弟叶节点, 是指与叶节点对应相同父节点的叶子节点。非兄弟叶节点, 是指与叶节点没有相同父节点的叶子节点。

可选的, 区块链节点可以从哈希树中, 筛选需保留的交易记录所对应叶节点的第一祖先节点和该叶节点的兄弟叶节点。可以理解, 第一祖先节点, 是指叶节点的全部祖先节点。因为根节点必然是需保留的交易记录所对应叶节点的祖先节点, 所以, 第一祖先节点中包括根节点。区块链节点可以从哈希树中确

定交易记录所对应叶节点的非兄弟叶节点，确定该非兄弟叶节点最接近根节点的那个祖先节点，得到第二祖先节点。可以理解，非兄弟叶节点可以为一个或多个，当多个非兄弟叶节点的最接近根节点的祖先节点不同时，则筛选出每个最接近根节点的祖先节点，当多个非兄弟叶节点的最接近根节点的祖先节点相同时，则可以仅筛选出最接近根节点的那个祖先节点。

区块链节点可以根据叶节点、第一祖先节点、兄弟叶节点和第二祖先节点，构建节点保留树。

现结合图 3 进行解释说明。比如，需保留的交易记录为 Tx(3)，其所对应的叶节点即为 H (3)，H (3) 的兄弟叶节点为 H (4)，H (3) 的第一祖先节点为 H (3+4) 和根节点。H (1) 和 H (2) 是 H (3) 的非兄弟叶节点，非兄弟叶节点 H (1) 和 H (2) 的最接近根节点的第二祖先节点都为 H (1+2)。所以，区块链节点可以根据 Tx(3) 所对应的叶节点 H (3)、H (4)、H (3+4)、H (1+2) 以及根节点，构建节点保留树。

上述实施例中，根据该需保留的交易记录在原始哈希树对应的叶节点、第一祖先节点、兄弟叶节点和第二祖先节点，构建节点保留树，在保留原始哈希树校验数据真实性的功能的前提下，能够剔除不必要的节点和交易记录数据，一定程度上节省了存储资源。

如图 5 所示，在一个实施例中，提供了一种数据处理方法，该数据处理方法具体包括以下步骤：

S502，在区块链上分别创建一个未对账集合和对账差错集合。

需要说明的是，初始创建的未对账集合和对账差错集合可以是空集合，里面并不包括交易记录，后续根据实际的对账情况再加入相应交易记录。

可以理解，未对账集合和对账差错集合相当于区块链的状态数据。

S504，响应于接收第一对账设备针对智能合约发送的对账调用请求，通过智能合约将对账调用请求中携带的待对账的第一交易记录存储至第一区块文件的区块中。

S506，通过智能合约获取第一交易记录对应的对账关联标识；从区块链上创建的未对账集合中，查找与对账关联标识对应的第二对账设备的第二交易记录。响应于未查找到第二交易记录，执行步骤 S508，响应于查找到第二交易记录，执行步骤 S510。

S508, 将第一交易记录添加至未对账集合中。

其中, 对账关联标识, 用于将由对账双方所使用的对账设备分别发送的、且表示同一笔交易的交易记录进行关联标示。

可以理解, 由于区块链是从第一区块开始, 依次往后生成区块以存储交易记录, 所以, S506 中的未对账集合中所包含的交易记录, 即为截止到第一交易记录存储至区块为止, 未进行对账的交易记录。比如, 第一区块文件中前 4 个区块已经存满了, 所以将第一交易记录 R1 存储至第 5 个区块, 那么, 未对账集合中包含的就是截止到第 5 个区块为止当前尚未进行对账的交易记录。区块链节点可以将当前的未对账集合中查找是否存在与第一交易记录 R1 对应于同一对账关联标识的第二交易记录 R2, 如果不存在, 则可以将第一交易记录 R1 也添加至未对账集合中。

S510, 从未对账集合中移除第二交易记录; 比对第一交易记录和第二交易记录; 响应于比对不一致, 将第一交易记录和第二交易记录添加至对账差错集合。

同样地, 由于区块链是从第一区块开始, 依次往后生成区块以存储交易记录, 所以, 添加第一交易记录和第二交易记录后的对账差错集合中所包含的交易记录, 即为截止到第一交易记录存储至区块为止, 对账出现错误的交易记录。

可以理解, 步骤 S504~ S510 描述的是一个实时对账的过程, 即, 区块链节点可以一边将第一对账设备的对账调用请求中携带的第一交易记录存储至区块, 一边对该第一交易记录进行对账处理, 从而实现边存储边对账的实时对账处理。

需要说明的是, 这里的“第一对账设备”和“第二对账设备”仅用于区分, 并不用做其他限定。可以理解, 智能合约在接收到任一方的对账调用请求后, 都可以将其中携带的交易记录存储至区块中。

此外, 由于对账双方在调用智能合约进行对账时, 并非完全一致, 必然会有快有慢, 而且选取对账的交易记录的顺序也可能会有不同, 当一个对账方通过对账设备发送需要对账的交易记录至智能合约, 而另一个对账方尚未发送对账数据过来时, 则可以将该需要对账的交易记录添加至未对账集合中。

可以理解, 当一个对账方发送需要对账的交易记录至智能合约, 另一个对账方也已经发送过来对应于同一笔交易的交易记录时, 则可以将对应于同一笔交易的不同交易记录进行比对, 如果不一致, 则说明对账出错, 则可以将比对

不一致的这些交易记录添加至对账差错集合。

S512, 针对第一区块文件中的每个区块, 将区块中的交易记录与对账差错集合和未对账集合中的交易记录进行比对。响应于区块中的交易记录包括于对账差错集合或未对账集合, 执行步骤 S514~ S516, 响应于区块中的交易记录不包括于对账差错集合和未对账集合, 则执行步骤 S518。

可以理解, 针对第一区块文件中的每个区块, 区块链节点可以按照预设预设周期执行步骤 S512 (相当于, 定期清理区块链中的数据)。在其他实施例中, 区块链节点也可以在第一区块文件中的区块数量达到预设数量阈值时, 触发执行步骤 S512 (相当于, 当区块链中的区块数量过多、占用的存储空间过大时, 再清理区块链中的数据)。

S514, 在区块的哈希树中, 确定与交易记录相应的叶节点; 从哈希树中, 筛选叶节点的第一祖先节点和兄弟叶节点; 针对叶节点的非兄弟叶节点, 筛选非兄弟叶节点的最接近根节点的第二祖先节点; 根据叶节点、兄弟叶节点、第一祖先节点和第二祖先节点, 构建节点保留树。

S516, 存储交易记录至节点保留树的相应叶节点中; 将区块的区块头和存储了交易记录的节点保留树关联更新至第二区块文件中; 区块头在第二区块文件中相邻于前一区块的区块头。

S518, 将区块的区块头相邻于前一区块的区块头更新至第二区块文件中。

S520, 将区块的位置索引由指向第一区块文件的第一存储位置修改为指向第二区块文件的第二存储位置。

S522, 在针对每个区块的更新处理完毕后, 删除第一区块文件。

上述实施例中, 在第二区块文件中, 由于区块头并未发生改变而且区块头仍然依次相邻, 所以维持了区块链上通过哈希值首尾相连的特性, 这样一来, 根据保存在节点保留树的叶节点的交易记录, 以及节点保留树中的节点, 就可以证明该交易记录属于原来所对应的区块, 相当于仍然保留了原始哈希树校验数据真实性的功能, 第二区块文件中保留的交易记录也就仍然无法伪造和篡改。在针对每个区块的更新处理完毕后, 删除第一区块文件。由于第二区块文件中并未存储原有区块中不需要保留的交易记录, 而仅保存了需保留的交易记录和用于校验该交易记录真实性的节点保留树, 所以相较于第一区块文件大大节省了存储资源。

如图 6 所示, 在一个实施例中, 提供了一种数据处理装置 600, 该装置可以通过硬件或者软硬结合的方式实现为计算机设备的全部或者部分, 可选的, 该计算机设备可以是图 1 中的区块链节点。该装置 600 包括: 节点保留模块 602、交易存储模块 604、区块更新模块 606 以及删除模块 608, 其中:

节点保留模块 602, 用于针对第一区块文件中的每个区块, 检测区块中需保留的交易记录; 从区块的哈希树中, 筛选用于证明交易记录属于区块的节点; 根据筛选出的节点构建节点保留树。

交易存储模块 604, 用于存储交易记录至节点保留树的相应叶节点中。

区块更新模块 606, 用于将区块的区块头和存储了交易记录的节点保留树关联更新至第二区块文件中; 该区块的区块头在第二区块文件中相邻于前一区块的区块头。

删除模块 608, 用于在针对每个区块的更新处理完毕后, 删除第一区块文件。

在一个实施例中, 节点保留模块 602, 用于响应于检测出区块中需保留的交易记录, 执行从区块的哈希树中, 筛选用于证明交易记录属于区块的节点的步骤; 节点保留模块 620, 还用于响应于检测到区块中不存在需保留的交易记录, 则通知区块更新模块 606 将区块的区块头相邻于前一区块的区块头更新至第二区块文件中。

在一个实施例中, 交易记录是用于对账的交易记录; 节点保留模块 602 还用于将区块中的交易记录与未成功对账的集合中的交易记录进行比对; 响应于区块中的交易记录包括于未成功对账的集合, 判定交易记录为需保留的交易记录。

如图 7 所示, 在一个实施例中, 未成功对账的集合包括未对账集合。该装置 600 还包括:

实时对账模块 601, 用于响应于接收到第一对账设备发送的对账调用请求, 通过智能合约将待对账的第一交易记录存储至第一区块文件的区块中; 该第一交易记录携带于对账调用请求中; 通过智能合约获取第一交易记录对应的对账关联标识; 从区块链上创建的未对账集合中, 查找与对账关联标识对应的第二对账设备的第二交易记录; 响应于未找到第二交易记录, 将第一交易记录添加至未对账集合中。

在一个实施例中, 未成功对账的集合还包括对账差错集合。实时对账模块 601 还用于响应于找到第二交易记录, 从未对账集合中移除第二交易记录; 比对

第一交易记录和第二交易记录；响应于比对不一致，将第一交易记录和第二交易记录添加至对账差错集合。

在一个实施例中，节点保留模块 602 还用于在区块的哈希树中，确定与交易记录相应的叶节点；从哈希树中，筛选叶节点的第一祖先节点和兄弟叶节点；针对叶节点的非兄弟叶节点，筛选非兄弟叶节点的最接近根节点的第二祖先节点；根据叶节点、第一祖先节点、兄弟叶节点和第二祖先节点，构建节点保留树。

在一个实施例中，哈希树为双层哈希树，双层哈希树的根节点的哈希值，由每个叶节点的哈希值串联合并后进行哈希运算得到；筛选的节点即为哈希树的全部节点。

在一个实施例中，区块更新模块 606 还用于获取第二区块文件的第二存储位置；将区块的位置索引由指向第一区块文件的第一存储位置修改为指向第二存储位置。

图 8 为一个实施例中计算机设备的内部结构示意图。参照图 8，该计算机设备可以是图 1 中所示的区块链节点 130 中的各个服务器。该计算机设备包括通过系统总线连接的处理器、存储器和网络接口。其中，存储器包括非易失性存储介质和内存存储器。该计算机设备的非易失性存储介质可存储操作系统和计算机程序。该计算机程序被执行时，可使得处理器执行上述各个方法实施例所示的数据处理方法的全部或者部分步骤。该计算机设备的处理器用于提供计算和控制能力，支撑整个计算机设备的运行。该内存存储器中可储存有计算机程序，该计算机程序被处理器执行时，可使得处理器执行上述各个方法实施例所示的数据处理方法的全部或者部分步骤。计算机设备的网络接口用于进行网络通信。

本领域技术人员可以理解，图 8 中示出的结构，仅仅是与本申请方案相关的部分结构的框图，并不构成对本申请方案所应用于其上的计算机设备的限定，具体的计算机设备可以包括比图中所示更多或更少的部件，或者组合某些部件，或者具有不同的部件布置。

在一个实施例中，本申请提供的数据处理装置可以实现为一种计算机程序的形式，计算机程序可在如图 8 所示的计算机设备上运行，计算机设备的非易失性存储介质可存储组成该数据处理装置的各个程序模块，比如，图 6 所示的节点保留模块 602、交易存储模块 604、区块更新模块 606 以及删除模块 608。

各个程序模块所组成的计算机程序用于使该计算机设备执行本说明书中描述的本申请各个实施例的数据处理方法中的步骤，例如，计算机设备可以通过如图6所示的数据处理装置600中的节点保留模块602针对第一区块文件中的每个区块，检测区块中需保留的交易记录；从区块的哈希树中，筛选用于证明交易记录属于区块的节点；根据筛选出的节点构建节点保留树。计算机设备可以通过交易存储模块604存储交易记录至节点保留树的相应叶节点中。计算机设备可以通过区块更新模块606将区块的区块头和存储了交易记录的节点保留树关联更新至第二区块文件中；该区块的区块头在第二区块文件中相邻于前一区块的区块头。计算机设备可以通过删除模块608在针对每个区块的更新处理完毕后，删除第一区块文件。

在一个实施例中，提供了一种计算机设备，包括存储器和处理器，存储器存储有计算机程序，计算机程序被处理器执行时，使得处理器执行上述数据处理方法的步骤。此处数据处理方法的步骤可以是上述各个实施例的数据处理方法中的步骤。

在一个实施例中，提供了一种计算机可读存储介质，存储有计算机程序，计算机程序被处理器执行时，使得处理器执行上述数据处理方法的步骤。此处数据处理方法的步骤可以是上述各个实施例的数据处理方法中的步骤。

需要说明的是，本申请各实施例中的“第一”和“第二”仅用作区分，而并不用于大小、先后、从属等方面的限定。

应该理解的是，虽然本申请各实施例中的各个步骤并不是必然按照步骤标号指示的顺序依次执行。除非本文中有明确的说明，这些步骤的执行并没有严格的顺序限制，这些步骤可以以其它的顺序执行。而且，各实施例中至少一部分步骤可以包括多个子步骤或者多个阶段，这些子步骤或者阶段并不必然是在同一时刻执行完成，而是可以在不同的时刻执行，这些子步骤或者阶段的执行顺序也不必然是依次进行，而是可以与其它步骤或者其它步骤的子步骤或者阶段的至少一部分轮流或者交替地执行。

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非易失性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，本申请所提供的各实施例中所使用的对存储器、存储、数据

库或其它介质的任何引用，均可包括非易失性和/或易失性存储器。非易失性存储器可包括只读存储器（ROM）、可编程 ROM（PROM）、电可编程 ROM（EPROM）、电可擦除可编程 ROM（EEPROM）或闪存。易失性存储器可包括随机存取存储器（RAM）或者外部高速缓冲存储器。作为说明而非局限，RAM 以多种形式可得，诸如静态 RAM（SRAM）、动态 RAM（DRAM）、同步 DRAM（SDRAM）、双数据率 SDRAM（DDRSDRAM）、增强型 SDRAM（ESDRAM）、同步链路（Synchlink） DRAM（SLDRAM）、存储器总线（Rambus）直接 RAM（RDRAM）、直接存储器总线动态 RAM（DRDRAM）、以及存储器总线动态 RAM（RDRAM）等。

以上所述实施例的各技术特征可以进行任意的组合，为使描述简洁，未对上述实施例中的各个技术特征所有可能的组合都进行描述，然而，只要这些技术特征的组合不存在矛盾，都应当认为是本说明书记载的范围。

以上所述实施例仅表达了本申请的几种实施方式，其描述较为具体和详细，但并不能因此而理解为对发明专利范围的限制。应当指出的是，对于本领域的普通技术人员来说，在不脱离本申请构思的前提下，还可以做出若干变形和改进，这些都属于本申请的保护范围。因此，本申请专利的保护范围应以所附权利要求为准。