

(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局



(10) 国际公布号

WO 2020/258856 A1

(43) 国际公布日

2020 年 12 月 30 日 (30.12.2020)

(51) 国际专利分类号:

G06F 16/21 (2019.01)

G06F 16/27 (2019.01)

G06F 16/22 (2019.01)

(21) 国际申请号:

PCT/CN2020/072030

(22) 国际申请日:

2020 年 1 月 14 日 (14.01.2020)

(25) 申请语言:

中文

(26) 公布语言:

中文

(30) 优先权:

201910576961.X 2019年6月28日 (28.06.2019) CN

(71) 申请人: 创新先进技术有限公司 (ADVANCED NEW TECHNOLOGIES CO., LTD.) [—/CN]; 开曼群岛大开曼岛乔治镇医院路27号开曼企业中心, Grand Cayman KY1-9008 (KY)。

(72) 发明人: 陆钟豪 (LU, Zhonghao); 中国浙江省杭州市余杭区文一西路969号3号楼5楼阿里巴巴集团法务部, Zhejiang 311121 (CN)。俞本权 (YU, Benquan); 中国浙江省杭州市余杭区文一西路969号3号楼5楼阿里巴巴集团法务部, Zhejiang

311121 (CN)。卓海振 (ZHUO, Haizhen); 中国浙江省杭州市余杭区文一西路969号3号楼5楼阿里巴巴集团法务部, Zhejiang 311121 (CN)。

(74) 代理人: 北京博思佳知识产权代理有限公司 (BEIJING BESTIPR INTELLECTUAL PROPERTY LAW CORPORATION); 中国北京市海淀区上地三街9号嘉华大厦B座409, Beijing 100085 (CN)。

(81) 指定国 (除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW。

(54) Title: HIERARCHICAL STORAGE METHOD AND APPARATUS BASED ON BLOCKCHAIN, AND ELECTRONIC DEVICE

(54) 发明名称: 基于区块链的分级存储方法及装置、电子设备

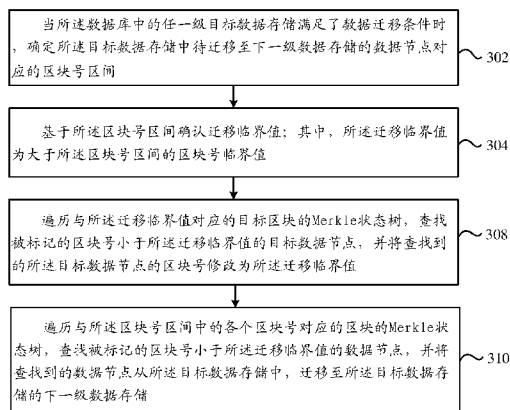


图 3

- 302 When target data storage of any level in a database meets a data migration condition, determine a block number interval corresponding to a data node to be migrated to the next-level data storage in the target data storage
- 304 Determine a migration critical value on the basis of the block number interval, wherein the migration critical value is a block number critical value greater than the block number interval
- 308 Traverse a Merkle state tree of a target block corresponding to the migration critical value, search for a target data node with a labeled block number less than the migration critical value, and modify the block number of the found target data node into the migration critical value
- 310 Traverse the Merkle state tree of a block corresponding to each block number in the block number interval, search for a labeled data node with a block number less than the migration critical value, and migrate the found data node to the next-level data storage of the target data storage from the target data storage

(57) Abstract: A hierarchical storage method based on a blockchain. A Merkle state tree composed of account state data of a blockchain is stored in a database; the database comprises multi-level data storage; and a data node on the Merkle state tree is labeled with a block number indicating a block where the data node is located when the data node has data update. The hierarchical storage method comprises: when data storage of any level meets a data migration condition, determining a block number interval corresponding to a data node to be migrated in the data storage; determining a migration critical value on the basis of the block number interval, wherein the migration critical value is a block number critical value greater than the block number interval; traversing a Merkle state tree of the block corresponding to the migration critical value, and modifying the block number of a labeled target data node with a block number less than the migration critical value into the migration critical value; and traversing the Merkle state tree of the block corresponding to each block number in the block number interval, and migrating a labeled data node with a block number less than the migration critical value to the next-level data storage from target data storage.

(84) 指定国 (除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:

— 包括国际检索报告 (条约第21条(3))。

(57) 摘要: 一种基于区块链的分级存储方法, 数据库中存储了由区块链的账户状态数据构成的Merkle状态树; 数据库包括多级数据存储; Merkle状态树上的数据节点被标记了指示数据节点发生数据更新时所在区块的区块号; 当任一级数据存储满足数据迁移条件时, 确定该数据存储中待迁移的数据节点对应的区块号区间; 基于区块号区间确定迁移临界值; 迁移临界值为大于区块号区间的区块号临界值; 遍历与迁移临界值对应的区块的Merkle状态树, 将被标记的区块号小于迁移临界值的目标数据节点的区块号修改为迁移临界值; 遍历与区块号区间中的各个区块号对应的区块的Merkle状态树, 将被标记的区块号小于迁移临界值的数据节点, 从目标数据存储中迁移至下一级数据存储。

基于区块链的分级存储方法及装置、电子设备

技术领域

[01] 本说明书一个或多个实施例涉及区块链技术领域，尤其涉及一种基于区块链的分级存储方法及装置、电子设备。

5 背景技术

[02] 区块链技术，也被称之为分布式账本技术，是一种由若干台计算设备共同参与“记账”，共同维护一份完整的分布式数据库的新兴技术。由于区块链技术具有去中心化、公开透明、每台计算设备可以参与数据库记录、并且各计算设备之间可以快速的进行数据同步的特性，使得区块链技术已在众多的领域中广泛的进行应用。

10 发明内容

[03] 本说明书提出一种基于区块链的分级存储方法，所述区块链的账户状态数据被组织成 Merkle 状态树在数据库中存储；所述数据库包括多级数据存储；所述 Merkle 状态树上的数据节点被标记了区块号；所述区块号指示所述数据节点发生数据更新时所在的区块；所述方法包括：

15 [04] 当所述数据库中的任一级目标数据存储满足了数据迁移条件时，确定所述目标数据存储中待迁移至下一级数据存储的数据节点对应的区块号区间；

[05] 基于所述区块号区间确定迁移临界值；其中，所述迁移临界值为大于所述区块号区间的区块号临界值；

20 [06] 遍历与所述迁移临界值对应的目标区块的 Merkle 状态树，查找被标记的区块号小于所述迁移临界值的目标数据节点，并将查找到的所述目标数据节点的区块号修改为所述迁移临界值；

[07] 遍历与所述区块号区间中的各个区块号对应的区块的 Merkle 状态树，查找被标记的区块号小于所述迁移临界值的数据节点，并将查找到的数据节点从所述目标数据存储中，迁移至所述目标数据存储的下一级数据存储。

25 [08] 可选的，所述基于所述区块号区间确定迁移临界值，包括：

[09] 如果所述区块号区间的右区间为开区间，将所述区块号区间的右端点值，确定为

所述迁移临界值;

[10] 如果所述区块号区间的右区间为闭区间, 将所述区块号区间的右端点值与所述区块链的区块号递增步长的和确定为所述迁移临界值。

[11] 可选的, 所述数据库为 Key-Value 数据库;

- 5 [12] 所述 Merkle 状态树上的数据节点以 Key-Value 键值对的形式存储在所述数据库中; 其中, 所述 Key-Value 键值对的 key 为所述数据节点包含的数据内容的 hash 值; 所述 Key-Value 键值对的 value 为所述数据节点包含的数据内容。

[13] 可选的, 所述方法还包括:

- [14] 确定最新区块的 Merkle 状态树上发生数据更新的数据节点; 为所述最新区块的
10 Merkle 状态树上发生数据更新的数据节点标记所述最新区块的区块号。

[15] 可选的, 所述数据库中存储的所述 Merkle 状态树上的数据节点的 Value 中, 包含预设的区块号字段; 为所述 Merkle 状态树上的数据节点标记的区块号, 添加在所述区块号字段中。

- [16] 可选的, 所述数据库中预设了指向所述 Merkle 状态树上的数据节点的 Value 的存
15 储位置的区块号字段; 为所述 Merkle 状态树上的数据节点标记的区块号, 添加在所述区块号字段中。

[17] 可选的, 所述 Merkle 树为融合了 Trie 字典树的树形结构的 Merkle 树变种。

[18] 可选的, 所述 Merkle 状态树为 Merkle Patricia Tree 状态树。

[19] 可选的, 所述数据库为 LevelDB 数据库; 或者基于 LevelDB 架构的数据库;

- 20 [20] 可选的, 所述数据库为基于 LevelDB 架构的 Rocksdb 数据库。

[21] 可选的, 所述多级数据存储对应的存储介质的读写性能, 存在性能差异; 其中, 级数高的数据存储对应的存储介质的读写性能, 高于级数低的数据存储对应的存储介质的读写性能。

- [22] 本说明书还提出一种基于区块链的分级存储装置, 所述区块链的账户状态数据被
25 组织成 Merkle 状态树在数据库中存储; 所述数据库包括多级数据存储; 所述 Merkle 状态树上的数据节点被标记了区块号; 所述区块号指示所述数据节点发生数据更新时所在的区块; 所述装置包括:

[23] 确定模块，当所述数据库中的任一级目标数据存储满足了数据迁移条件时，确定所述目标数据存储中待迁移至下一级数据存储的数据节点对应的区块号区间；基于所述区块号区间确认迁移临界值；其中，所述迁移临界值为大于所述区块号区间的区块号临界值；

5 [24] 修改模块，遍历与所述迁移临界值对应的目标区块的 Merkle 状态树，查找被标记的区块号小于所述迁移临界值的目标数据节点，并将查找到的所述目标数据节点的区块号修改为所述迁移临界值；

[25] 迁移模块，遍历与所述区块号区间中的各个区块号对应的区块的 Merkle 状态树，查找被标记的区块号小于所述迁移临界值的数据节点，并将查找到的数据节点从所述目
10 标数据存储中，迁移至所述目标数据存储的下一级数据存储。

[26] 可选的，所述确定模块：

[27] 如果所述区块号区间的右区间为开区间，将所述区块号区间的右端点值，确定为所述迁移临界值；

[28] 如果所述区块号区间的右区间为闭区间，将所述区块号区间的右端点值与所述区
15 块链的区块号递增步长的和确定为所述迁移临界值。

[29] 可选的，所述数据库为 Key-Value 数据库；

[30] 所述 Merkle 状态树上的数据节点以 Key-Value 键值对的形式存储在所述数据库中；其中，所述 Key-Value 键值对的 key 为所述数据节点包含的数据内容的 hash 值；所述 Key-Value 键值对的 value 为所述数据节点包含的数据内容。

20 [31] 可选的，所述装置还包括：

[32] 标记模块，确定最新区块的 Merkle 状态树上发生数据更新的数据节点；为所述最新区块的 Merkle 状态树上发生数据更新的数据节点标记所述最新区块的区块号。

[33] 可选的，所述数据库中存储的所述 Merkle 状态树上的数据节点的 Value 中，包含预设的区块号字段；为所述 Merkle 状态树上的数据节点标记的区块号，添加在所述区
25 块号字段中。

[34] 可选的，所述数据库中预设了指向所述 Merkle 状态树上的数据节点的 Value 的存储位置的区块号字段；为所述 Merkle 状态树上的数据节点标记的区块号，添加在所述区块号字段中。

[35] 可选的，所述 Merkle 树为融合了 Trie 字典树的树形结构的 Merkle 树变种。

[36] 可选的，所述 Merkle 状态树为 Merkle Patricia Tree 状态树。

[37] 可选的，所述数据库为 LevelDB 数据库；或者基于 LevelDB 架构的数据库；

[38] 可选的，所述数据库为基于 LevelDB 架构的 Rocksdb 数据库。

- 5 [39] 可选的，所述多级数据存储对应的存储介质的读写性能，存在性能差异；其中，级数高的数据存储对应的存储介质的读写性能，高于级数低的数据存储对应的存储介质的读写性能。在以上技术方案中，可以实现对数据库中存储的 Merkle 状态树上的数据节点进行剪枝，将记录了历史状态数据的数据节点从 Merkle 状态树上移除，迁移至下一级数据存储，并在本级数据存储中继续存储和保留记录了最新状态数据的数据节点，
- 10 进而完成针对数据库中存储的 Merkle 状态树的分级存储。

附图说明

[40] 图 1 是一示例性实施例提供的一种将区块链的账户状态数据组织成 MPT 状态树的示意图；

[41] 图 2 是一示例性实施例提供的一种 MPT 状态树上的 node 复用的示意图；

- 15 [42] 图 3 是一示例性实施例提供的一种基于区块链的分级存储方法的流程图；

[43] 图 4 是一示例性实施例提供的一种电子设备的结构示意图；

[44] 图 5 是一示例性实施例提供的一种基于区块链的分级存储装置的框图。

具体实施方式

- 20 [45] 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述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。

- 25 [46] 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。

[47] 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。

[48] 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合于特定机构内部使用。

[49] 基于区块链的基本特性，区块链通常是由若干个区块构成。在这些区块中分别记录有与该区块的创建时刻对应的时间戳，所有的区块严格按照区块中记录的时间戳，构成一条在时间上有序的数据链条。

[50] 对于物理世界产生的真实数据，可以将其构建成区块链所支持的标准的交易（transaction）格式，然后发布至区块链，由区块链中的节点设备进行共识，并在达成共识后，由区块链中作为记账节点的节点设备，将这笔交易打包进区块，在区块链中进行持久化存证。

[51] 在区块链领域，有一个重要的概念就是账户（Account）；以以太坊为例，以太坊通常将账户划分为外部账户和合约账户两类；外部账户就是由用户直接控制的账户；而合约账户则是由用户通过外部账户创建的，包含合约代码的账户（即智能合约）。

[52] 当然，对于一些基于以太坊的架构而衍生出的区块链项目（比如蚂蚁区块链），还可以对区块链支持的账户类型，进行进一步的扩展，在本说明书中不进行特别限定。

[53] 对于区块链中的账户而言，通常会通过一个结构体，来维护账户的账户状态。当区块中的交易被执行后，区块链中与该交易相关的账户的状态通常也会发生变化。

[54] 以以太坊为例，账户的结构体通常包括 Balance, Nonce, Code 和 storage 等字段。其中：

[55] Balance 字段，用于维护账户目前的账户余额；

[56] Nonce 字段, 用于该账户的交易次数; 它是用于保障每笔交易能且只能被处理一次的计数器, 有效避免重放攻击。

[57] code 字段, 用于维护该账户的合约代码; 在实际应用中, code 字段中通常仅维护合约代码的 hash 值; 因而, code 字段通常也称之为 codehash 字段。对于外部账户而言, 该字段为空值。

[58] storage 字段, 用于维护该账户的存储 (默认为空)。在实际应用中, storage 字段仅维护基于账户的存储内容构建的 MPT (Merkle Patricia Trie) 树的根节点; 因此, storage 字段通常也称之为 storageRoot 字段。

[59] 其中, 对于外部账户而言, 以上示出的 code 字段和 storage 字段为空值。

10 [60] 而大多数区块链项目, 通常都会使用 Merkle 树; 或者, 基于 Merkle 树的数据结构, 来存储和维护数据。以以太坊为例, 以太坊使用了 MPT 树 (一种 Merkle 树变种), 作为数据组织形式, 用来组织和管理账户状态、交易信息等重要数据。

[61] 以太坊针对区块链中需要存储和维护的数据, 设计了三颗 MPT 树, 分别是 MPT 状态树、MPT 交易树和 MPT 收据树。

15 [62] MPT 状态树, 是区块链中所有账户的账户状态数据 (state), 组织成的 MPT 树; MPT 交易树是区块中的交易数据 (transaction), 组织成的 MPT 树; MPT 收据树, 是区块中的交易执行完毕后生成的与每笔交易对应的交易收据 (receipt), 组织成的 MPT 树。以上示出的 MPT 状态树、MPT 交易树和 MPT 收据树的根节点的 hash 值, 都会被添加至区块头中。

20 [63] 其中, MPT 交易树和 MPT 收据树, 与区块相对应, 每一个区块都有自己的 MPT 交易树和 MPT 收据树。而 MPT 状态树是一个全局的 MPT 树, 并不与某一个特定的区块相对应, 而是涵盖了区块链中所有账户的账户状态数据。

[64] 对于组织成的 MPT 交易树、MPT 收据树和 MPT 状态树, 最终都会在采用多级数据存储结构的 Key-Value 型数据库 (比如, LevelDB) 中进行存储。

25 [65] 而采用多级存储结构的上述数据库, 通常可以被划分为 n 级数据存储; 例如, 各级数据存储可以依次设为 L0, L1, L2, L3...L(n-1); 对于上述数据库中的各级数据存储而言, 等级编号越小通常级别越高; 例如, L0 存储的是最新的若干区块的数据, L1 存储的是次新的若干区块数据, 依次类推。

[66] 其中，各级数据存储对应的存储介质的读写性能，通常也可以存在性能差异；级别高（即等级编号较小的）的数据存储对应的存储介质的读写性能，可以高于级别低的数据存储对应的存储介质的读写性能。

[67] 例如，在实际应用中，级别高的数据存储，可以使用读写性能较高的存储介质；

5 而级别低的数据存储，可以使用单位成本低，且容量较大的存储介质。在实际应用中，随着区块高度的增长，在数据库中存储的数据，会包含很多历史数据；而且，区块号越小的区块中的数据越久远，越不重要。因此，为了降低整体的存储成本，通常需要对不同区块高度的数据进行“区别对待”；

[68] 例如，可以将区块号较小的区块中的数据，存储至成本较低的存储介质上；而将

10 区块号较大的区块中的数据，存储在成本较高的存储介质上。

[69] 在针对数据库中存储的 MPT 交易树、MPT 收据树和 MPT 状态树等数据进行分级存储时，由于 MPT 交易树和 MPT 收据树，与各个区块相对应，实际上是“区块间无关”的数据；因此，对于 MPT 交易树和 MPT 收据树，很容易进行分级存储；例如，直接按照 MPT 交易树和 MPT 收据树上的 node 所属的区块号进行数据迁移即可完成分级存储。

15 [70] 基于此，本说明书将不再具体阐述 MPT 交易树和 MPT 收据树的分级存储，而重点阐述 MPT 状态树的分级存储。

[71] 请参见图 1，图 1 为本说明书示出的一种将区块链的账户状态数据组织成 MPT 状态树的示意图。

[72] MPT 树，是一种经过改良的，融合了 Merkle 树和 Trie 字典树（也称之为前缀树）

20 两种树形结构的优点的 Merkle 树变种。

[73] 在 MPT 树中通常包括三种数据节点，分别为叶子节点（leaf node），扩展节点（extension node）和分支节点（branch node）。

[74] 叶子节点，表示为[key, value]的一个键值对，其中 key 是种特殊十六进制编码。

25 [75] 扩展节点，也是[key, value]的一个键值对，但是这里的 value 是其他节点的 hash 值（hash 指针）。也就是说通过 hash 指针链接到其他节点。

分支节点，因为 MPT 树中的 key 被编码成一种特殊的 16 进制的表示，再加上最后的 value，所以分支节点是一个长度为 17 的 list，前 16 个元素对应着 key 中的 16 个可能的十六进制字符（一个字符对应一个半字节 nibble）。如果有一个[key, value]对在这个分支节点终止，最后一个元素代表一个 value 值，即分支节点既可以是搜索路径的终

止也可以是路径的中间节点。

[76] 假设需要组织成 MTP 状态树的账户状态数据如下表 1 所示：

账户地址 (Key)							账户状态(Value)
a	7	1	1	3	5	5	state1
a	7	7	d	3	3	7	state2
a	7	f	9	3	6	5	state3
a	7	7	d	3	9	7	state4

表 1

[77] 在表 1 中，账户地址是由若干 16 进制的字符构成的字符串。账户状态 state，是由上述 Balance, Nonce, Code 和 storage 等字段构成的结构体。

[78] 最终按照表 1 中的账户状态数据组织成的 MPT 状态树，参见图 1 所示；如图 1 所示，按照表 1 中的账户状态数据组织成的 MPT 状态树，是由 4 个叶子节点，2 个分支节点，和 2 个扩展节点构成。

[79] 在图 1 中，prefix 字段为扩展节点和叶子节点共同具有的前缀字段。该 prefix 字段的取值，在实际应用中可以用于表示节点类型。

[80] prefix 字段的取值为 0，表示包含偶数个 nibbles 的扩展节点；如前所述，nibble 表示半字节，由 4 位二进制组成，一个 nibble 可以对应一个组成账户地址的字符。

[81] prefix 字段的取值为 1，表示包含奇数个 nibble(s)的扩展节点；

[82] prefix 字段的取值为 2，表示包含偶数个 nibbles 的叶子节点；

[83] prefix 字段的取值为 3，表示包含奇数个 nibble(s)的叶子节点。

[84] 而分支节点，由于其是并列单 nibble 的前缀节点，因此分支节点不具有上述 prefix 字段。

[85] 扩展节点中的 Shared nibble 字段，对应该扩展节点所包含的键值对的 key 值，表示账户地址之间的共同字符前缀；比如，上表中的所有账户地址均具有共同的字符前缀 a7。Next Node 字段中填充下一个节点的 hash 值（hash 指针）。

[86] 分支节点中的 16 进制字符 0~f 字段，对应该分支节点所包含的键值对的 key 值；如果该分支节点为账户地址在 MPT 树上的搜索路径上的中间节点，则该分支节点的 Value 字段可以为空值。0~f 字段中用于填充下一个节点的 hash 值。

[87] 叶子节点中的 Key-end，对应该叶子节点所包含的键值对的 key 值，表示账户地址

的最后几个字符。从根节点搜索到叶子节点的搜索路径上的各个节点的 key 值，构成了一个完整的账户地址。该叶子节点的 Value 字段填充账户地址对应的账户状态数据；例如，可以对上述 Balance, Nonce, Code 和 storage 等字段构成的结构体进行编号后，填充至叶子节点的 Value 字段。

5 [88] 进一步的，如图 1 所示的 MPT 状态树上的 node，最终也是以 Key-Value 键值对的形式存储在数据库中；

[89] 其中，当 MPT 状态树上的 node 在数据库中进行存储时，MPT 状态树上的 node 的键值对中的 key，为 node 所包含的数据内容的 hash 值；MPT 状态树上的 node 的键值对中的 Value，为 node 所包含的数据内容。

10 [90] 也即，在将 MPT 状态树上的 node 存储至数据库时，可以计算该 node 所包含的数据内容的 hash 值（即对 node 整体进行 hash 计算），并将计算出的 hash 值作为 key，将该 node 所包含的数据内容作为 value，生成 Key-Value 键值对；然后，将生成的 Key-Value 键值对存储至数据库中。

[91] 由于 MPT 状态树上的 node，是以 node 所包含的数据内容的 hash 值为 Key，node 15 所包含的数据内容为 value 进行存储；因此，在需要查询 MPT 状态树上的 node 时，通常可以基于 node 所包含的数据内容的 hash 值作为 key 来进行内容寻址。而采用“内容寻址”，对于一些“内容重复”的 node，则通常可以进行“复用”，以节约数据存储的存储空间。

[92] 如图 2 所示，图 2 为本说明书示出的一种 MPT 状态树上的 node 复用的示意图。

20 [93] 在实际应用中，区块链每产生一个最新区块，则在该最新区块中的交易被执行之后，区块链中与这些被执行的交易相关账户的账户状态，通常也会随之发生变化；

[94] 例如，当区块中的一笔“转账交易”执行完毕后，与该“转账交易”相关的转出方账户和转入方账户的余额（即这些账户的 Balance 字段的取值），通常也会随之发生变化。

25 [95] 而节点设备在区块链产生的最新区块中的交易执行完毕后，由于当前区块链中的账户状态发生了变化，因此节点设备需要根据区块链中所有账户当前的账户状态数据，来构建 MPT 树，用于维护区块链中所有账户的最新状态。

[96] 也即，每当区块链中产生一个最新区块，并且该最新区块中的交易执行完毕后，导致区块链中的账户状态发生变化，节点设备都需要基于区块链中所有账户最新的账户

状态数据，重新构建一颗 MPT 树。

[97] 换句话说，区块链中每一个区块，都有一个与之对应的 MPT 状态树；该 MPT 状态树，维护了在该区块中的交易在执行完毕后，区块链中所有账户最新的账户状态。

[98] 而需要说明的是，一个最新区块中的交易执行完毕后，可能仅仅会导致部分账户的账户状态发生变化；因此，在更新 MPT 状态树时，并不需要基于区块链中所有的账户当前的状态数据，重新构建一颗完整的 MPT 状态树，而只需要在该最新区块之前的区块对应的 MPT 状态树的基础上，对部分账户状态发生变化的账户对应的 node 进行更新即可。而对于 MPT 状态树上与账户状态未发生变化的账户对应的 node 而言，由于这些 node 为发生数据更新，可以直接复用该最新区块之前的区块对应的 MPT 状态树上相应的 node 即可。

[99] 如图 2 所示，假设表 1 中的账户状态数据，为 Block N 中的交易执行完毕后，区块链上所有账户的最新账户状态；基于表 1 中的账户状态数据组织成的 MPT 状态树，仍如图 1 所示。

[100] 假设当 Block N+1 中的交易执行完毕后，导致上述表 1 中的账户地址为“a7f9365”的账户状态，由“state3”更新为“state5”；此时，在 Block N+1 更新 MPT 状态树时，并不需要基于 Block N+1 中的交易执行完毕后，区块链中所有的账户当前的状态数据，重新构建一颗 MPT 状态树。

[101] 请参见图 2，在这种情况下，可以仅将 Block N 对应的 MPT 树上（即图 1 示出的 MPT 状态树），“key-end”为“9365”的叶子节点中的 Value，由“state3”更新为“state5”，并继续更新从 root 节点到该叶子节点的路径上的所有节点的 hash 指针；也即，当 MPT 状态树上的叶子节点发生更新，由于该叶子节点整体的 hash 值发生更新，那么从根节点到该叶子节点的路径上的所有的节点的 hash 指针也会随之发生更新。例如，请继续参见图 2，除了需要更新“key-end”为“9365”的叶子节点中的 Value 值以外，还需要更新该叶子节点的上一个分支节点（Branch Node）的 f 字段中填充的，指向该叶子节点的哈希指针；进一步的，还可以继续向根节点追溯，继续更新该分支节点的上一个根节点（Root Extension Node）的“Next Node”字段中填充的，指向该分支节点的 hash 指针。

[102] 而除了以上发生更新的节点以外，其它未发生更新的节点，都可以直接复用 Block N 的 MPT 状态树上对应的节点即可；

[103] 其中，由于 Block N 对应的 MPT 树，最终需要作为历史数据进行保留；因此，在 Block N+1 更新 MPT 状态树时，对于这些发生更新的 node，并不是对 Block N 对应的 MPT 状态树上原来的 node 的基础上，直接进行修改更新，而是在 Block N+1 对应的 MPT 树上重新创建这些发生更新的 node。

5 [104] 也即，对于与 Block N+1 对应的 MPT 状态树上，实际上只需要重新创建少量发生更新的 node，对于其它未发生更新的 node，可以通过直接复用 Block N 对应的 MPT 状态树上对应的节点。

[105] 例如，如图 2 所示，对于 Block N+1 对应的 MPT 状态树上，实际上只需要重新创建少量发生更新的 node；比如，图 2 中仅需要重新创建一个作为根节点的扩展节点、一个分支节点和一个叶子节点；对于未发生更新的 node，可以通过在该 MPT 状态树上这些重新创建的 node 中，添加指向 Block N 对应的 MPT 状态树上的相应 node 的 hash 指针来完成 node 的复用。而 Block N 对应的 MPT 状态树上那些更新前的 node，将作为历史账户状态数据进行保存；比如，图 2 示出的“key-end”为“9365”，且 Value 为“state3”的叶子节点，将作为历史数据进行保留。在以上例子中，以 Block N+1 的 MPT 状态树上的少量 node 发生内容更新，可以“复用”上一个区块 Block N 的大多数 node 为例进行了说明。而在实际应用中，Block N+1 的 MPT 状态树上也可能会较上一个区块 Block N 新增 node。

10

15

[106] 在这种情况下，该新增的 node 虽然无法直接从上一个区块 Block N 的 MPT 树中进行复用，但有可能从更早之前的区块的 MPT 状态树上进行“复用”；

20 [107] 例如，Block N+1 的 MPT 状态树上新增的 node，虽然在 Block N 的 MPT 状态树上出现过，但出现在更早的 Block 的 MPT 状态树上；比如，出现在 Block N-1 的 MPT 状态树上；因此，Block N+1 的 MPT 状态树上新增的 node，可以直接复用 Block N-1 的 MPT 状态树上对应的 node 即可。

[108] 可见，MPT 状态树的节点复用，一共有两种“复用”的情形：

25 [109] 一种情形是，一个区块的 MPT 状态树上只有少量的节点发生内容更新，则可以“复用”上一个区块的大多数节点；

[110] 另一种情形是，一个区块的 MPT 状态树较上一个区块的 MPT 状态树新增了节点，则可以“复用”更早之前的区块的 MPT 状态树上对应的节点。

[111] 然而，通过节点复用，虽然可以节约数据库的存储空间，但由于各个区块的 MPT

状态树上的节点之间，可能存在复杂的复用关系，每个区块的 MPT 状态树上的 node，都可能被下一个区块，或者是下一个区块之后的若干连续的区块进行复用；因此，这种复杂的节点复用关系，势必会对 MPT 状态树的分级存储造成困难。

5 [112] 例如，在需要将一些节点作为历史数据，从本级数据存储向下一级数据存储迁移时，由于这些节点可能会被下一个区块；甚至是该下一个区块之后的若干个区块复用；而这些节点将会被哪些节点复用，又是无法准确预知的；因此，这就会导致无法对数据库中存储的 MPT 状态树上的节点进行精确剪枝；其中，所谓剪枝，是指清除各区块的 MPT 状态树上的节点之间的复用关系，将记录了历史状态数据的节点从 MPT 状态树上移除，并保留记录了最新状态数据的 node。在这种情况下，显然无法满足分级存储的需求。

10

[113] 基于此，本说明书提出一种针对由区块链的账户状态数据组成的 Merkle 状态树进行分级存储的方法。

[114] 在实现时，仍然可以将区块链的账户状态数据组织成 Merkle 状态树，在采用多级数据存储结构的数据库中进行存储；例如，仍然可以采用 MPT 树的数据结构，将区块链的账户状态数据组织成 MPT 状态树；

15

[115] 而对于在数据库中存储的 Merkle 状态树，可以针对该 Merkle 状态树上的 node，分别标记一个区块号，该区块号具体用于指示该 node 发生数据更新时所在区块的区块号；

[116] 例如，在一种实施方式中，区块链产生的最新区块中的交易执行完毕后，可以基于该最新区块中的交易的执行结果，来确定该最新区块的 Merkle 状态树上发生数据更新的 node；其中，发生数据更新的 node，通常包含 value 值发生更新的 node 和新增的 node。当确定该最新区块的 Merkle 状态树上发生数据更新的 node 之后，可以为这些发生数据更新的 node 标记该最新区块的区块号，表示该 node 在该区块号对应的区块发生了数据更新。

20

25 [117] 进一步的，当数据库中的任一级目标数据存储满足了数据迁移条件时；比如，当该目标数据存储的存储容量达到阈值；首先，可以确定该目标数据存储中需要迁移至下一级数据存储的 node 对应的区块号区间，并基于该区块号区间确定迁移临界值；其中，该迁移临界值为大于该区块号区间的区块号临界值；

[118] 例如，在一种实施方式中，如果该区块号区间的右区间为开区间，则将该区块号

区间的右端点值，确定为上述迁移临界值；比如，假设该区块号区间为 $[a, b)$ 时，则将 b 确定为上述迁移临界值；如果该区块号区间的右区间为闭区间，将该区块号区间的右端点值与区块链的区块号递增步长的和，确定为上述迁移临界值；比如，假设该区块号区间为 $[a, b]$ ，区块链的区块号递增步长为 1 （即区块链中的区块号以 1 为递增步长致密递增）时，则将 $b+1$ 确定为上述迁移临界值。

[119] 当确定了上述迁移临界值后，可以遍历与该迁移临界值对应的目标区块的 Merkle 状态树，查找被标记的区块号小于该迁移临界值的目标 node，并将查找到的目标 node 的区块号修改为该迁移临界值；

[120] 进一步的，在将查找到的目标 node 的区块号强制修改为该迁移临界值之后，可以继续遍历与上述区块号区间中的各个区块号对应的区块的 Merkle 状态树，查找被标记的区块号小于该迁移临界值的 node，然后将查找到的 node 从上述目标数据存储中，迁移至该目标数据存储的下一级数据存储；例如，先将查找到的 node 写入到该目标数据存储的下一级数据存储，在写入成功后，再将该 node 从目标数据存储中清除。

[121] 通过以上技术方案，可以实现对数据库中存储的 Merkle 状态树上的 node 进行精确剪枝，将记录了历史状态数据的 node 从 Merkle 状态树上移除，迁移至下一级数据存储，并在本级数据存储中继续存储和保留记录了最新状态数据的 node，进而完成针对数据库中存储的 Merkle 状态树的分级存储。

[122] 请参见图 3，图 3 是一示例性实施例提供的一种基于区块链的分级存储方法的流程图。所述方法应用于区块链节点设备；其中，区块链的账户状态数据被组织成 Merkle 状态树在数据库中存储；该数据库包括多级数据存储；上述 Merkle 状态树上的数据节点被标记了区块号；该区块号指示该数据节点发生数据更新时所在的区块；所述方法包括以下步骤：

[123] 步骤 302，当所述数据库中的任一级目标数据存储满足了数据迁移条件时，确定所述目标数据存储中待迁移至下一级数据存储的数据节点对应的区块号区间；

[124] 步骤 304，基于所述区块号区间确认迁移临界值；其中，所述迁移临界值为大于所述区块号区间的区块号临界值；

[125] 步骤 306，遍历与所述迁移临界值对应的目标区块的 Merkle 状态树，查找被标记的区块号小于所述迁移临界值的目标数据节点，并将查找到的所述目标数据节点的区块号修改为所述迁移临界值；

[126] 步骤 308, 遍历与所述区块号区间中的各个区块号对应的区块的 Merkle 状态树, 查找被标记的区块号小于所述迁移临界值的数据节点, 并将查找到的数据节点从所述目标数据存储中, 迁移至所述目标数据存储的下一级数据存储。

5 [127] 上述数据库, 具体可以是采用多级数据存储结构的 Key-Value 型数据库 (比中进行存储; 例如, 在示出的一种实施方式中, 上述数据库可以为 LevelDB 数据库; 或者, 基于 LevelDB 架构的数据库; 比如, Rocksdb 数据库就是一种典型的基于 LevelDB 数据库架构的数据库。

10 [128] 区块链中的账户状态数据, 可以被组织成 Merkle 状态树的数据结构, 在上述数据库中进行存储; 例如, 上述 Merkle 状态树具体可以是 MPT 树, 可以采用 MPT 树的数据结构, 将区块链的账户状态数据组织成 MPT 状态树。

[129] 以下以采用 MPT 树的数据结构, 将区块链中的账户状态数据组织成 MPT 状态树为例, 对本说明书的技术方案进行详细描述;

[130] 其中, 需要强调的是, 以采用 MPT 树的数据结构来组织区块链中的账户状态数据, 仅为示例性的。

15 [131] 在实际应用中, 对于基于以太坊架构而衍生出的区块链项目, 除了可以采用诸如 MPT 树等改良版的 Merkle 树以外, 也可以采用其他形式的类似于 MPT 树的融合了 Trie 字典树的树形结构的 Merkle 树变种, 在本说明书中不再进行一一列举。

20 [132] 在本说明书中, 接入区块链的用户客户端, 可以将数据打包成区块链所支持的标准交易格式, 然后发布至区块链; 而区块链中的节点设备, 可以基于搭载的共识算法与其它节点设备一起, 对用户客户端发布至区块链的这些交易进行共识, 以此来为区块链产生最新区块;

[133] 其中, 区块链中支持的共识算法, 通常分为节点设备需要争夺每一轮的记账周期的记账权的共识算法, 和预先为每一轮记账周期选举记账节点 (不需要争夺记账权) 的共识算法。

25 [134] 例如, 前者以工作量证明 (Proof of Work, POW)、股权证明 (Proof of Stake, POS)、委任权益证明 (Delegated Proof of Stake, DPOS) 等共识算法为代表; 后者以实用拜占庭容错 (Practical Byzantine Fault Tolerance, PBFT) 等共识算法为代表。

[135] 对于采用工作量证明 (Proof of Work, POW) 以及股权证明 (Proof of Stake, POS)、委任权益证明 (Delegated Proof of Stake, DPOS) 等共识算法的区块链网络中, 争夺记

账权的节点设备，都可以在接收到交易后执行该笔交易。争夺记账权的节点设备中可能其中一个在本轮争夺记账权的过程中胜出，成为记账节点。记账节点可以将收到的交易与其它交易一起打包并生成最新区块，并将生成的最新区块发送至其它节点设备进行共识。

5 [136] 对于采用实用拜占庭容错（Practical Byzantine Fault Tolerance, PBFT）等共识算法的区块链网络中，具有记账权的节点设备在本轮记账前已经商定好。因此，节点设备在接收到交易后，如果自身不是本轮的记账节点，则可以将该交易发送至记账节点。

[137] 对于本轮的记账节点，在将该交易与其它交易一起打包并生成最新区块的过程中或者之前，可以执行该交易。记账节点在将该交易与其它交易一起打包生成新区块后，
10 可以将生成的最新区块或者该最新区块的区块头发送至其它节点设备进行共识。

[138] 如上所述，无论区块链采用以上示出的哪种共识算法，本轮的记账节点都可以将接收到的交易打包并生成最新区块，并将生成的最新区块或者该最新区块的区块头发送至其它节点设备进行共识验证。如果其它节点设备接收到最新区块或者该最新区块的区块头后，经验证没有问题，可以将该最新区块追加到原有的区块链末尾，从而完成区块链的记账过程。
15

[139] 在本说明书中，区块链中的节点设备在执行了经过共识产生的最新区块中打包的交易之后，区块链中与这些被执行的交易相关的账户状态，通常也会随之发生变化；因此，节点设备在最新区块中打包的交易执行完毕后，可以根据区块链中所有账户最新的账户状态数据，组织成 MPT 状态树的数据结构。

20 [140] 其中，根据区块链中所有账户最新的账户状态数据，组织成 MPT 状态树时，仍然可以采用如图 2 示出的复用该最新区块之前的区块对应的 MPT 树上的节点的方式，在本说明书中不再赘述。

[141] 当节点设备根据区块链中所有账户最新的账户状态数据，组织成 MPT 状态树后，可以将该 MPT 状态树上的数据节点，以 Key-Value 键值对的形式在采用多级数据存储结构的 Key-Value 型数据库中进行存储。
25

[142] 例如，在实际应用中，对于最新的若干区块对应的 MPT 状态树上的数据节点，可以默认存放在上述数据库中的级别最高的 L0 级数据存储中。对于次新的若干区块对应的 MPT 状态树上的数据节点，可以存放在上述数据库中的级别次高的 L1 级数据存储中；以此类推。其中，每一级数据存储所存储的 MPT 状态树对应的区块数，在本说明书中

不进行特别限定；比如，可以规定级别最高的 L0 级数据存储中存储最新的 N 个区块的 MPT 状态树，级别次高的 L1 级数据存储次新的 N 个区块的 MPT 状态树；以此类推。

[143] 在本说明书，对于构建的 MPT 状态树上的数据节点，可以被标记区块号；其中，为 MPT 状态树上的数据节点所标记的区块号，具体用于指示该数据节点发生数据更新时所在的区块；

[144] 例如，如图 2 所示，以 Block N+1 对应的 MPT 状态树上发生数据更新的 node 为例，为这些 node 标记的区块号，即为 N+1，以表示这些 node 在 Block N+1 中的交易执行完毕后，发生了数据更新。

[145] 在实现时，节点设备可以在本地启动一个“区块状态更新线程”，来维护和更新 MPT 状态树上的 node 的状态。当区块链产生的最新区块中的交易执行完毕后，该“区块状态更新线程”可以确定该最新区块对应的 MPT 状态树上发生数据更新的 node；其中，发生数据更新的 node，通常包含 value 值发生更新的 node 和新增的 node。

[146] 例如，在实现时，可以直接根据该最新区块中的交易的执行结果，来确定该最新区块对应的 MPT 状态树上发生数据更新的 node；或者，也可以通过判断 MPT 状态树上的 node 是否为复用的 node，来确定该最新区块对应的 MPT 状态树上发生数据更新的 node；比如，如果是复用的 node，则表明该 node 是未发生数据更新的 node；

[147] 当确定该最新区块对应的 MPT 状态树上发生数据更新的 node 之后，该“区块状态更新线程”，可以为这些发生数据更新的 node 标记该最新区块的区块号，表示该 node 在该区块号对应的区块发生了数据更新。

[148] 通过这种方式，每当区块链产生一个最新区块，“区块状态更新线程”都可以及时的对该最新区块对应的 MPT 状态树上发生数据更新的 node，标记该最新区块的区块号，使得通过遍历上述数据库中存储的 MPT 状态树上的各个 node，就能够通过为各个 node 标记的区块号，获知到该 node 发生数据更新时所在的区块；由于 MPT 状态树上的各个 node（尤其是 MPT 树上的叶子节点）最近一次发生数据更新后的 value 值，通常可以指示该 node 的最新状态；因此，通过查看为各个 node 标记的区块号，则可以获知各个 node 的最新状态，是在哪个区块产生的；从而这种为 node 标记区块号的机制，可以对上述数据库的各级数据存储的数据迁移提供依据。

[149] 其中，需要说明的是，“区块状态更新线程”为 MPT 状态树上的 node 标记区块号的动作，可以发生在构建 MPT 状态树的过程中，也可以发生在将构建完成的 MPT 状

态树存储至上述数据库的过程中，在本说明书中不进行特别限定。

[150]也即，“区块状态更新线程”可以在节点设备构建MPT状态树的过程中，为MPT状态树上发生数据更新的node标记区块号，也可以在将构建完成的MPT状态树存储至上述数据库的过程中，为MPT状态树上发生数据更新的node标记区块号。

5 [151]其中，需要说明的是，在本说明书中，为MPT状态树上的node标记的区块号，具体可以标记在与node的Value对应的区块号字段中。在实际应用中，上述区块号字段具体可以与node的Value存储在一起，也可以不与node的Value存储在一起，而是存储在另外物理位置。

10 [152]在示出的一种实施方式中，可以在上述数据库中存储的MPT状态树上的node的Value之中，设置一区块号字段；为该MPT状态树上的node标记的区块号，可以添加在该node的value中设置的该区块号字段之中。其中，该区块号字段在node的Value之中的位置，在本说明书中不进行特别限定；

15 [153]例如，在一个例子中，可以在MPT状态树上每个发生数据更新的node的Value之后（也可以放置在Value之前），增加一个区块号字段，然后在该区块号字段中为该node标记区块号。

[154]在示出的另一种实施方式中，上述区块号字段具体也可以不与MPT状态树上的node的Value存储在一起，而是设置在单独的物理位置上；在这种情况下，可以在上述数据库中，设置一个指向MPT状态树上的node的Value的存储位置的区块号字段；为该MPT状态树上的node标记的区块号，可以添加在设置在单独的物理位置上的该区块号字段之中。

[155]例如，以上述数据库为基于LevelDB架构的Rocksdb数据库为例，可以将上述区块号字段设置在单独的column family（列族）之中。

25 [156]通过将区块号字段不与node的Value存储在一起，而是存储在单独的物理位置上，使得在需要遍历为MPT状态树上的node标记的区块号时，可以只从单独的物理位置上读取区块号即可，而并不需要读取该node整个的value值，可以提升遍历为MPT状态树上的node标记的区块号时的查找效率。

[157]在本说明书中，节点设备还可以在本地启动一个“迁移线程”，用于将上述数据库中各级数据存储中存储的MPT状态树上的node数据，向下级数据存储进行迁移。

[158]上述“迁移线程”，具体可以执行一个定时任务，可以周期性的确定上述数据库

中的各级数据存储是否满足预设的数据迁移条件；

[159] 其中，上述数据库中的各级数据存储的数据迁移条件，可以基于实际的数据迁移需求来设置，在本说明书中不进行特别限定；

5 [160] 例如，在实际应用中，上述数据库的各级数据存储的数据迁移条件，具体可以是各级数据存储的存储容量达到阈值；或者，也可以是各级数据存储所存储的数据所对应的区块数达到阈值。

[161] 上述“迁移线程”，在确定上述数据库中任一级目标数据存储，满足了数据迁移条件，则该“迁移线程”可以针对该目标数据存储进行数据迁移处理，将该目标数据存储中存储的部分区块的 MPT 状态树，作为历史数据向下一级数据存储进行迁移。

10 [162] 在实现时，当目标数据存储满足了数据迁移条件时，上述“迁移线程”可以确定该目标数据存储中，需要迁移至下一级数据存储的 node 对应的区块号区间；

[163] 其中，该目标数据存储中需要迁移至下一级数据存储的 node 对应的区块号区间，可以基于上一次数据迁移发生后的下一区块的区块号，和该目标数据存储一次能够迁移走的最大区块数来确定；比如，假设上一次数据迁移后的下一个区块是 Block N，该目标数据存储一次能够迁移走的区块数为 30，那么上述区块号区间具体可以是[N, N+29]；
15 或者，也可以是[N, N+30)。

[164] 当上述“迁移线程”确定出该目标数据存储中，需要迁移至下一级数据存储的 node 对应的区块号区间之后，可以基于该区块号区间，来确定本次数据迁移的迁移临界值；其中，该迁移临界值具体可以是一个大于该区块号区间的区块号临界值；

20 [165] 需要说明的是，在实际应用中，上述区块号区间的右区间具体可以是开区间，也可以是闭区间；而在基于上述区块号区间来确定上述迁移临界值时，上述区块号区间的右区间为开区间或者闭区间时，确定出的迁移临界值也可以存在一定的差异。

[166] 在示出的一种实施方式中，如果上述区块号区间的右区间为开区间，在基于上述区块号区间来确定上述迁移临界值时，可以将上述区块号区间的右端点值，确定为上述
25 迁移临界值；

[167] 例如，假设上述区块号区间为[a, b)，那么基于该区块号区间确定出的迁移临界值为该区块号区间的右端点值 b。

[168] 在示出的另一种实施方式中，如果上述区块号区间的右区间为闭区间，在基于上

述区块号区间来确定上述迁移临界值时，可以将上述区块号区间的右端点值与区块链的区块号递增步长的和，确定为上述迁移临界值；

[169] 例如，假设上述区块号区间为[a, b]，区块链的区块号递增步长为 1，那么基于该区块号区间确定出的迁移临界值为该区块号区间的右端点值 b+1。

5 [170] 其中，上述区块链的区块号递增步长，通常可以是 1；也即，区块链的区块号按照 1 为步长进行致密递增；比如，区块号按照 1、2、3、4... 的顺序进行递增。

[171] 当然，在实际应用中，区块链的区块号递增步长具体也可以是大于 1 的整数；比如，区块号也可以按照 1、3、5、7... 的顺序进行递增，在本说明书中不进行特别限定。

10 [172] 进一步的，当上述“迁移线程”，基于上述区块号区间，确定出本次数据迁移的迁移临界值之后，可以根据该迁移临界值，将该目标数据存储中存储的与上述区块号区间对应的区块的 MPT 状态树，作为历史数据向下一级数据存储进行迁移。

[173] 具体地，上述“迁移线程”，首先可以遍历与上述迁移临界值对应的目标区块的 MPT 状态树，查找该 MPT 状态树中，被标记的区块号小于上述迁移临界值的目标数据节点，然后将为查找到的上述目标数据节点标记的区块号修改为上述迁移临界值；

15 [174] 例如，假设上述区块号区间为[a, b)，上述迁移临界值为该区块号区间的右端点值 b，那么上述“迁移线程”可以遍历第 b 号区块的 MPT 状态树，查找该 MPT 状态树上，被标记的区块号小于上述迁移临界值的数据节点，然后将为查找到的这些数据节点标记的区块号修改 b。

20 [175] 进一步的，当上述“迁移线程”，通过遍历与上述迁移临界值对应的目标区块的 MPT 状态树，将该 MPT 状态树中，被标记的区块号小于上述迁移临界值的目标数据节点，全部修改为上述迁移临界值后，此时上述“迁移线程”可以依次遍历与上述区块号区间中的各个区块号对应的区块的 MPT 状态树，查找这些区块的 MPT 状态树上，被标记的区块号小于上述迁移临界值的数据节点；此时，在这些区块的 MPT 状态树上查找到的，被标记的区块号小于上述迁移临界值的数据节点，即为可以作为迁移至该目标数据存储的下一级数据存储的历史数据，上述“迁移线程”可以将查找到的这些数据节点，迁移至该目标数据存储的下一级数据存储；

[176] 例如，在实现时，上述“迁移线程”可以复制查找到的这些数据节点，并将复制的这些数据存储至下一级数据存储，然后在将复制的这些数据节点成功存储至下一级数据存储之后，再将这些数据节点从上述目标数据存储中清除。

[177] 通过这种方式，由于与上述区块号区间中的各个区块号对应的区块的 MPT 状态树上，被标记的区块号小于上述迁移临界值的 node 的区块号，已经被修改为上述迁移临界值；因此，按照上述方式完成数据迁移后，对于这些被标记的区块号修改为上述迁移临界值的 node 而言，将会继续在本级数据存储中存储和保留；这相当于对数据库中存储的 MPT 状态树上的 node 进行了一次剪枝，将记录了历史状态数据的 node 从 MPT 状态树上移除，迁移至下一级数据存储，并在本级数据存储中继续存储和保留记录了最新状态数据的 node，进而完成针对数据库中存储的 MPT 状态树的分级存储。

[178] 其中，需要说明的是，上述“区块状态更新线程”和上述“迁移线程”对应的处理动作，可以并发执行；也即，上述“区块状态更新线程”和上述“迁移线程”可以对 MPT 状态树上的同一个 node 进行并发竞争处理。因此，在实际应用中，为了避免上述“区块状态更新线程”和上述“迁移线程”对同一个 node 进行并发竞争处理，可能造成的处理错误，可以通过技术手段，来保证在任一时刻，只能有一个线程来访问 MPT 状态树上的同一个 node。

[179] 其中，在保证在任一时刻，只能有一个线程来访问 MPT 状态树上的同一个 node 所采用的具体技术手段，在本说明书中不进行特别限定；例如，在实现时，可以采用互斥锁技术、单线程技术来实现，在本说明书中不再进行详述。

[180] 与上述方法实施例相对应，本申请还提供了装置的实施例。

[181] 与上述方法实施例相对应，本说明书还提供了一种基于区块链的分级存储装置的实施例。

[182] 本说明书的基于区块链的分级存储装置的实施例可以应用在电子设备上。装置实施例可以通过软件实现，也可以通过硬件或者软硬件结合的方式实现。以软件实现为例，作为一个逻辑意义上的装置，是通过其所在电子设备的处理器将非易失性存储器中对应的计算机程序指令读取到内存中运行形成的。

[183] 从硬件层面而言，如图 4 所示，为本说明书的基于区块链的分级存储装置所在电子设备的一种硬件结构图，除了图 4 所示的处理器、内存、网络接口、以及非易失性存储器之外，实施例中装置所在的电子设备通常根据该电子设备的实际功能，还可以包括其他硬件，对此不再赘述。

[184] 图 5 是本说明书一示例性实施例示出的一种基于区块链的分级存储装置的框图。

[185] 请参考图 5，所述基于区块链的分级存储装置 50 可以应用在前述图 4 所示的电子

设备中，其中所述区块链的账户状态数据被组织成 Merkle 状态树在数据库中存储；所述数据库包括多级数据存储；所述 Merkle 状态树上的数据节点被标记了区块号；所述区块号指示所述数据节点发生数据更新时所在的区块；所述装置 50 包括：

5 [186] 确定模块 501，当所述数据库中的任一级目标数据存储满足了数据迁移条件时，确定所述目标数据存储中待迁移至下一级数据存储的数据节点对应的区块号区间；基于所述区块号区间确认迁移临界值；其中，所述迁移临界值为大于所述区块号区间的区块号临界值；

10 [187] 修改模块 502，遍历与所述迁移临界值对应的目标区块的 Merkle 状态树，查找被标记的区块号小于所述迁移临界值的目标数据节点，并将查找到的所述目标数据节点的区块号修改为所述迁移临界值；

[188] 迁移模块 503，遍历与所述区块号区间中的各个区块号对应的区块的 Merkle 状态树，查找被标记的区块号小于所述迁移临界值的数据节点，并将查找到的数据节点从所述目标数据存储中，迁移至所述目标数据存储的下一级数据存储。

[189] 在本实施例中，所述确定模块 501：

15 [190] 如果所述区块号区间的右区间为开区间，将所述区块号区间的右端点值，确定为所述迁移临界值；

[191] 如果所述区块号区间的右区间为闭区间，将所述区块号区间的右端点值与所述区块链的区块号递增步长的和确定为所述迁移临界值。

[192] 在本实施例中，所述数据库为 Key-Value 数据库；

20 [193] 所述 Merkle 状态树上的数据节点以 Key-Value 键值对的形式存储在所述数据库中；其中，所述 Key-Value 键值对的 key 为所述数据节点包含的数据内容的 hash 值；所述 Key-Value 键值对的 value 为所述数据节点包含的数据内容。

[194] 在本实施例中，所述装置 50 还包括：

25 [195] 标记模块 504（图 5 中未示出），确定最新区块的 Merkle 状态树上发生数据更新的数据节点；为所述最新区块的 Merkle 状态树上发生数据更新的数据节点标记所述最新区块的区块号。

[196] 在本实施例中，所述数据库中存储的所述 Merkle 状态树上的数据节点的 Value 中，包含预设的区块号字段；为所述 Merkle 状态树上的数据节点标记的区块号，添加在所

述区块号字段中。

[197] 在本实施例中,所述数据库中预设了指向所述 Merkle 状态树上的数据节点的 Value 的存储位置的区块号字段;为所述 Merkle 状态树上的数据节点标记的区块号,添加在所述区块号字段中。

5 [198] 在本实施例中,所述 Merkle 状态树为 Merkle Patricia Tree 状态树。

[199] 在本实施例中,所述数据库为 LevelDB 数据库;或者基于 LevelDB 架构的数据库;

[200] 在本实施例中,所述数据库为基于 LevelDB 架构的 Rocksdb 数据库。

[201] 在本实施例中,所述多级数据存储对应的存储介质的读写性能,存在性能差异;其中,级数高的数据存储对应的存储介质的读写性能,高于级数低的数据存储对应的存储介质的读写性能。

10

[202] 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,计算机的具体形式可以是个人计算机、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、电子邮件收发设备、游戏控制台、平板计算机、可穿戴设备或者这些设备中的任意几种设备的组合。

15

[203] 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。

[204] 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。

20

[205] 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。

25

[206] 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。

[207] 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。

[208] 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。

[209] 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”。

[210] 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。

权利要求书

1. 一种基于区块链的分级存储方法, 所述区块链的账户状态数据被组织成 Merkle 状态树在数据库中存储; 所述数据库包括多级数据存储; 所述 Merkle 状态树上的数据节点被标记了区块号; 所述区块号指示所述数据节点发生数据更新时所在的区块; 所述方法包括:

当所述数据库中的任一级目标数据存储满足了数据迁移条件时, 确定所述目标数据存储中待迁移至下一级数据存储的数据节点对应的区块号区间;

基于所述区块号区间确定迁移临界值; 其中, 所述迁移临界值为大于所述区块号区间的区块号临界值;

遍历与所述迁移临界值对应的目标区块的 Merkle 状态树, 查找被标记的区块号小于所述迁移临界值的目标数据节点, 并将查找到的所述目标数据节点的区块号修改为所述迁移临界值;

遍历与所述区块号区间中的各个区块号对应的区块的 Merkle 状态树, 查找被标记的区块号小于所述迁移临界值的数据节点, 并将查找到的数据节点从所述目标数据存储中, 迁移至所述目标数据存储的下一级数据存储。

2. 根据权利要求 1 所述的方法, 所述基于所述区块号区间确定迁移临界值, 包括:

如果所述区块号区间的右区间为开区间, 将所述区块号区间的右端点值, 确定为所述迁移临界值;

如果所述区块号区间的右区间为闭区间, 将所述区块号区间的右端点值与所述区块链的区块号递增步长的和确定为所述迁移临界值。

3. 根据权利要求 1 所述的方法, 所述数据库为 Key-Value 数据库;

所述 Merkle 状态树上的数据节点以 Key-Value 键值对的形式存储在所述数据库中; 其中, 所述 Key-Value 键值对的 key 为所述数据节点包含的数据内容的 hash 值; 所述 Key-Value 键值对的 value 为所述数据节点包含的数据内容。

4. 根据权利要求 1 所述的方法, 所述方法还包括:

确定最新区块的 Merkle 状态树上发生数据更新的数据节点; 为所述最新区块的 Merkle 状态树上发生数据更新的数据节点标记所述最新区块的区块号。

5. 根据权利要求 1 或 4 所述的方法, 所述数据库中存储的所述 Merkle 状态树上的数据节点的 Value 中, 包含预设的区块号字段; 为所述 Merkle 状态树上的数据节点标记的区块号, 添加在所述区块号字段中。

6. 根据权利要求 1 或 4 所述的方法, 所述数据库中预设了指向所述 Merkle 状态树

上的数据节点的 Value 的存储位置的区块号字段; 为所述 Merkle 状态树上的数据节点标记的区块号, 添加在所述区块号字段中。

7. 根据权利要求 1 所述的方法, 所述 Merkle 树为融合了 Trie 字典树的树形结构的 Merkle 树变种。

5 8. 根据权利要求 7 所述的方法, 所述 Merkle 状态树为 Merkle Patricia Tree 状态树。

9. 根据权利要求 1 所述的方法, 所述数据库为 LevelDB 数据库; 或者基于 LevelDB 架构的数据库。

10. 根据权利要求 9 所述的方法, 所述数据库为基于 LevelDB 架构的 Rocksdb 数据库。

10 11. 根据权利要求 1 所述的方法, 所述多级数据存储对应的存储介质的读写性能, 存在性能差异; 其中, 级数高的数据存储对应的存储介质的读写性能, 高于级数低的数据存储对应的存储介质的读写性能。

12. 一种基于区块链的分级存储装置, 所述区块链的账户状态数据被组织成 Merkle 状态树在数据库中存储; 所述数据库包括多级数据存储; 所述 Merkle 状态树上的数据
15 节点被标记了区块号; 所述区块号指示所述数据节点发生数据更新时所在的区块; 所述装置包括:

确定模块, 当所述数据库中的任一级目标数据存储满足了数据迁移条件时, 确定所述目标数据存储中待迁移至下一级数据存储的数据节点对应的区块号区间; 基于所述区块号区间确认迁移临界值; 其中, 所述迁移临界值为大于所述区块号区间的区块号临界
20 值;

修改模块, 遍历与所述迁移临界值对应的目标区块的 Merkle 状态树, 查找被标记的区块号小于所述迁移临界值的目标数据节点, 并将查找到的所述目标数据节点的区块号修改为所述迁移临界值;

迁移模块, 遍历与所述区块号区间中的各个区块号对应的区块的 Merkle 状态树, 查找被标记的区块号小于所述迁移临界值的数据节点, 并将查找到的数据节点从所述目标数据存储中, 迁移至所述目标数据存储的下一级数据存储。
25

13. 根据权利要求 12 所述的装置, 所述确定模块:

如果所述区块号区间的右区间为开区间, 将所述区块号区间的右端点值, 确定为所述迁移临界值;

30 如果所述区块号区间的右区间为闭区间, 将所述区块号区间的右端点值与所述区块链的区块号递增步长的和确定为所述迁移临界值。

14. 根据权利要求 12 所述的装置, 所述数据库为 Key-Value 数据库;

所述 Merkle 状态树上的数据节点以 Key-Value 键值对的形式存储在所述数据库中;
其中, 所述 Key-Value 键值对的 key 为所述数据节点包含的数据内容的 hash 值; 所述 Key-Value 键值对的 value 为所述数据节点包含的数据内容。

5 15. 根据权利要求 12 所述的装置, 所述装置还包括:

标记模块, 确定最新区块的 Merkle 状态树上发生数据更新的数据节点; 为所述最新区块的 Merkle 状态树上发生数据更新的数据节点标记所述最新区块的区块号。

16. 根据权利要求 12 或 15 所述的装置, 所述数据库中存储的所述 Merkle 状态树上的数据节点的 Value 中, 包含预设的区块号字段; 为所述 Merkle 状态树上的数据节点
10 标记的区块号, 添加在所述区块号字段中。

17. 根据权利要求 12 或 15 所述的装置, 所述数据库中预设了指向所述 Merkle 状态树上的数据节点的 Value 的存储位置的区块号字段; 为所述 Merkle 状态树上的数据节点标记的区块号, 添加在所述区块号字段中。

18. 根据权利要求 12 所述的装置, 所述 Merkle 树为融合了 Trie 字典树的树形结构
15 的 Merkle 树变种。

19. 根据权利要求 18 所述的装置, 所述 Merkle 状态树为 Merkle Patricia Tree 状态树。

20. 根据权利要求 12 所述的装置, 所述数据库为 LevelDB 数据库; 或者基于 LevelDB 架构的数据库。

20 21. 根据权利要求 20 所述的装置, 所述数据库为基于 LevelDB 架构的 Rocksdb 数据库。

22. 根据权利要求 12 所述的装置, 所述多级数据存储对应的存储介质的读写性能, 存在性能差异; 其中, 级数高的数据存储对应的存储介质的读写性能, 高于级数低的数据存储对应的存储介质的读写性能。

25 23. 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 1-11 中任一项所述的方法。

30 24. 一种计算机可读存储介质, 其上存储有计算机指令, 其特征在于, 该指令被处理器执行时实现如权利要求 1-11 中任一项所述方法的步骤。

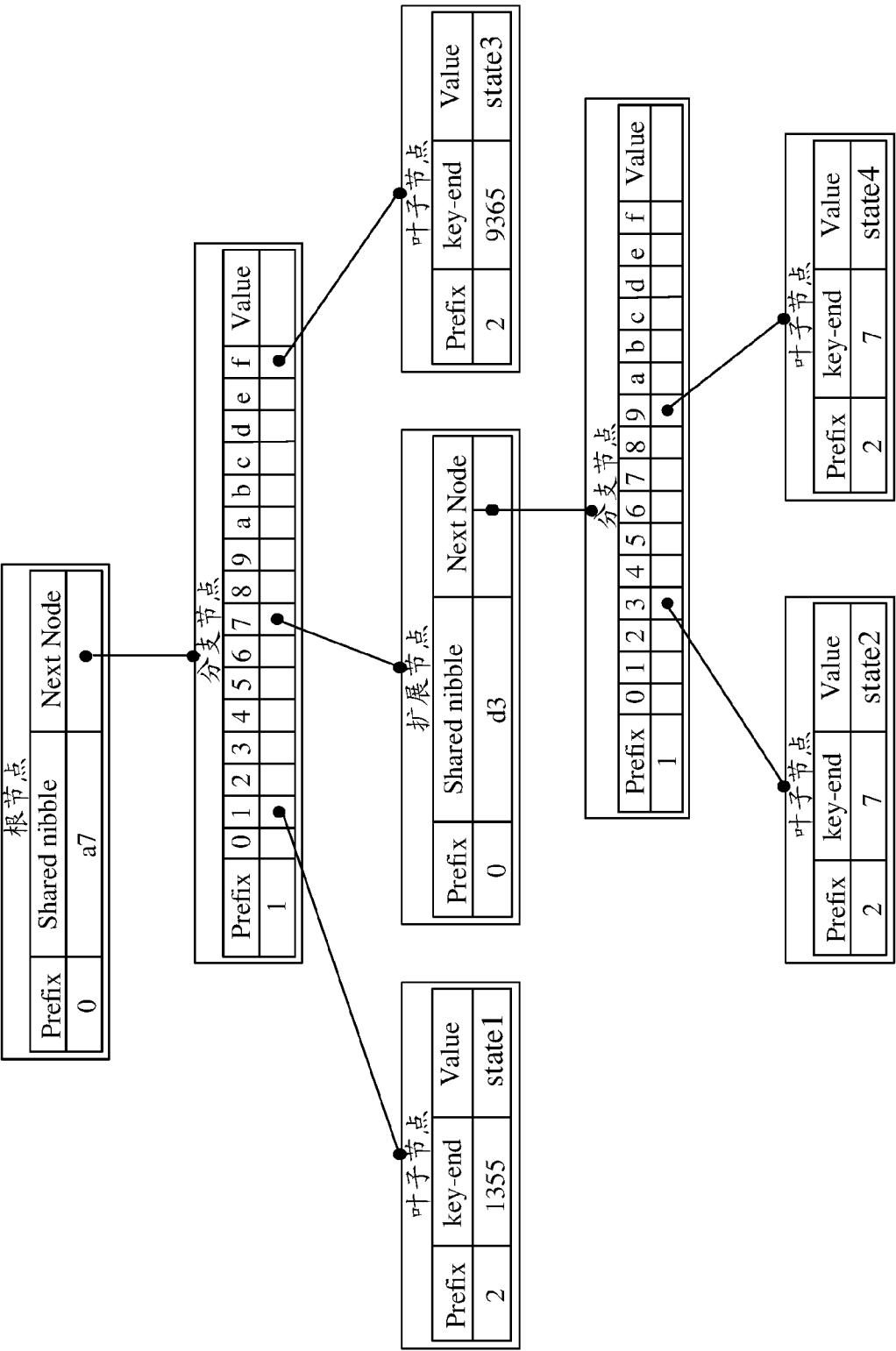


图 1

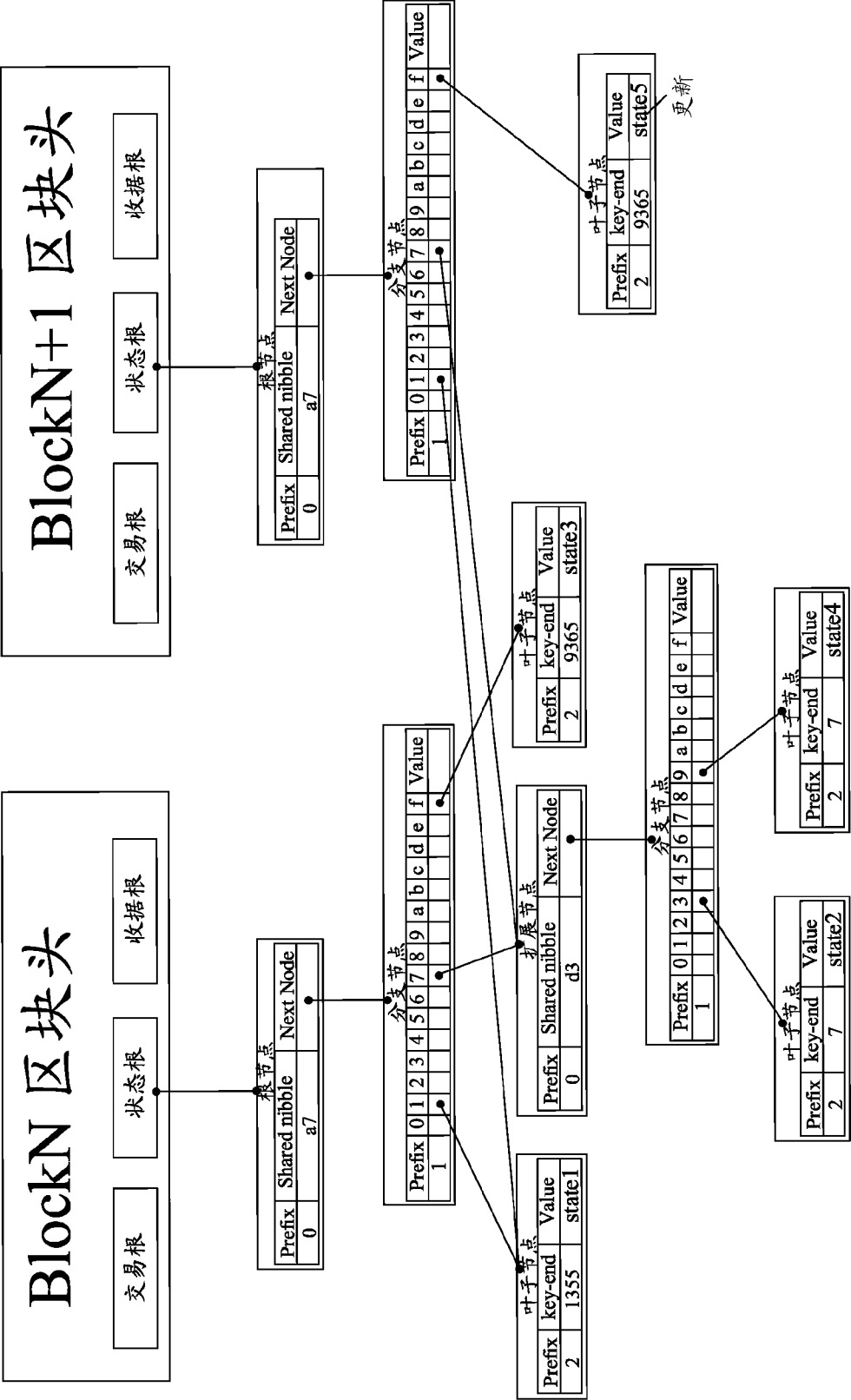


图 2

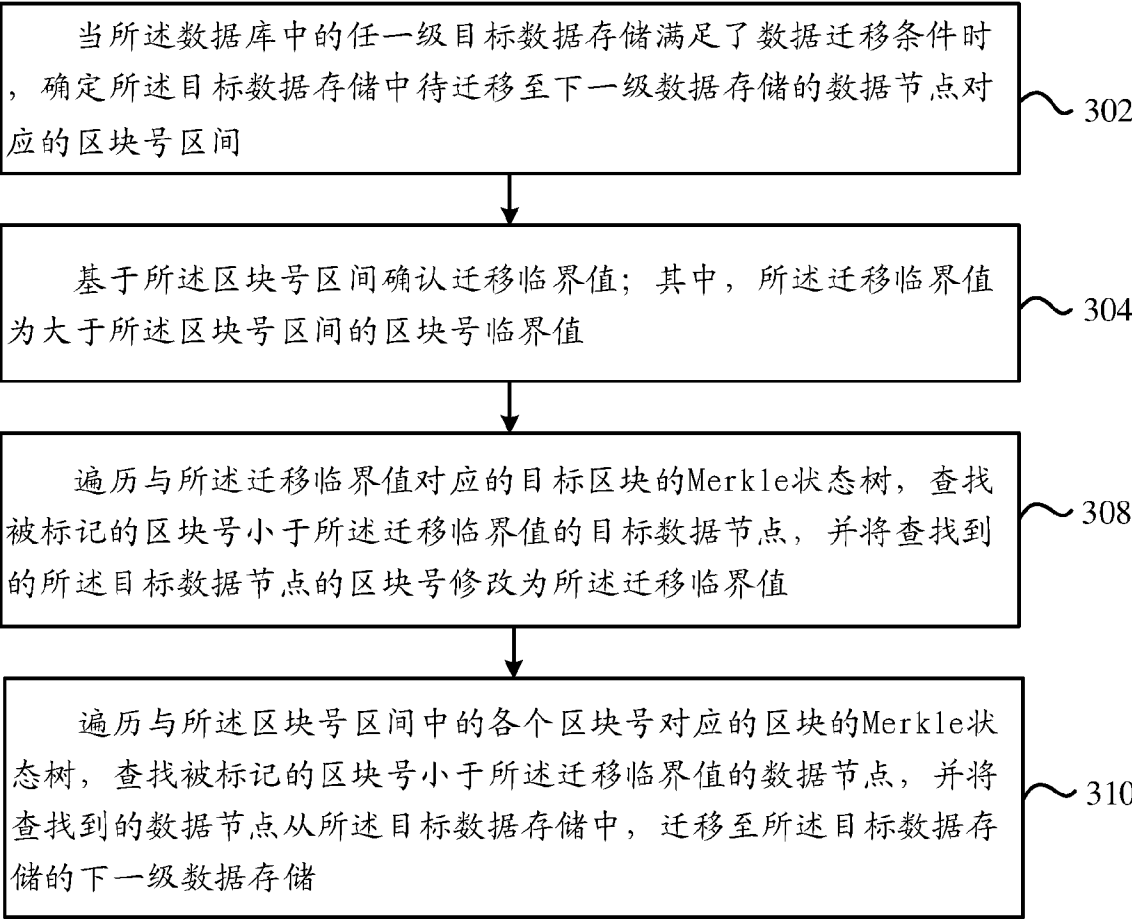


图 3

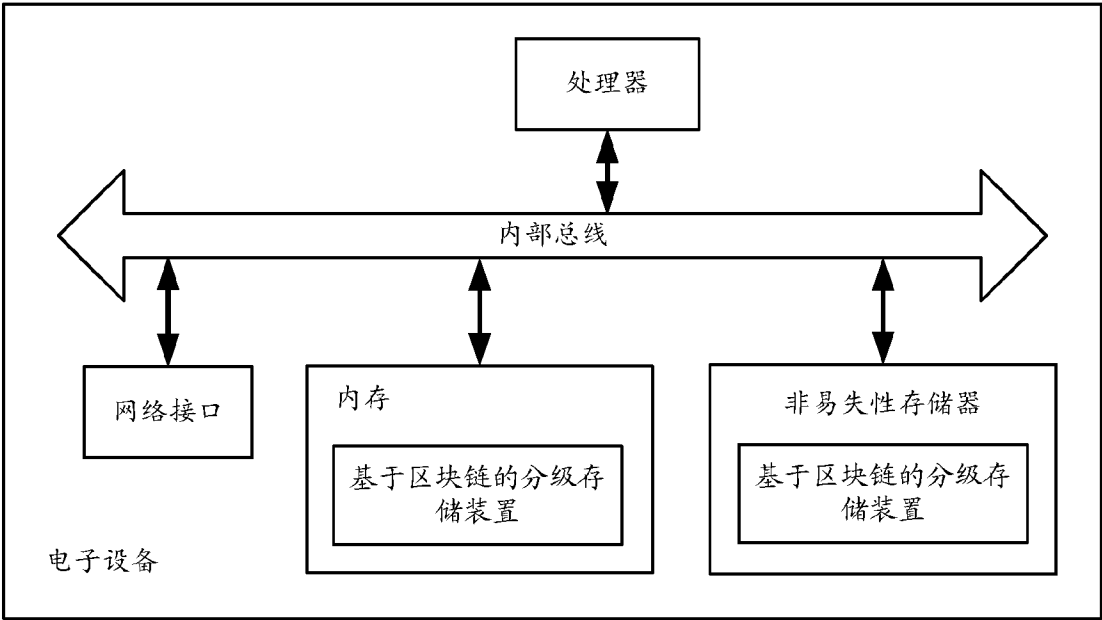


图 4



图 5

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2020/072030

A. CLASSIFICATION OF SUBJECT MATTER

G06F 16/21(2019.01)i; G06F 16/22(2019.01)i; G06F 16/27(2019.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

G06F

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNPAT, WPI, EPODOC, CNKI, IEEE: 区块链, 状态, 分级, 多级, 存储, 默克尔, 区块号, 迁移, 临界值, 阈值, 键, block, chain, status, hierarchical, multi w level, storage, Merkle, Merkel, MPT, identify, migration, threshold, key

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
PX	CN 110347660 A (ALIBABA GROUP HOLDING LIMITED) 18 October 2019 (2019-10-18) claims 1-24	1-24
PX	CN 110334154 A (ALIBABA GROUP HOLDING LIMITED) 15 October 2019 (2019-10-15) claims 1-20	1-24
Y	CN 109144414 A (ALIBABA GROUP HOLDING LIMITED) 04 January 2019 (2019-01-04) description, paragraphs [0045]-[0062] and [0091]-[0092]	1-24
Y	CN 109684333 A (HANGZHOU FUZAMEI TECHNOLOGY CO., LTD.) 26 April 2019 (2019-04-26) description, paragraphs [0002], [0037]	1-24
A	CN 109559234 A (HANGZHOU FUZAMEI TECHNOLOGY CO., LTD.) 02 April 2019 (2019-04-02) entire document	1-24
A	CN 109756573 A (SUZHOU LIANDU CULTURE MEDIA CO., LTD.) 14 May 2019 (2019-05-14) entire document	1-24
A	CN 108282474 A (SHANDONG UNIVERSITY) 13 July 2018 (2018-07-13) entire document	1-24



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

“A” document defining the general state of the art which is not considered to be of particular relevance

“E” earlier application or patent but published on or after the international filing date

“L” document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

“O” document referring to an oral disclosure, use, exhibition or other means

“P” document published prior to the international filing date but later than the priority date claimed

“T” later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

“X” document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

“Y” document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

“&” document member of the same patent family

Date of the actual completion of the international search

28 March 2020

Date of mailing of the international search report

13 April 2020

Name and mailing address of the ISA/CN

China National Intellectual Property Administration (ISA/
CN)
No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing
100088
China

Facsimile No. (86-10)62019451

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2020/072030

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	110347660	A	18 October 2019	None			
CN	110334154	A	15 October 2019	None			
CN	109144414	A	04 January 2019	US	2020034311	A1	30 January 2020
				WO	2020023843	A1	30 January 2020
CN	109684333	A	26 April 2019	None			
CN	109559234	A	02 April 2019	None			
CN	109756573	A	14 May 2019	None			
CN	108282474	A	13 July 2018	None			

国际检索报告

国际申请号

PCT/CN2020/072030

A. 主题的分类

G06F 16/21(2019.01)i; G06F 16/22(2019.01)i; G06F 16/27(2019.01)i

按照国际专利分类(IPC)或者同时按照国家分类和IPC两种分类

B. 检索领域

检索的最低限度文献(标明分类系统和分类号)

G06F

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库(数据库的名称, 和使用的检索词(如使用))

CNPAT, WPI, EPDOC, CNKI, IEEE: 区块链, 状态, 分级, 多级, 存储, 默克尔, 区块号, 迁移, 临界值, 阈值, 键, block, chain, status, hierarchical, multi w level, storage, Merkle, Merkel, MPT, identify, migration, threshold, key

C. 相关文件

类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求
PX	CN 110347660 A (阿里巴巴集团控股有限公司) 2019年 10月 18日 (2019 - 10 - 18) 权利要求1-24	1-24
PX	CN 110334154 A (阿里巴巴集团控股有限公司) 2019年 10月 15日 (2019 - 10 - 15) 权利要求1-20	1-24
Y	CN 109144414 A (阿里巴巴集团控股有限公司) 2019年 1月 4日 (2019 - 01 - 04) 说明书第[0045]-[0062], [0091]-[0092]段	1-24
Y	CN 109684333 A (杭州复杂美科技有限公司) 2019年 4月 26日 (2019 - 04 - 26) 说明书第[0002], [0037]段	1-24
A	CN 109559234 A (杭州复杂美科技有限公司) 2019年 4月 2日 (2019 - 04 - 02) 全文	1-24
A	CN 109756573 A (苏州链读文化传媒有限公司) 2019年 5月 14日 (2019 - 05 - 14) 全文	1-24
A	CN 108282474 A (山东大学) 2018年 7月 13日 (2018 - 07 - 13) 全文	1-24

☐ 其余文件在C栏的续页中列出。☒ 见同族专利附件。

* 引用文件的具体类型:

“A” 认为不特别相关的表示了现有技术一般状态的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件(如具体说明的)

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期

2020年 3月 28日

国际检索报告邮寄日期

2020年 4月 13日

ISA/CN的名称和邮寄地址

中国国家知识产权局(ISA/CN)

中国北京市海淀区蓟门桥西土城路6号 100088

传真号 (86-10)62019451

授权官员

李玉坤

电话号码 86-(10)-53961358

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2020/072030

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	110347660	A	2019年 10月 18日	无			
CN	110334154	A	2019年 10月 15日	无			
CN	109144414	A	2019年 1月 4日	US	2020034311	A1	2020年 1月 30日
				WO	2020023843	A1	2020年 1月 30日
CN	109684333	A	2019年 4月 26日	无			
CN	109559234	A	2019年 4月 2日	无			
CN	109756573	A	2019年 5月 14日	无			
CN	108282474	A	2018年 7月 13日	无			