

(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

(43) 国际公布日
2020 年 7 月 2 日 (02.07.2020)



(10) 国际公布号
WO 2020/134115 A1

(51) 国际专利分类号:
G06F 16/22 (2019.01)

(21) 国际申请号: PCT/CN2019/100495

(22) 国际申请日: 2019 年 8 月 14 日 (14.08.2019)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
201811605446.1 2018 年 12 月 26 日 (26.12.2018) CN

(71) 申请人: 百度在线网络技术(北京)有限公司(BAIDU ONLINE NETWORK TECHNOLOGY (BEIJING) CO., LTD.) [CN/CN]; 中国北京市海淀区上地十街 10 号百度大厦三层, Beijing 100085 (CN)。

(72) 发明人: 肖伟(XIAO, Wei); 中国北京市海淀区上地十街 10 号百度大厦三层, Beijing 100085 (CN)。

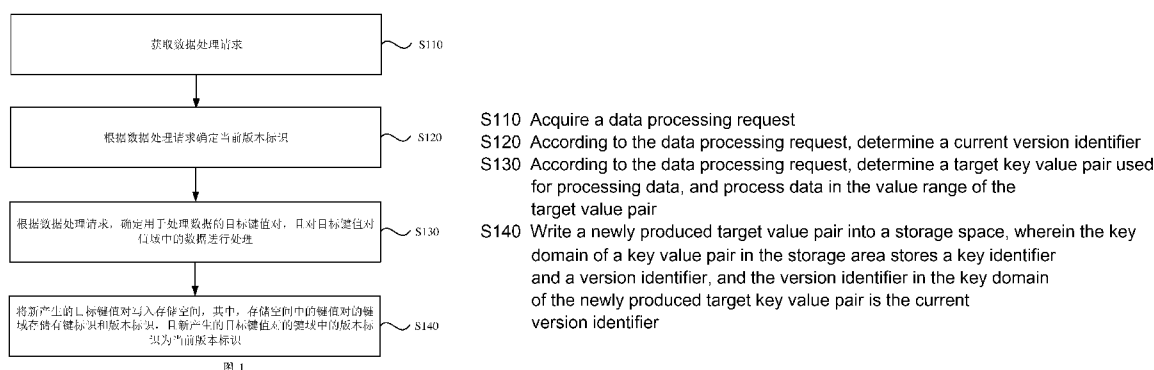
(74) 代理人: 北京品源专利代理有限公司(BEYOND ATTORNEYS AT LAW); 中国北京市海淀区莲花池东路 39 号西金大厦 6 层, Beijing 100036 (CN)。

(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, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW。

(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,

(54) Title: DATA PROCESSING METHOD, APPARATUS AND DEVICE, AND STORAGE MEDIUM

(54) 发明名称: 一种数据存储方法、装置、设备及存储介质



(57) Abstract: Disclosed in embodiments of the present application are a data processing method, apparatus, and device, and a storage medium. The method comprises: acquiring a data processing request; according to the data processing request, determining a current version identifier; according to the data processing request, determining a target key value pair used for processing data, and processing data in the value range of the target value pair; writing a newly produced target value pair into a storage space, wherein the key domain of a key value pair in the storage area stores a key identifier and a version identifier, and the version identifier in the key domain of the newly produced target key value pair is the current version identifier.

(57) 摘要: 本申请实施例公开了一种数据存储方法、装置、设备及存储介质。其中, 该方法包括: 获取数据处理请求; 根据所述数据处理请求确定当前版本标识; 根据所述数据处理请求, 确定用于处理数据的目标键值对, 且对所述目标键值对值域中的数据进行处

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))。

一种数据存储方法、装置、设备及存储介质

本申请要求在2018年12月26日提交中国专利局、申请号为201811605446.1的中国专利申请的优先权，该申请的全部内容通过引用结合在本申请中。

技术领域

本申请实施例涉及计算机数据存储技术，例如涉及一种数据存储方法、装置、设备及存储介质。

背景技术

目前的区块链系统所采用的底层数据存储技术有多种，其中比较广泛采用的一种就是键值对（Key Value，简称KV）存储系统。KV存储系统在区块链系统上层软件的控制下，支持区块链系统的数据读写等访问处理。

在KV存储系统中，会使用到很多KV分别存储区块链系统所需存储的各种数据，有些KV会随着区块的产生而需要修改。例如，在区块链网络中发起事务处理请求后，需要区块链网络中的节点执行事务处理请求中的智能合约，智能合约的执行需要基于一些已有数据，同时也需要修改一些已有数据。如对于转账功能的智能合约，需要读取转入账户和转出账户的KV，其中，KV的键域（key）可以存储账户名称，值域（value）可以存储账户余额的数值；在智能合约运行后，会根据转账数额来修改转入账户和转出账户的value值。实际上，区块产生过程会涉及大量KV的变化。

目前区块链节点会为KV存储系统建立索引，以方便访问数据，但是索引会由于KV的变化而发生大量变化，这通常由KV存储系统本身采用的索引方式所决定。由于区块链系统上层软件的数据处理需要，会产生新增KV、删除KV或修改KV的调整，则相应的需要调整KV的索引。

如果出现区块链的分叉而导致某些区块链节点本地存储的区块链数据要进行回滚和重做时，回滚操作要把底层KV存储系统中的存储数据逐步还原到分叉点，然后再在新的区块链里面执行重做命令，即重新执行所有事务的业务逻辑，产生的区块数据再插入到KV存储系统中，这势必导致巨大的资源开销。

因此，由于区块链系统的数据处理特点，导致采用已有KV存储系统来支撑区块链节点的数据存储时，性能不佳。

发明内容

以下是对本文详细描述的主题的概述。本概述并非是为了限制权利要求的保护范围。

本申请实施例提供一种数据存储方法、装置、设备及存储介质，以优化 KV 存储系统的存储方式，以改善存储系统的性能。

本申请实施例提供了一种数据存储方法，包括：获取数据处理请求；根据所述数据处理请求确定当前版本标识；根据所述数据处理请求，确定用于处理数据的目标键值对，且对所述目标键值对值域中的数据进行处理；将新产生的目标键值对写入存储空间，其中，存储空间中的键值对的键域存储有键标识和版本标识，且新产生的目标键值对的键域中的版本标识为当前版本标识。

本申请另一实施例还提供了一种数据存储装置，包括：处理请求获取模块，设置为获取数据处理请求；当前标识确定模块，设置为根据所述数据处理请求确定当前版本标识；数据处理模块，设置为根据所述数据处理请求，确定用于处理数据的目标键值对，且对所述目标键值对值域中的数据进行处理；写入模块，设置为将新产生的目标键值对写入存储空间，其中，存储空间中的键值对的键域存储有键标识和版本标识，且新产生的目标键值对的键域中的版本标识为当前版本标识。

本申请另一实施例还提供了一种设备，包括：一个或多个处理器；存储装置，设置为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，使得所述一个或多个处理器实现本申请任意实施例所述的数据存储方法。

本申请另一实施例还提供了一种存储介质，其上存储有计算机程序，该程序被处理器执行时实现本申请任意实施例所述的数据存储方法。

在阅读并理解了附图和详细描述后，可以明白其他方面。

附图说明

- 图 1 是本申请实施例一中提供的一种数据存储方法的流程图；
- 图 2 是本申请实施例二中提供的一种数据存储方法的流程图；
- 图 3 是本申请实施例三中提供的一种数据存储方法的流程图；
- 图 4 是本申请实施例四中提供的一种数据存储方法的流程图；
- 图 5A 是本申请实施例五中提供的一种数据存储方法的流程图；
- 图 5B 是本申请实施例五中提供的一种区块链分叉示意图；
- 图 6 是本申请实施例六中提供的一种数据存储方法的流程图；
- 图 7 是本申请实施例七中提供的一种数据存储装置的结构示意图；
- 图 8 是本申请实施例八中提供的一种设备的结构示意图。

具体实施方式

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本申请，而非对本申请的限定。另外还需要说明的是，为了便于描述，附图中仅示出了与本申请相关的部分而非全部结构。

实施例一

图 1 为本申请实施例一提供的一种数据存储方法的流程图，本实施例可适用于 KV 存储系统作为底层存储系统时，对上层软件所下发的数据进行处理的情况。尤其是，该 KV 存储系统在支撑区块链系统的数据存储时，能与区块链系统的数据处理特点更加匹配。该方法可以由数据存储装置来执行，该装置可采用软件和/或硬件的方式实现，并可集成于承载计算设备中，特别是作为区块链节点的计算设备中。参见图 1，该方法具体包括：

S110，获取数据处理请求。

本实施例中，数据处理请求是指用于对数据进行处理请求；具体的，数据处理请求为上层软件定时、达到一定数据量或接收到某一触发机制等任何需要处理数据的时机，向底层存储系统所发送的用于请求底层存储系统对待处理数据进行写入、更改、读取或删除等某种处理操作。本申请实施例尤其适用于区块链系统的数据处理，下述实施例将以区块链数据的处理为例进行说明。但本领域技术人员可以理解，本申请实施例的技术方案不限于使用在区块链系统中。

可选的，数据处理请求中可以包括待处理的目标数据；目标数据可以为区块的数据、事务数据或其他的业务数据或管理数据等，事务数据是在区块链系统对发起的事务请求进行处理的数据。区块的数据可以包括存储在区块内的数据，以及依托于区块链但独立于区块链存储在区块链节点本地的数据。数据处理请求中还可以包括目标数据标识，如区块标识或事务标识等，区块标识是指用于唯一识别某一区块的标志，如区块 ID；事务标识是指用于唯一识别某一类或某一个事务数据的标志。

S120，根据数据处理请求确定当前版本标识。

本实施例中，版本标识是一个起到唯一标识作用的标识符，用于标识对当前数据处理请求所涉及数据的更新操作。

当前版本标识是指本次数据处理请求需处理的数据所使用的版本标识。本次数据处理请求可以是一个或多个。一个版本标识可以适用于一个数据处理请求所涉及的数据，也可以适用于一批数据处理请求所涉及的数据。具体的，可以是不同的区块中所产生的数据处理请求具有不同的版本标识，也就是说，同一区块的处理过程中所产生的数据处理请求的涉及数据具有相同的版本标识。或者，还可以是一个事务数据的处理过程中所产生的数据处理请求的涉及数据具有一个版本标识。区块或事务数据的处理过程所涉及的数据可能有很多，例

如，事务请求的数据需要存储，则事务请求数据是写入请求涉及的数据；由于执行某个事务请求的智能合约，导致某个账户的账户余额发生了变化，则该账户余额是写入请求涉及的数据。

版本标识可以是随机生成的，还可以是按设定规则生成的，如可以是流水号或流水号+区块标识/事务标识等。为了便于后续查询、更改等操作，本实施例中优选按设定规则生成版本标识。设定规则例如是，版本标识为整数值，确定新的版本标识是在前一版本标识的基础上加一。可选的，数据处理请求中还可以包括前一版本标识。因此，可以从数据处理请求中提取前一版本标识，而后依据前一版本标识生成当前版本标识。

S130，根据数据处理请求，确定用于处理数据的目标键值对，且对目标键值对值域中的数据进行处理。

本实施例中，键值对的键域可以用于存储数据的键标识和版本标识，键标识可以为能体现数据实际含义的标识，如区块标识或事务标识等；键值对的值域可以用于存储数据本身。目标键值对用于存储数据处理请求的目标数据，支持数据处理，可以是数据处理请求中目标数据所在的已有键值对，或新产生的键值对。

若数据处理请求为对目标数据进行读取操作，则可以将数据处理请求中目标数据所在的已有键值对，作为目标键值对。示例性的，根据数据处理请求，确定用于处理数据的目标键值对可以包括：根据数据处理请求中目标数据的键标识和版本标识，确定目标数据所在的已有键值对，作为目标键值对。本实施例中，版本标识是指所需读取数据的版本标识，可以是前一版本标识，还可以是历史任一版本标识。具体的，可以以目标数据的键标识和版本标识为索引，在已有键值对中进行查找，将匹配的已有键值对作为目标键值对。

若数据处理请求为对目标数据的写入操作，或对已有键值对中值域的更改或删除，则可以将新产生的键值对作为目标键值对。需要说明的是，本实施例中，在需要对数据进行删除或更改时，在新产生的键值对中进行操作，而保留已有键值对，这样在需要切换到之前其他版本标识的键值对数据进行查看时，不需要进行大量键值对的数据恢复，切换速度快。因此，在需要对数据进行删除或更改时，还需基于数据处理请求中目标数据的键标识和版本标识，确定目标数据所在的已有键值对，而后基于已有键值对及目标键值对对目标数据进行删除或更改操作。

通常，上层软件能够确定需要访问处理的目标数据，所以能够通过数据处理请求传递对应键标识和版本标识的信息。

具体的，在根据数据处理请求确定当前版本标识之后，可以根据数据处理请求确定用于处理数据的目标键值对；而后依据数据处理请求给定的数据的处

理方式，对目标键值对值域中的数据进行处理。

可选的，数据处理请求给定的数据处理方式，如更改、查询、写入或删除等方式不同，对数据进行处理的策略不同，下述实施例将详细介绍不同的处理方式对应的处理策略。

S140，将新产生的目标键值对写入存储空间，其中，存储空间中的键值对的键域存储有键标识和版本标识，且新产生的目标键值对的键域中的版本标识为当前版本标识。

本实施例中，存储空间是指用于存储目标键值对所需要的存储介质，可以是任何大容量的存储设备，如可以是内存存储空间，也可以是磁盘存储空间等。由于磁盘断电后保存的数据不会丢失，且在不清除的情况下永久保存，而上层软件需要底层存储系统将其所获取的区块的数据或事务数据永久存储，因此，本实施例优选存储空间为磁盘存储空间。为了便于对键值对进行管理及后续查询等，可以将一个区块内的数据所对应的键值对或有关联关系的键值对关联存储。可选的，磁盘存储空间所使用的 KV 存储系统的索引可以为有序表或索引树。其中，有序表即为键值对以递增或递减的方式有序排列；索引树是指将键值对按树状结构排列，索引树可以为二叉索引树，且可以是位于索引树右边的键值对的键标识大于位于索引树左边的键值对的键标识等。

本申请实施例提供的技术方案，通过数据处理请求确定当前版本标识，以及用于处理数据的目标键值对，并对目标键值对值域中的数据进行如写入、更改、删除或读取等处理，且在目标键值对为新产生的目标键值对的情况下，将当前版本标识存储在目标键值对的键域；而后将新产生的目标键值对写入存储空间，进而完成数据的存储。本方案在需要对数据进行写入、更改或删除等处理时，保留已有键值对，在新产生的键值对中进行操作，便于后续基于版本标识快速切换至之前其他版本标识的键值对数据进行查看。特别是，由于区块链系统存在区块回滚和重做的需求，例如发生分叉时需要回滚，那么基于本申请实施例所提供的 KV 存储系统能够方便的切换至之前任何版本的键值对数据，不需要进行大量键值对的数据恢复。相比于现有区块链的数据存储方式而言，提高了版本恢复速度，为区块链的数据存储提供了一种新思路。

实施例二

图 2 为本申请实施例二提供的一种数据存储方法的流程图，本实施例在上述实施例的基础上，提供了一种基于数据对象来执行的具体方案。参见图 2，该方法具体包括：

S210，获取数据处理请求。

S220，根据数据处理请求确定前一版本数据对象。

本实施例中，数据对象也可称为句柄或指针，可以是底层存储系统对一个或一批数据处理请求进行处理的虚标识，该虚标识用于在程序执行的过程中能方便使用计算机设备的硬件资源和软件构建的虚拟资源，主要是能够以虚标识对应存储资源，表明此数据对象需要使用某个存储系统以及具体的存储空间。具体的，某个数据对象的主程序在运行的过程中，可调用相应的功能函数对数据处理请求中的数据进行处理。该数据对象的主程序和功能函数所针对的存储系统是相同的。

本申请实施例中，数据对象与版本标识一一对应，不同的版本标识，所对应的版本数据对象不同。可选的，本实施例中，数据对象的名称可以包括版本标识。前一版本数据对象为生成当前版本数据对象所需的旧版本数据对象，由上层软件决定，实际上，可以由上层软件确定前一版本的版本标识。本申请实施例的各个版本数据对象所对应的存储系统都是全部键值对存储系统，其中包括了各个版本的键值对。

S230，基于前一版本数据对象，产生当前版本数据对象，并确定当前版本标识，其中，当前版本数据对象与前一版本数据对象所操作的存储空间相同。

具体的，将前一版本数据对象的名称，例如前一版本标识，作为数据对象主程序的输入参数，例如由版本生成函数来基于前一版本标识生成当前版本标识，进而产生当前版本数据对象。

例如，若版本生成函数为 `tree.version()`，前一版本数据对象的名称即前一版本标识为 `old_version`，`ctx` 为当前版本数据对象的名称，即当前版本标识可以表示为：`ctx=tree.version(old_version)`。

可选的，若数据处理请求为当前区块产生过程中涉及的数据处理请求，则在生成当前版本标识时，前一版本标识可以为前一区块的版本标识；若数据处理请求为当前事务数据产生过程中涉及的数据处理请求，在为该数据处理请求生成当前版本标识时，前一版本标识可以为前一事务数据的版本标识。

可选的，不同数据对象所操作的存储空间均相同，但不同数据对象调用相同功能函数时，向该功能函数中所传递的参数不同，可选的，向功能函数所传递的参数需包括版本标识，其中，版本标识为数据对象所对应的当前版本标识；还可以包括目标数据及目标数据标识等。例如，若数据处理请求为目标数据的写入操作，则当前版本数据对象可以调用 `ctx.insert(目标数据, 目标数据标识, new_version)`，其中，`new_version` 是当前版本数据对象在调用 `ctx.insert()` 函数时传递的参数，是当前版本数据对象所对应的当前版本标识。

此外，由于不同数据对象所操作的存储空间均相同，因此在基于前一版本数据对象，产生当前版本数据对象时，也可默认维护一个版本标识，无需将前一版本数据对象的版本标识传入。

S240, 根据数据处理请求, 确定用于处理数据的目标键值对, 且对目标键值对值域中的数据进行处理。

具体的, 在根据数据处理请求确定当前版本标识之后, 可以根据数据处理请求确定用于处理数据的目标键值对; 而后依据数据处理请求调用相应的功能函数, 对目标键值对进行处理。

示例性的, 根据数据处理请求, 确定用于处理数据的目标键值对, 且对目标键值对值域中的数据进行处理还可以包括: 在当前数据对象中, 根据数据处理请求调用相应数据操作的功能函数, 将当前版本数据对象的当前版本标识作为功能函数的输入参数, 以执行数据处理请求。

具体的, 若数据处理请求为对目标数据进行查询或读取操作, 则可以在确定前一或历史任一版本数据对象后, 将数据处理请求中目标数据所在的已有键值对, 作为目标键值对; 而后基于前一或历史任一版本数据对象调用查询函数, 将目标键值对中的键域(键标识及前一或历史任一版本标识)作为查询函数的输入参数, 并执行该查询函数, 从目标键值对的值域读取目标数据并反馈。

若数据处理请求为对目标数据的写入操作, 则可以将新产生的键值对作为目标键值对。而后采用当前版本数据对象调用插入函数, 并将目标数据、目标数据标识及当前版本标识等作为插入函数的输入参数, 执行该插入函数, 得到目标键值对的键域存储有目标数据标识(作为键标识)和当前版本标识, 目标键值对的值域存储有目标数据。

若数据处理请求为对已有键值对中值域的更改操作, 则可以将新产生的键值对作为目标键值对。而后采用当前版本数据对象调用更改函数, 并将目标数据、已有键值对中的键域(键标识及前一版本标识)及当前版本标识等作为更改函数的输入参数, 执行该更改函数, 得到目标键值对的键域存储有已有键值对的键标识和当前版本标识, 目标键值对的值域存储有目标数据。

若数据处理请求为对已有键值对中值域的删除操作, 则可以将新产生的键值对作为目标键值对。而后采用当前版本数据对象调用删除函数, 并将目标数据、已有键值对中的键域(键标识及前一版本标识)及当前版本标识等作为删除函数的输入参数, 执行该删除函数, 得到目标键值对的键域存储有已有键值对的键标识和当前版本标识, 目标键值对的值域为空。

S250, 调用生效控制指令, 以控制当前版本数据对象生效, 将当前版本数据对象中新产生的目标键值对写入存储空间。

其中, 存储空间中的键值对的键域存储有键标识和版本标识, 且新产生的目标键值对的键域中的版本标识为当前版本标识。

生效控制指令用于控制当前版本数据对象生效, 也即控制当前版本数据对象所处理的所有数据处理请求生效, 进而基于当前版本标识所产生的目标键值

对生效。基于数据对象对数据处理请求的处理，可根据实际缓存的存储容器进行设置；可以是在缓存中完成，在确认生效后，则统一将这些数据处理请求所更新过的键值对，存储到持久化存储空间中。可以是将新键值对从内存拷贝至磁盘空间，如果新键值对已经在磁盘的缓存空间中，则可以控制使新键值对生效为持久化存储即可。

可选的，底层存储系统可以在执行完步骤 S240 之后，等待预设时段后，自动调用生效控制指令；还可以是接收到某一触发操作，则调用生效指令等。示例性的，调用生效控制指令可以包括：在产生当前版本生效条件，或接收到上层软件发送的当前版本生效指令时，调用生效控制指令。

其中，当前版本生效条件是指当前版本生效所需满足的条件，可以是上层软件向底层存储系统下发新的数据处理请求，还可以是等待时间达到了自动生效条件等。可选的，产生当前版本生效条件包括下述至少一项：1) 待处理数据的区块标识发生变更；2) 待处理数据的事务标识发生变更；3) 当前时间满足生效时间间隔等。本实施例中，当前版本生效指令是指上层软件所发送的，用于表明当前版本数据对象生效的指令，可以包括当前版本标识，或者数据处理请求中的数据标识如区块标识或事务标识等。

具体的，当底层存储系统接收到上层软件下发的新的数据处理请求，即可以确定待处理数据的区块标识或事务标识发生变更，说明前一区块或前一事务数据已经处理完毕，则调用生效控制指令；也可以是当前数据处理请求执行完的时间戳至当前时间戳之间的时间间隔大于或等于生效时间间隔，则自动调用生效控制指令；还可以底层存储系统接收到上层软件发生的当前版本生效指令，则调用生效控制指令等。而后控制当前版本数据对象生效，将在当前版本数据对象中新产生的目标键值对写入持久化存储空间。

需要说明的是，本实施例中基于生效控制指令限定了键值对或数据写入存储空间的权限，更有利于对存储空间的管控。

在底层存储系统未调用生效控制指令时，若接收到上层软件发送的版本标识修改指令，则可以先依据版本标识修改指令修改当前版本标识，并同步更新当前版本数据对象中键值对键域中的当前版本标识；而后控制当前版本数据对象生效。示例性的，控制当前版本数据对象生效之前，还可以包括：根据上层软件传输的版本标识修改指示，修改当前版本标识，并更新键值对键域中的当前版本标识。

本实施例中，版本标识修改指令是指上层软件所发送的，用于修改当前版本数据对象中新产生的键值对键域中的版本标识的指令。版本标识中一般可以包括流水号和其他具有逻辑含义的标识，例如区块标识和事务标识，或者是时间标识等作为附加版本标识。上层软件优选是可以修改这些附加版本标识。

具体的，版本标识修改指令可以包括当前版本标识、修改前的区块标识或事务标识、修改后的区块标识或事务标识等；或者还可以包括修改后的版本标识。可选的，版本标识修改指令用于修改组成版本标识的区块标识或事务标识等。

具体的，当上层软件确认向底层存储系统所发送的数据处理请求中包括的目标数据标识如区块标识有误时，可以向底层存储系统发送包括修改前的区块标识和修改后的区块标识的版本标识修改指令；底层存储系统接收到该版本修改指令后，依据版本修改指令的指示用修改后的区块标识替换修改前的区块标识，进而完成当前版本标识的修改；而后同步更新当前版本数据对象中键值对键域中的当前版本标识；在基于当前版本标识的键值对修改完成的情况下，且基于数据对象对数据处理请求完成后，调用生效控制指令，控制当前版本数据对象生效。

示例性的，将新产生的目标键值对写入存储空间之前，还可以包括：调用失效控制指令，以控制当前版本数据对象失效。本实施例中，失效控制指令与生效控制指令相对应，用于控制当前版本数据对象失效，也即控制当前版本数据对象所对应的当前版本标识失效，进而基于当前版本标识所产生的目标键值对失效。可选的，底层存储系统在未将新产生的目标键值对写入持久化存储空间之前，若本地设备宕机或掉电等，则自动调用失效控制指令；还可以是接收到某一触发操作，则调用失效控制指令。失效控制指令可以使内存中的新键值对失效，或者删除作为缓存的磁盘空间中的新键值对。示例性的，调用失效控制指令可以包括：在产生当前版本失效条件，或接收到上层软件发送的当前版本失效指令时，调用失效控制指令。

其中，当前版本失效条件是指确定当前版本失效所需满足的条件，可以是本地设备宕机、掉电或受到外界因素攻击等导致新产生的目标键值对未处理完成等。当前版本失效指令是指上层软件所发送的，用于表明当前版本数据对象失效的指令，可以包括当前版本标识，或者数据处理请求中的数据标识如区块标识或事务标识等。

具体的，当底层存储系统接收到上层软件发送的当前版本失效指令，或在本地设备宕机或掉电等时，调用失效控制指令，以控制当前版本数据对象失效，进而当前版本数据对象中新产生的目标键值对失效，因此可直接丢弃缓存中所更新过的键值对。也就是说，若在将新产生的目标键值对写入持久化存储空间之前，底层存储系统调用了失效控制指令，则结束本次数据处理请求，可以向上层软件反馈数据处理请求失败信息等。持久化存储空间通常是磁盘。

本申请实施例提供的技术方案，在获取数据处理请求之后，提供了一种基于版本数据对象确定当前版本标识的方案；在确定当前版本标识之后，根据数

据处理请求确定用于处理数据的目标键值对，并对目标键值对值域中的数据进行如写入、更改、删除或读取等处理，且在目标键值对为新产生的目标键值对的情况下，将当前版本标识存储在目标键值对的键域；而后调用生效控制指令，以控制当前版本数据对象生效，将当前版本数据对象中新产生的目标键值对写入存储空间。本方案在需要对数据进行写入、更改或删除等处理时，保留已有键值对，在新产生的键值对中进行操作，便于后续基于版本标识快速切换至之前其他版本标识的键值对数据进行查看。特别是，由于区块链系统存在区块回滚和重做的需求，例如发生分叉时需要回滚，那么基于本申请实施例所提供的KV存储系统能够方便的切换至之前任何版本的键值对数据，不需要进行大量键值对的数据恢复，优化了区块链KV存储系统的索引方式，相比于现有区块链的数据存储方式而言，为区块链的数据存储提供了一种新思路。此外，基于生效控制指令限定了键值对或数据写入存储空间的权限，更有利于对存储空间的管控。

实施例三

图3为本申请实施例三提供的一种数据存储方法的流程图，本实施例在上述实施例的基础上，又提供了一种根据数据处理请求确定当前版本标识的方案。参见图3，该方法具体包括：

S310，获取数据处理请求。

S320，确定前一版本标识，按照设定编号规则产生当前版本标识。

或，S320的操作也可以具体是：确定前一版本标识，根据前一版本标识的流水号标识，按照设定编号规则产生当前版本标识中的流水号标识，以及根据数据处理请求确定的版本附加标识，与流水号标识组成当前版本标识。

本实施例中，流水号标识即为流水号值；设定编号规则是指预先设定的，版本标识序号或流水号标识的确定规则；可选的，设定编号规则可以为将前一区块个数或事务数据个数加1或减1作为编号值或流水号标识；由于数据处理请求中的目标数据可以是区块的数据或事务数据，且一个数据处理请求中的目标数据对应一个版本标识，因此，当前版本标识可以为前一版本标识加预设整数值如1或减1得到；当前版本标识中的流水号标识可以为前一版本标识的流水号标识加1或减1得到。示例性的，版本附加标识为区块标识或事务标识。

前一版本标识为生成当前版本标识所需的旧版本标识，可以是前一版本数据对象的名称。数据处理请求中可以包括待处理的目标数据、目标数据标识及前一版本标识等，因此，可从数据处理请求中获取前一版本标识。示例性的，确定前一版本标识可以包括：从数据处理请求中提取前一版本标识；或根据数据处理请求中的业务数据标识确定前一版本标识。

其中，业务数据即为待处理的目标数据，业务数据标识为区块标识或事务标识。具体的，可以从数据处理请求中直接获取前一版本标识；还可以根据数据处理请求中业务数据标识即目标数据标识确定前一版本的区块标识或事务标识作为版本附加标识，并根据目标数据确定当前区块个数或事务数据个数，进而依据当前区块个数或事务数据个数减 1 或加 1 得到前一版本标识流水号，而后依据版本附加标识及流水号标识，确定前一版本标识。

具体的，在确定前一版本标识后，可以基于前一版本标识及设定编号规则产生当前版本标识；也可以是基于前一版本标识的流水号标识及设定编号规则产生当前版本标识的流水号标识，并从数据处理请求中提取目标数据标识作为版本附加标识，将流水号标识及版本附加标识作为当前版本标识。

S330，根据数据处理请求，确定用于处理数据的目标键值对，且对目标键值对值域中的数据进行处理。

S340，将新产生的目标键值对写入存储空间，其中，存储空间中的键值对的键域存储有键标识和版本标识，且新产生的目标键值对的键域中的版本标识为当前版本标识。

本申请实施例提供的技术方案，在获取数据处理请求之后，提供了一种基于设定编号规则确定当前版本标识的方案，增加了方案的灵活度；在确定当前版本标识之后，根据数据处理请求确定用于处理数据的目标键值对，并对目标键值对值域中的数据进行处理如写入、更改、删除或读取等处理，且在目标键值对新产生的目标键值对的情况下，将当前版本标识存储在目标键值对的键域；而后将新产生的目标键值对写入存储空间，进而完成数据的存储。本方案在需要对数据进行写入、更改或删除等处理时，保留已有键值对，在新产生的键值对中进行操作，便于后续基于版本标识快速切换至之前其他版本标识的键值对数据进行查看，特别是，由于区块链系统存在区块回滚和重做的需求，例如发生分叉时需要回滚，那么基于本申请实施例所提供的 KV 存储系统能够方便的切换至之前任何版本的键值对数据，不需要进行大量键值对的数据恢复，优化了区块链 KV 存储系统的索引方式，相比于现有区块链的数据存储方式而言，为区块链的数据存储提供了一种新思路。

实施例四

图 4 为本申请实施例四提供的一种数据存储方法的流程图，本实施例在上述实施例的基础上，进一步对根据数据处理请求，确定用于处理数据的目标键值对，且对目标键值对值域中的数据进行处理进行解释说明。参见图 4，该方法具体包括：

S410，获取数据处理请求。

S420, 根据数据处理请求确定当前版本标识。

S430, 判断数据处理请求中指定对目标数据的操作方式。如果数据处理请求为目标数据的写入操作, 则执行步骤 S440; 如果数据处理请求为将已有键值对值域数据更改为目标数据, 则执行步骤 S450; 如果数据处理请求为已有键值对的删除操作, 则执行步骤 S460; 如果数据处理请求为已有键值对的读取操作, 则执行步骤 S470。

需要说明的是, 一个数据处理请求对目标数据的操作方式至少涉及写入操作、删除操作、读取操作及更改操作中的一种。

S440, 产生新键值对作为目标键值对, 在目标键值对的键域写入新的键标识和当前版本标识, 在目标键值对的值域写入目标数据, 并执行步骤 S480。

本实施例中, 新的键标识是指区别于已有键值对的键标识, 也就是说第一次出现的键标识。

具体的, 若数据处理请求为对目标数据的写入操作, 则可以产生新键值对作为目标键值对; 并从数据处理请求中获取目标数据标识如区块标识或事务标识等, 作为新的键标识; 而后将新的键标识和当前版本标识写在目标键值对的键域, 将目标数据写入目标键值对的值域。

例如, 某一新账户第一次执行转账操作, 目标数据为转账事务数据(转账金额), 键标识可以为账户名, 对目标数据的处理为将转账事务数据写入键值对的值域。

S450, 产生新键值对作为目标键值对, 在目标键值对的键域写入已有键值对的键标识和当前版本标识, 在目标键值对的值域写入目标数据, 并执行步骤 S480。

具体的, 如果数据处理请求为将已有键值对值域数据更改为目标数据, 则产生新键值对作为目标键值对, 并根据数据处理请求中目标数据的键标识和前一版本标识, 确定目标数据所在的已有键值对; 而后将已有键值对的键标识作为目标键值对的键标识, 并将该键标识和当前版本标识写入目标键值对的键域, 将目标数据写入目标键值对的值域。

可选的, 如果数据处理请求为在已有键值对值域数据基础上增加或减去目标数据, 则可以产生新键值对作为目标键值对, 并根据数据处理请求中目标数据的键标识和前一版本标识, 确定目标数据所在的已有键值对; 而后将已有键值对的键标识作为目标键值对的键标识, 并将该键标识和当前版本标识写入目标键值对的键域; 获取已有键值对值域数据, 将采用值域数据加或减去目标数据的结果写入目标键值对的值域。

需要说明的是, 本实施例中, 在确定数据处理请求为对已有键值对值域数据的更改时, 通过新增加一个 KV 存储更改数据, 相比于在原 KV 的基础上进

行修改，保留已有键值对，便于后续基于版本标识快速切换至之前其他版本标识的键值对数据进行查看。

S460，产生新键值对作为目标键值对，在目标键值对的键域写入已有键值对的键标识和当前版本标识，将目标键值对的值域设置为空，并执行步骤 S480。

具体的，如果数据处理请求为已有键值对的删除操作，则产生新键值对作为目标键值对，并根据数据处理请求中目标数据的键标识和版本标识，确定目标数据所在的已有键值对；而后将已有键值对的键标识作为目标键值对的键标识，并将该键标识和当前版本标识写入目标键值对的键域，将目标键值对的值域设置为空。已有键值对仍然保留，但由于上层软件中已经获知当前版本标识对应的键值对是有效的，因此可以基于当前版本标识访问到相关数据，相对的已有键值对可以确认是无效的，不会再被上层软件访问，相当于进行了删除操作。但是当上层软件需要重新使用已经无效的已有键值对时，只需要更改访问数据时使用的版本标识，就可以恢复对已有键值对的数据控制。

S470，确定已有键值对作为目标键值对，将目标键值对值域中的数据进行读取反馈，并执行步骤 S480。

具体的，如果数据处理请求为已有键值对的读取操作或查询操作，则可以将已有键值对作为目标键值对，并根据数据处理请求中目标键值对的键标识和版本标识，确定目标键值对的存储地址，而后依据存储地址定位至目标键值对位置处，并从目标键值对的值域中读取数据，并反馈至上层软件。

S480，将新产生的目标键值对写入存储空间，其中，存储空间中的键值对的键域存储有键标识和版本标识，且新产生的目标键值对的键域中的版本标识为当前版本标识。

本申请实施例提供的技术方案，在根据数据处理请求确定当前版本标识之后，基于数据处理请求所给定的数据操作方式如写入操作、读取操作、删除操作或更改操作等，采用对应的处理策略执行数据处理请求，并将此过程中新产生的目标键值对写入存储空间，进而完成数据的存储。尤其是，当需要对数据进行写入、更改或删除处理时，通过新增加一个 KV 存储写入、更改或删除数据，相比于在原 KV 的基础上进行上述操作，保留已有键值对，便于后续基于版本标识快速切换至之前其他版本标识的键值对数据进行查看，优化了区块链 KV 存储系统的索引方式，为区块链的数据存储提供了一种新思路。

实施例五

图 5A 为本申请实施例五提供的一种数据存储方法的流程图，本申请实施例在上述实施例的基础上，提供了一种应用于区块链系统时，处理分叉情况的方式，有效解决了分叉时需要进行数据回滚和重做的问题，该方法可应用于区

块链节点。参见图 5A，该方法具体包括：

S510，获取数据处理请求。

S520，根据数据处理请求确定当前版本标识。

S530，根据数据处理请求，确定用于处理数据的目标键值对，且对目标键值对值域中的数据进行处理。

S540，将新产生的目标键值对写入存储空间，其中，存储空间中的键值对的键域存储有键标识和版本标识，且新产生的目标键值对的键域中的版本标识为当前版本标识。

S550，根据新产生的目标键值对，在键标识和版本标识的映射关系表中，添加键标识和当前版本标识的映射关系。

本实施例中，底层存储系统预先构建键标识与版本标识的映射关系表，用于供上层软件查询。可选的，键标识与版本标识的映射关系表中，罗列了存储系统中所有键值对的键标识，每个键标识可以对应一个或多个版本标识，版本标识随着键值对的更新而添加。

具体的，在将新产生的目标键值对写入存储空间后，可以将新产生的目标键值对的键标识和版本标识对应存储在映射关系表中。具体为，将新产生的目标键值对的键标识输入至键标识与版本标识的映射关系表中进行匹配，若匹配成功，则说明该目标键值对的键标识已存储，可以将该键标识关联的版本标识添加在映射关系表中该键标识对应位置处。若匹配失败，则说明该目标键值对的键标识未存储，则可以在映射关系表中新添加一个键标识与版本标识的索引对。例如，在不同区块的处理过程中，可能针对同一账户发生了多次转账操作，则该账户的余额会发生多次变化。则用于存储该账户余额的键值对，就具有多个不同的版本标识。

为了便于上层软件快速查询、统计以定位版本标识所对应修改过的键标识（也就是键标识关联的值域修改后），底层存储系统可以构建版本标识和键标识的反向索引表，即一个版本标识对应多个键标识，进而基于版本标识，可快速获取版本标识所对应修改过的键标识。例如，在某个区块的处理过程中，涉及了十个键值对的更新，则对应于该版本标识，记录这十个值域被修改的键值对的键标识。示例性，将新产生的目标键值对写入存储空间的过程中，还可以包括：在版本标识和键标识的反向索引中，增加对应于当前版本标识的新产生目标键值对的键标识。

S560，通过上层软件确定产生区块回滚和重做需求时，获取从其他节点接收的区块数据，以及区块数据的键标识和版本标识的映射关系表。

本实施例中，区块回滚和重做需求可以是在区块链出现分叉时，上层软件所确定的区块链节点本地存储的区块链数据需进行回滚和重做。所谓分叉，是

某个区块链节点本地存储的区块链，末尾的几个区块与其他节点产生的区块不相符，需接受其他节点的区块，而丢弃本地的区块。如图 5B 所示，某个区块链节点本地存储的区块链包括 A、B、C、D 四个区块，通过共识机制，确定 D 区块需要丢弃，应接受其他节点提供的 E、F 这两个区块。则相当于，D 区块需要回滚，E 和 F 区块需要根据从其他节点获取的区块数据来进行重做。

具体的，在区块链节点基于上层软件确定产生区块回滚和重做需求时，可以向其他节点发送区块数据获取请求，以请求其他节点反馈所需的区块数据，以及区块数据的键标识和版本标识的映射关系表；区块链节点接收其他节点反馈的区块数据，以及区块数据的键标识和版本标识的映射关系表。

S570，通过上层软件向本地 KV 存储系统发起数据写入请求，将获取到的区块数据写入本地的存储空间，并将获取的映射关系表添加至本地的映射关系表中。

本实施例中，本地 KV 存储系统即为区块链节点本地的底层存储系统；数据写入请求是指区块链节点基于上层软件向本地 KV 存储系统所发送的，用于请求本地 KV 存储系统将数据写入其存储空间。

具体的，区块链节点可以将其他节点反馈的区块数据的键标识和版本标识的映射关系表，添加至本地映射关系表中的映射关系表中。本地映射关系表中的版本标识和键标识，并不需要删除，而是将获取到映射关系表中的新增内容添加到本地即可。

可以通过上层软件向本地 KV 存储系统发起数据写入请求，以请求本地 KV 存储系统将区块数据写入本地的存储空间中。

其他区块链节点发送的区块数据和映射关系表，可以包括全量数据，也可以仅包括需要重做区块的数据，例如，E 区块和 F 区块涉及的区块数据和版本标识。

需要说明的是，由于本地 KV 存储系统所存储的键值对之间具有空间关联关系，因此修改某一键值对将引起大量已存储键值对的索引需调整。为了减少本地 KV 存储系统调整大量已存储键值对的索引带来的资源开支及工作量，可以在不修改有误索引的情况下，增加正确索引。示例性的，将获取的映射关系表添加至本地的映射关系表中可以包括：将获取的映射表相对于本地映射关系表的增量键标识和版本标识，添加至本地映射关系表中，且保留本地映射关系表中的已有键标识和版本标识，其中，增量版本标识中包括生效区块版本标识，本地映射关系表的已有版本标识中包括失效区块版本标识。

本实施例中，区块链节点本地的映射关系表中，需回滚的区块所关联的区块版本标识为失效区块版本标识；对应的，从其他节点获取的所需区块关联的区块版本标识为生效区块版本标识。具体的，区块链节点在保留本地映射关系

表中已有键标识和版本标识的情况下，将从其他节点获取的映射表相对于本地映射关系表的生效区块版本标识和版本标识，添加至本地映射关系表中。

需要说明的是，本实施例在通过上层软件确定产生区块回滚和重做需求时，并没有真正执行区块回滚和重做操作，而是基于节点所维护的键标识和版本标识的映射关系表，直接存储了其他节点提供的区块数据和版本标识。

S580, 通过上层软件根据区块回滚和重做需求所确定的生效区块版本标识，向本地 KV 存储系统发起区块数据的数据处理请求。

具体的，在执行步骤 S570 之后，可以通过上层软件根据区块回滚和重做需求所确定的生效区块标识和版本标识，向本地 KV 存储系统发起区块数据的查询、删除、更改或写入等数据处理请求，也就是说本地 KV 存储系统将执行步骤 S510 至步骤 S550 的操作。

需要说明的是，步骤 S560 至步骤 S580 的操作过程，无需在步骤 S550 之后执行，只要本地 KV 存储系统已向其上层软件提供键标识和版本标识的映射关系表即可；也就是说也可以在步骤 S510 至 S550 任一步骤之后或之前执行等。

本申请实施例提供的技术方案，在通过上层软件确定产生区块回滚和重做需求时，可从其他节点接收区块数据，及区块数据的键标识和版本标识的映射关系表；而后通过上层软件向本地 KV 存储系统发起数据写入请求，以请求本地 KV 存储系统将区块数据写入本地的存储空间中，同时可以将获取的映射关系表添加至本地的映射关系表中，进而可完成分叉区块的调整，相比于现有的回滚和重做方式，本方案并没有真正执行区块回滚和重做操作，仅基于节点所维护的键标识和版本标识的映射关系表，即可解决区块链节点的区块回滚和重做。优化了区块链 KV 存储系统的索引方式，减少了索引所占用的数据量，且为区块回滚和重做提供了一种新思路。

以前述示例来说，区块链节点直接从其他节点接受了 E 区块和 F 区块在执行过程中所涉及的数据，既包括区块中的事务数据，也包括在执行事务请求后所修改的数据，在区块执行过程中所涉及的数据都具有相同的版本标识。因此，区块链节点只需要通过上层软件来确定，E 区块和 F 区块的版本标识有效，D 区块的版本标识无效，就能在后续过程中，只处理有效区块对应数据，不会处理无效区块对应数据了。但无效区块对应数据并未从存储空间删除。上述操作过程，不需要将无效区块的事务请求回滚来消除对已存储数据的影响，也无需将有效区块的事务请求重做来得到新数据。

此外，需要说明的是，若同一区块内涉及多个事务请求，需对同一目标键值对进行操作，而目标键值对的版本标识可以只在执行第一个事务请求时进行了修改，后续在执行其他事务请求时，只对目标键值对的值域数据进行修改，而不修改版本标识。例如，在一个区块的处理过程中，一个事务请求为账户 A

向账户 B 转账，另一事务请求为账户 A 向账户 C 转账；账户 A 所对应的目标键值对的版本标识，在执行账户 A 向账户 B 转账后，确定为当前版本标识；而在执行账户 A 向账户 C 转账后，账户 A 所对应的目标键值对的版本标识仍为当前版本标识，并不产生新的键值对。但是由于对同一目标键值对的操作均在同一区块内，因此并不影响基于版本标识来解决区块回滚问题。

实施例六

图 6 为本申请实施例六提供的一种数据存储方法的流程图，本实施例在上述实施例的基础上，进一步的优化。参见图 6，该方法具体包括：

S610，获取数据处理请求。

S620，根据数据处理请求确定当前版本标识。

S630，根据数据处理请求，确定用于处理数据的目标键值对，且对目标键值对值域中的数据进行处理。

S640，将新产生的目标键值对写入存储空间，其中，存储空间中的键值对的键域存储有键标识和版本标识，且新产生的目标键值对的键域中的版本标识为当前版本标识。

S650，根据新产生的目标键值对，在键标识和版本标识的映射关系表中，添加键标识和当前版本标识的映射关系。

S660，通过上层软件在键标识和版本标识的映射关系表中，确定待处理数据的键标识和生效版本标识，并根据键标识和生效版本标识产生数据更改请求、数据删除请求或数据查询请求。

具体的，本地 KV 存储系统向上层软件提供键标识和版本标识的映射关系表，区块链节点需要查询、删除或更改某一数据时，可以基于上层软件在键标识和版本标识的映射关系表中，确定待处理数据的键标识和生效版本标识；而后根据键标识和生效版本标识产生数据更改请求、数据删除请求或数据查询请求，并将该请求通过上层软件发送至本地 KV 存储系统，以使本地 KV 存储系统执行步骤 S610 至步骤 S650 的操作，并通过本地 KV 存储系统向区块链节点反馈相应的结果，如反馈读取的数据等。

需要说明的是，步骤 S660 的操作过程，无需在步骤 S650 之后执行，只要本地 KV 存储系统已向其上层软件提供键标识和版本标识的映射关系表即可；也就是说也可以在步骤 S610 至 S650 任一步骤之后或之前执行等。

在本地 KV 存储系统已向其上层软件提供键标识和版本标识的映射关系表的情况，示例性的，还可以包括：通过上层软件在新业务数据写入时，产生数据写入请求，并根据前一业务数据的版本标识作为数据写入请求依赖的前一版本标识，携带在数据写入请求中；其中，新业务数据为新区块或新事务数据。

具体的，区块链节点在需要本地 KV 存储系统存储数据时，可以基于上层软件在键标识和版本标识的映射关系表中，确定前一业务数据的版本标识，并将该版本标识作为数据写入请求依赖的前一版本标识；而后根据前一版本标识、新业务数据及新业务数据标识产生数据写入请求，并将该请求通过上层软件发送至本地 KV 存储系统，以使本地 KV 存储系统执行步骤 S610 至步骤 S640 的操作，以完成新业务数据的写入，并可执行步骤 S650，以更新键标识和版本标识的映射关系表，同时向上层软件反馈更新的键标识和版本标识的映射关系表。

本申请实施例提供的技术方案，区块链节点在需要查询、删除或更改某一数据时，可以基于上层软件在键标识和版本标识的映射关系表中，确定待处理数据的键标识和生效版本标识；而后根据键标识和生效版本标识产生数据更改请求、数据删除请求或数据查询请求，并将该请求通过上层软件发送至本地 KV 存储系统，以使本地 KV 存储系统执行将数据或键值对写入存储空间的操作。

实施例七

图 7 为本申请实施例七提供的一种数据存储装置的结构示意图，该装置可配置于计算设备中，特别是作为区块链节点的计算设备中，可执行本申请任意实施例所提供的数据存储方法，具备执行方法相应的功能模块和有益效果。如图 7 所示，该装置包括：

处理请求获取模块 710，设置为获取数据处理请求；

当前标识确定模块 720，设置为根据数据处理请求确定当前版本标识；

数据处理模块 730，设置为根据数据处理请求，确定用于处理数据的目标键值对，且对目标键值对值域中的数据进行处理；

写入模块 740，设置为将新产生的目标键值对写入存储空间，其中，存储空间中的键值对的键域存储有键标识和版本标识，且新产生的目标键值对的键域中的版本标识为当前版本标识。

本申请实施例提供的技术方案，通过数据处理请求确定当前版本标识，以及用于处理数据的目标键值对，并对目标键值对值域中的数据进行如写入、更改、删除或读取等处理，且在目标键值对为新产生的目标键值对的情况下，将当前版本标识存储在目标键值对的键域；而后将新产生的目标键值对写入存储空间，进而存储数据的存储。本方案在需要对数据进行写入、更改或删除等处理时，保留已有键值对，在新产生的键值对中进行操作，便于后续基于版本标识快速切换至之前其他版本标识的键值对数据进行查看，优化了区块链 KV 存储系统的索引方式。特别是，由于区块链系统存在区块回滚和重做的需求，例如发生分叉时需要回滚，那么基于本申请实施例所提供的 KV 存储系统能够方便的切换至之前任何版本的键值对数据，不需要进行大量键值对的数据恢复。

相比于现有区块链的数据存储方式而言，提高了版本恢复速度，为区块链的数据存储提供了一种新思路。

示例性的，当前标识确定模块 720 是设置为：

根据数据处理请求确定前一版本数据对象；

基于前一版本数据对象，产生当前版本数据对象，并确定当前版本标识，其中，当前版本数据对象与前一版本数据对象所操作的存储空间相同。

示例性的，数据处理模块 730 在根据数据处理请求，确定用于处理数据的目标键值对时，还设置为：

根据数据处理请求中目标数据的键标识和版本标识，确定目标数据所在的已有键值对，作为目标键值对。

示例性的，数据处理模块 730 还设置为：

如果数据处理请求为目标数据的写入操作，则产生新键值对作为目标键值对，在目标键值对的键域写入新的键标识和当前版本标识，在目标键值对的值域写入目标数据；

如果数据处理请求为将已有键值对值域数据更改为目标数据，则产生新键值对作为目标键值对，在目标键值对的键域写入已有键值对的键标识和当前版本标识，在目标键值对的值域写入目标数据；

如果数据处理请求为已有键值对的删除操作，则产生新键值对作为目标键值对，在目标键值对的键域写入已有键值对的键标识和当前版本标识，将目标键值对的值域设置为空；

如果数据处理请求为已有键值对的读取操作，则确定已有键值对作为目标键值对，将目标键值对值域中的数据进行读取反馈。

示例性的，数据处理模块 730 还设置为：

在当前版本数据对象中，根据数据处理请求调用相应数据操作的功能函数，将当前版本数据对象的当前版本标识作为功能函数的输入参数，以执行数据处理请求。

示例性的，当前标识确定模块 720 可以包括：

前一版本标识确定单元，设置为确定前一版本标识；

当前标识确定单元，设置为按照设定编号规则产生当前版本标识；或

当前标识确定单元，设置为在前一版本标识确定单元确定前一版本标识后，根据前一版本标识的流水号标识，按照设定编号规则产生当前版本标识中的流水号标识，以及根据数据处理请求确定的版本附加标识，与流水号标识组成所述当前版本标识。

示例性的，前一版本标识确定单元是设置为：

从数据处理请求中提取前一版本标识；或

根据数据处理请求中的业务数据标识确定前一版本标识，其中，业务数据标识为区块标识或事务标识。

示例性的，版本附加标识为区块标识或事务标识。

示例性的，写入模块 740 可以包括：

生效指令调用单元，设置为调用生效控制指令，以控制当前版本数据对象生效；

写入单元，设置为将当前版本数据对象中新产生的目标键值对写入存储空间。

示例性的，上述装置还可以包括：

失效指令调用模块，设置为将新产生的目标键值对写入存储空间之前，调用失效控制指令，以控制当前版本数据对象失效。

示例性的，还可以包括：修改更新单元，该单元设置为：

在控制当前版本数据对象生效之前，根据上层软件传输的版本标识修改指示，修改当前版本标识，并更新键值对键域中的当前版本标识。

示例性的，生效指令调用单元是设置为：

在产生当前版本生效条件，或接收到上层软件发送的当前版本生效指令时，调用生效控制指令。

示例性的，产生当前版本生效条件可以包括下述至少一项：

待处理数据的区块标识发生变更；

待处理数据的事务标识发生变更。

示例性的，失效指令调用模块还设置为：

在产生当前版本失效条件，或接收到上层软件发送的当前版本失效指令时，调用失效控制指令。

示例性的，上述装置还可以包括：

添加模块，设置为对目标键值对值域中的数据进行处理之后，根据新产生的目标键值对，在键标识和版本标识的映射关系表中，添加键标识和当前版本标识的映射关系。

示例性的，上述装置还可以包括：回滚和重做模块，该模块设置为：

通过上层软件确定产生区块回滚和重做需求时，获取从其他节点接收的区块数据，以及区块数据的键标识和版本标识的映射关系表；

通过上层软件向本地 KV 存储系统发起数据写入请求，将获取到的区块数据写入本地的存储空间，并将获取的映射关系表添加至本地的映射关系表中；

通过上层软件根据区块回滚和重做需求所确定的生效区块版本标识，向本地 KV 存储系统发起区块数据的数据处理请求。

示例性的，回滚和重做模块在将获取的映射关系表添加至本地的映射关系

表中时，还设置为：

将获取的映射表相对于本地映射关系表的增量键标识和版本标识，添加至本地映射关系表中，且保留本地映射关系表中的已有键标识和版本标识，其中，增量版本标识中包括生效区块版本标识，本地映射关系表的已有版本标识中包括失效区块版本标识。

示例性的，上述装置还可以包括：

请求产生模块，设置为通过上层软件在键标识和版本标识的映射关系表中，确定待处理数据的键标识和生效版本标识，并根据键标识和生效版本标识产生数据更改请求、数据删除请求或数据查询请求。

示例性的，请求产生模块还设置为：

通过上层软件在新业务数据写入时，产生数据写入请求，并根据前一业务数据的版本标识作为数据写入请求依赖的前一版本标识，携带在数据写入请求中；其中，新业务数据为新区块或新事务数据。

示例性的，存储空间为磁盘存储空间，磁盘存储空间中的 KV 存储系统所使用的索引为有序表或索引树。

示例性的，上述装置还可以包括：

增加模块，设置为在将新产生的目标键值对写入存储空间的过程中，在版本标识和键标识的反向索引中，增加对应于当前版本标识的新产生目标键值对的键标识。

实施例八

图 8 为本申请实施例八提供的一种设备的结构示意图。图 8 示出了适于用来实现本申请实施方式的示例性设备 12 的框图。图 8 显示的设备 12 仅仅是一个示例，不应对本申请实施例的功能和使用范围带来任何限制。设备 12 典型可以是计算设备，也可以是承担区块链网络节点功能的计算设备。

如图 8 所示，设备 12 以通用计算设备的形式表现。设 12 的组件可以包括但不限于：一个或者多个处理器或者处理单元 16，系统存储器 28，连接不同系统组件（包括系统存储器 28 和处理单元 16）的总线 18。

总线 18 表示几类总线结构中的一种或多种，包括存储器总线或者存储器控制器，外围总线，图形加速端口，处理器或者使用多种总线结构中的任意总线结构的局域总线。举例来说，这些体系结构包括但不限于工业标准体系结构 (ISA) 总线，微通道体系结构 (MAC) 总线，增强型 ISA 总线、视频电子标准协会 (VESA) 局域总线以及外围组件互连 (PCI) 总线。

设备 12 典型地包括多种计算机系统可读介质。这些介质可以是任何能够被设备 12 访问的可用介质，包括易失性和非易失性介质，可移动的和不可移动的

介质。

系统存储器 28 可以包括易失性存储器形式的计算机系统可读介质，例如随机存取存储器 (RAM) 30 和/或高速缓存存储器 32。设备 12 可以进一步包括其它可移动/不可移动的、易失性/非易失性计算机系统存储介质。仅作为举例，存储系统 34 可以用于读写不可移动的、非易失性磁介质 (图 8 未显示，通常称为“硬盘驱动器”)。尽管图 8 中未示出，可以提供用于对可移动非易失性磁盘 (例如“软盘”) 读写的磁盘驱动器，以及对可移动非易失性光盘 (例如 CD-ROM, DVD-ROM 或者其它光介质) 读写的光盘驱动器。在这些情况下，每个驱动器可以通过一个或者多个数据介质接口与总线 18 相连。系统存储器 28 可以包括至少一个程序产品，该程序产品具有一组 (例如至少一个) 程序模块，这些程序模块被配置以执行本申请各实施例的功能。

具有一组 (至少一个) 程序模块 42 的程序/实用工具 40，可以存储在例如系统存储器 28 中，这样的程序模块 42 包括但不限于操作系统、一个或者多个应用程序、其它程序模块以及程序数据，这些示例中的每一个或某种组合中可能包括网络环境的实现。程序模块 42 通常执行本申请所描述的实施例中的功能和/或方法。

设备 12 也可以与一个或多个外部设备 14 (例如键盘、指向设备、显示器 24 等) 通信，还可与一个或者多个使得用户能与该设备 12 交互的设备通信，和/或与使得该设备 12 能与一个或多个其它计算设备进行通信的任何设备 (例如网卡，调制解调器等等) 通信。这种通信可以通过输入/输出 (I/O) 接口 22 进行。并且，设备 12 还可以通过网络适配器 20 与一个或者多个网络 (例如局域网 (LAN)，广域网 (WAN) 和/或公共网络，例如因特网) 通信。如图所示，网络适配器 20 通过总线 18 与设备 12 的其它模块通信。应当明白，尽管图中未示出，可以结合设备 12 使用其它硬件和/或软件模块，包括但不限于：微代码、设备驱动器、冗余处理单元、外部磁盘驱动阵列、RAID 系统、磁带驱动器以及数据备份存储系统等。

处理单元 16 通过运行存储在系统存储器 28 中的程序，从而执行各种功能应用以及数据处理，例如实现本申请实施例所提供的数据存储方法。

实施例九

本申请实施例九还提供了一种计算机可读存储介质，其上存储有计算机程序，该程序被处理器执行时可实现上述实施例所述的数据存储方法。该计算机可读存储介质，可以配置于计算设备上，也可配置于区块链节点上。

本申请实施例的计算机存储介质，可以采用一个或多个计算机可读的介质的任意组合。计算机可读介质可以是计算机可读信号介质或者计算机可读存储

介质。计算机可读存储介质例如可以是——但不限于——电、磁、光、电磁、红外线、或半导体的系统、装置或器件，或者任意以上的组合。计算机可读存储介质的更具体的例子（非穷举的列表）包括：具有一个或多个导线的电连接、便携式计算机磁盘、硬盘、随机存取存储器（RAM）、只读存储器（ROM）、可擦式可编程只读存储器（EPROM 或闪存）、光纤、便携式紧凑磁盘只读存储器（CD-ROM）、光存储器件、磁存储器件、或者上述的任意合适的组合。在本文件中，计算机可读存储介质可以是任何包含或存储程序的有形介质，该程序可以被指令执行系统、装置或者器件使用或者与其结合使用。

计算机可读的信号介质可以包括在基带中或者作为载波一部分传播的数据信号，其中承载了计算机可读的程序代码。这种传播的数据信号可以采用多种形式，包括但不限于电磁信号、光信号或上述的任意合适的组合。计算机可读的信号介质还可以是计算机可读存储介质以外的任何计算机可读介质，该计算机可读介质可以发送、传播或者传输用于由指令执行系统、装置或者器件使用或者与其结合使用的程序。

计算机可读介质上包含的程序代码可以用任何适当的介质传输，包括——但不限于无线、电线、光缆、RF 等等，或者上述的任意合适的组合。

可以以一种或多种程序设计语言或其组合来编写用于执行本申请操作的计算机程序代码，所述程序设计语言包括面向对象的程序设计语言——诸如 Java、Smalltalk、C++，还包括常规的过程式程序设计语言——诸如“C”语言或类似的设计语言。程序代码可以完全地在用户计算机上执行、部分地在用户计算机上执行、作为一个独立的软件包执行、部分在用户计算机上部分在远程计算机上执行、或者完全在远程计算机或服务器上执行。在涉及远程计算机的情形中，远程计算机可以通过任意种类的网络——包括局域网（LAN）或广域网（WAN）——连接到用户计算机，或者，可以连接到外部计算机（例如利用因特网服务提供商来通过因特网连接）。

注意，虽然通过以上实施例对本申请进行了较为详细的说明，但是本申请不仅仅限于以上实施例，在不脱离本申请构思的情况下，还可以包括更多其他等效实施例，而本申请的范围由所附的权利要求范围决定。

1、一种数据存储方法，所述方法包括：

获取数据处理请求；

根据所述数据处理请求确定当前版本标识；

根据所述数据处理请求，确定用于处理数据的目标键值对，且对所述目标键值对值域中的数据进行处理；

将新产生的目标键值对写入存储空间，其中，存储空间中的键值对的键域存储有键标识和版本标识，且新产生的目标键值对的键域中的版本标识为当前版本标识。

2、根据权利要求1所述的方法，其中，根据所述数据处理请求确定当前版本标识，包括：

根据所述数据处理请求确定前一版本数据对象；

基于前一版本数据对象，产生当前版本数据对象，并确定当前版本标识，其中，所述当前版本数据对象与前一版本数据对象所操作的存储空间相同。

3、根据权利要求1或2所述的方法，其中，根据所述数据处理请求，确定用于处理数据的目标键值对，包括：

根据所述数据处理请求中目标数据的键标识和版本标识，确定目标数据所在的已有键值对，作为目标键值对。

4、根据权利要求1所述的方法，其中，根据所述数据处理请求，确定用于处理数据的目标键值对，且对所述目标键值对值域中的数据进行处理，包括：

如果所述数据处理请求为目标数据的写入操作，则产生新键值对作为目标键值对，在所述目标键值对的键域写入新的键标识和当前版本标识，在所述目标键值对的值域写入所述目标数据；

如果所述数据处理请求为将已有键值对值域数据更改为目标数据，则产生新键值对作为目标键值对，在所述目标键值对的键域写入已有键值对的键标识和当前版本标识，在所述目标键值对的值域写入所述目标数据；

如果所述数据处理请求为已有键值对的删除操作，则产生新键值对作为目标键值对，在所述目标键值对的键域写入已有键值对的键标识和当前版本标识，将所述目标键值对的值域设置为空；

如果所述数据处理请求为已有键值对的读取操作，则确定已有键值对作为目标键值对，将所述目标键值对值域中的数据进行读取反馈。

5、根据权利要求3所述的方法，其中，根据所述数据处理请求，确定用于处理数据的目标键值对，且对所述目标键值对值域中的数据进行处理，包括：

在当前版本数据对象中，根据所述数据处理请求调用相应数据操作的功能函数，将当前版本数据对象的当前版本标识作为所述功能函数的输入参数，以执行所述数据处理请求。

6、根据权利要求1或2所述的方法，其中，根据所述数据处理请求确定当前版本标识，包括：

确定前一版本标识，按照设定编号规则产生当前版本标识；或

确定前一版本标识，根据前一版本标识的流水号标识，按照设定编号规则产生当前版本标识中的流水号标识，以及根据所述数据处理请求确定的版本附加标识，与所述流水号标识组成所述当前版本标识。

7、根据权利要求6所述的方法，其中，确定前一版本标识，包括：

从所述数据处理请求中提取前一版本标识；或

根据所述数据处理请求中的业务数据标识确定前一版本标识，其中，所述业务数据标识为区块标识或事务标识。

8、根据权利要求6所述的方法，所述版本附加标识为区块标识或事务标识。

9、根据权利要求2所述的方法，其中，将新产生的目标键值对写入存储空间，包括：

调用生效控制指令，以控制所述当前版本数据对象生效，将所述当前版本数据对象中新产生的目标键值对写入存储空间。

10、根据权利要求2所述的方法，其中，将新产生的目标键值对写入存储空间之前，还包括：

调用失效控制指令，以控制所述当前版本数据对象失效。

11、根据权利要求9所述的方法，其中，控制所述当前版本数据对象生效之前，还包括：

根据上层软件传输的版本标识修改指示，修改所述当前版本标识，并更新键值对键域中的当前版本标识。

12、根据权利要求9所述的方法，其中，调用生效控制指令，包括：

在产生当前版本生效条件，或接收到上层软件发送的当前版本生效指令时，调用所述生效控制指令。

13、根据权利要求12所述的方法，其中，产生当前版本生效条件，包括下述至少一项：

待处理数据的区块标识发生变更；

待处理数据的事务标识发生变更。

14、根据权利要求10所述的方法，其中，调用失效控制指令，包括：

在产生当前版本失效条件，或接收到上层软件发送的当前版本失效指令时，调用失效控制指令。

15、根据权利要求1所述的方法，其中，对所述目标键值对值域中的数据进行处理之后，还包括：

根据新产生的目标键值对，在键标识和版本标识的映射关系表中，添加键

标识和当前版本标识的映射关系。

16、根据权利要求 15 所述的方法，所述方法应用于区块链节点，则所述方法还包括：

通过上层软件确定产生区块回滚和重做需求时，获取从其他节点接收的区块数据，以及所述区块数据的键标识和版本标识的映射关系表；

通过上层软件向本地 KV 存储系统发起数据写入请求，将获取到的所述区块数据写入本地的存储空间，并将获取的所述映射关系表添加至本地的映射关系表中；

通过上层软件根据区块回滚和重做需求所确定的生效区块版本标识，向本地 KV 存储系统发起区块数据的数据处理请求。

17、根据权利要求 16 所述的方法，其中，将获取的所述映射关系表添加至本地的映射关系表中，包括：

将获取的所述映射表相对于本地映射关系表的增量键标识和版本标识，添加至本地映射关系表中，且保留本地映射关系表中的已有键标识和版本标识，其中，增量版本标识中包括生效区块版本标识，本地映射关系表的已有版本标识中包括失效区块版本标识。

18、根据权利要求 15 所述的方法，所述方法还包括：

通过上层软件在键标识和版本标识的映射关系表中，确定待处理数据的键标识和生效版本标识，并根据所述键标识和生效版本标识产生数据更改请求、数据删除请求或数据查询请求。

19、根据权利要求 15 所述的方法，所述方法还包括：

通过上层软件在新业务数据写入时，产生数据写入请求，并根据前一业务数据的版本标识作为所述数据写入请求依赖的前一版本标识，携带在所述数据写入请求中；

其中，所述新业务数据为新区块或新事务数据。

20、根据权利要求 1 所述的方法，所述存储空间为磁盘存储空间，所述磁盘存储空间中的 KV 存储系统所使用的索引为有序表或索引树。

21、根据权利要求 1 所述的方法，其中，将新产生的目标键值对写入存储空间的过程中，还包括：

在版本标识和键标识的反向索引中，增加对应于当前版本标识的新产生目标键值对的键标识。

22、一种数据存储装置，所述装置包括：

处理请求获取模块，设置为获取数据处理请求；

当前标识确定模块，设置为根据所述数据处理请求确定当前版本标识；

数据处理模块，设置为根据所述数据处理请求，确定用于处理数据的目标

键值对，且对所述目标键值对值域中的数据进行处理；

写入模块，设置为将新产生的目标键值对写入存储空间，其中，存储空间中的键值对的键域存储有键标识和版本标识，且新产生的目标键值对的键域中的版本标识为当前版本标识。

23、一种设备，所述设备包括：

一个或多个处理器；

存储装置，设置为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，使得所述一个或多个处理器实现如权利要求 1-21 中任一所述的数据存储方法。

24、一种存储介质，其上存储有计算机程序，该程序被处理器执行时实现如权利要求 1-21 中任一所述的数据存储方法。

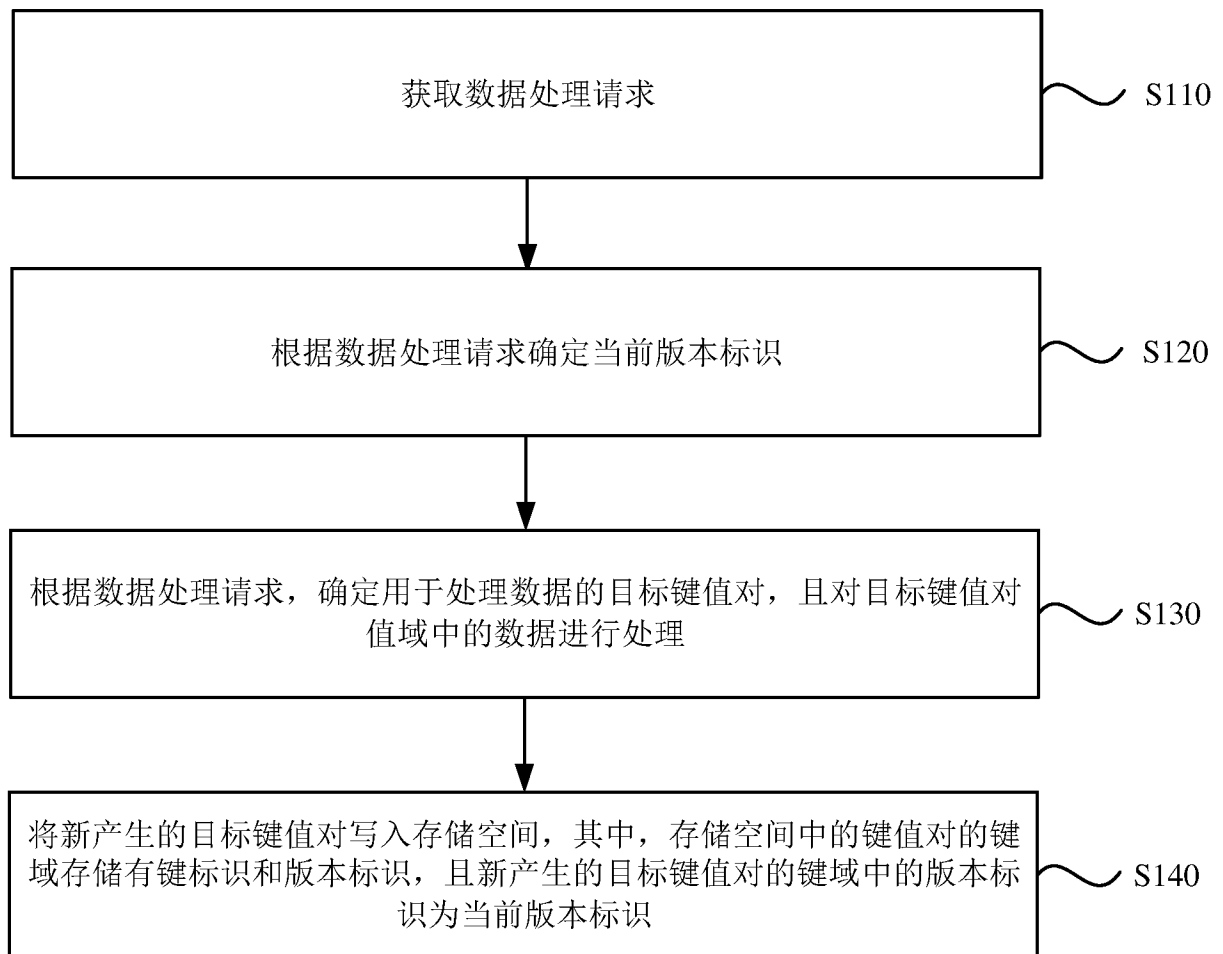


图 1

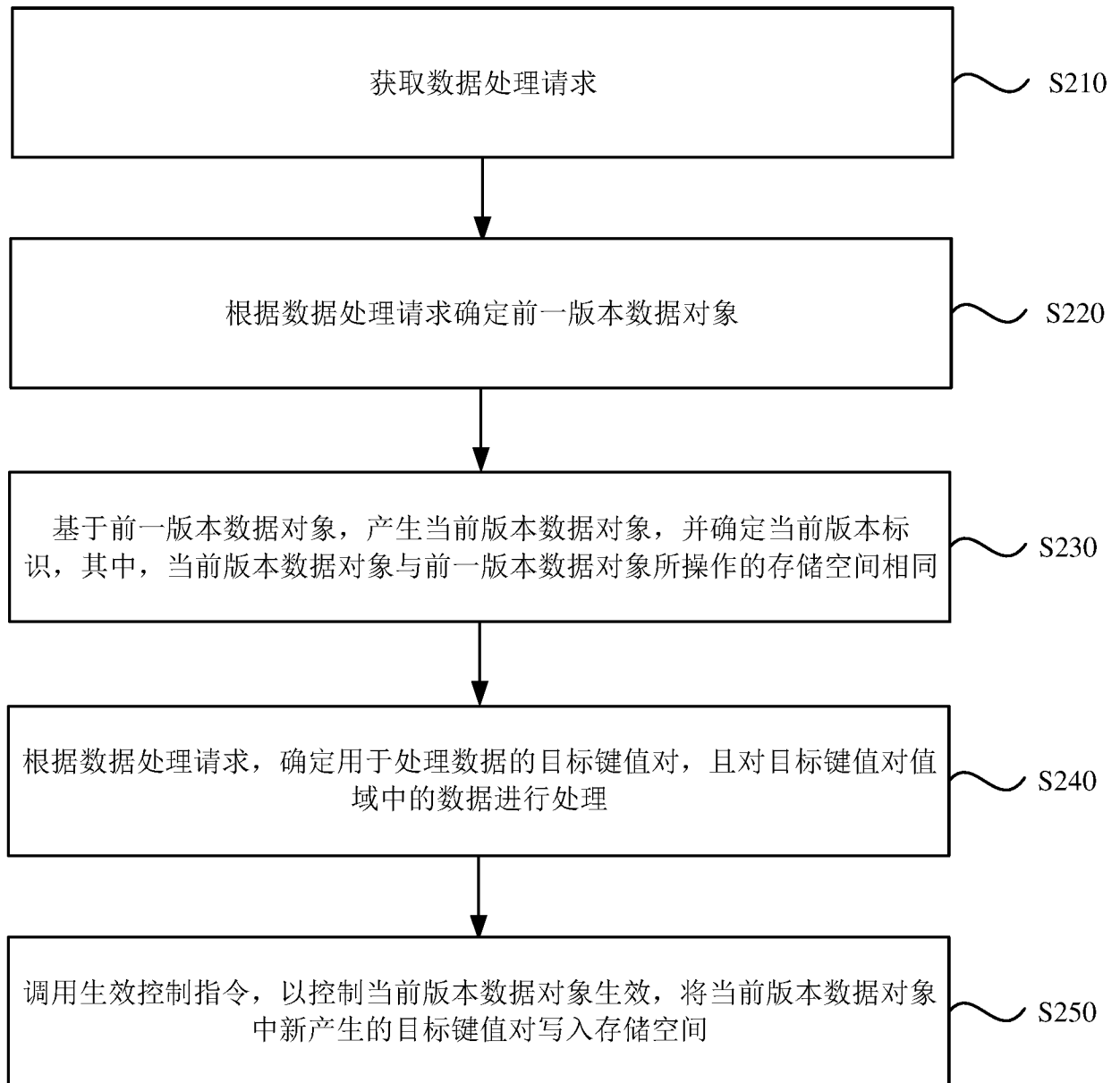


图 2

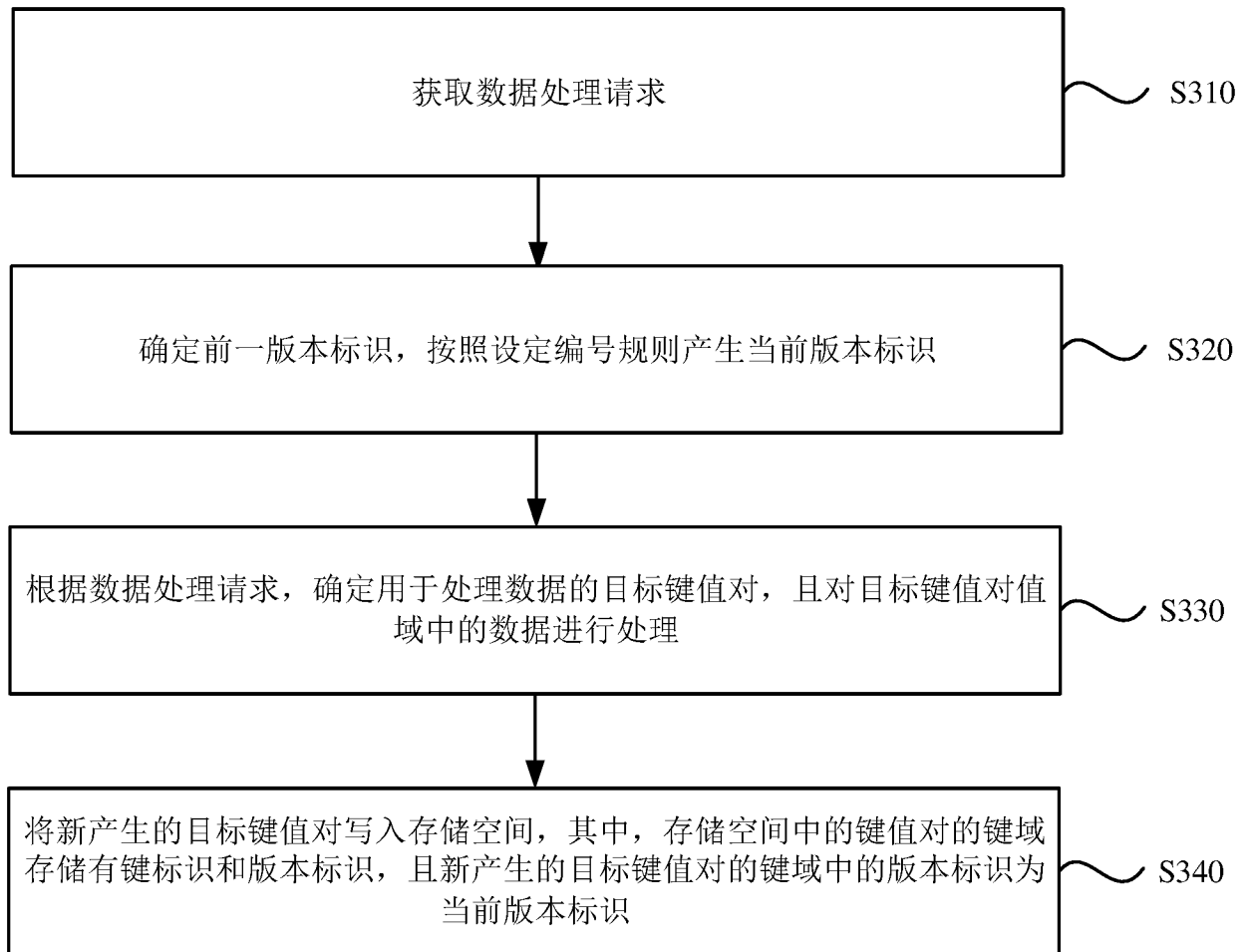


图 3

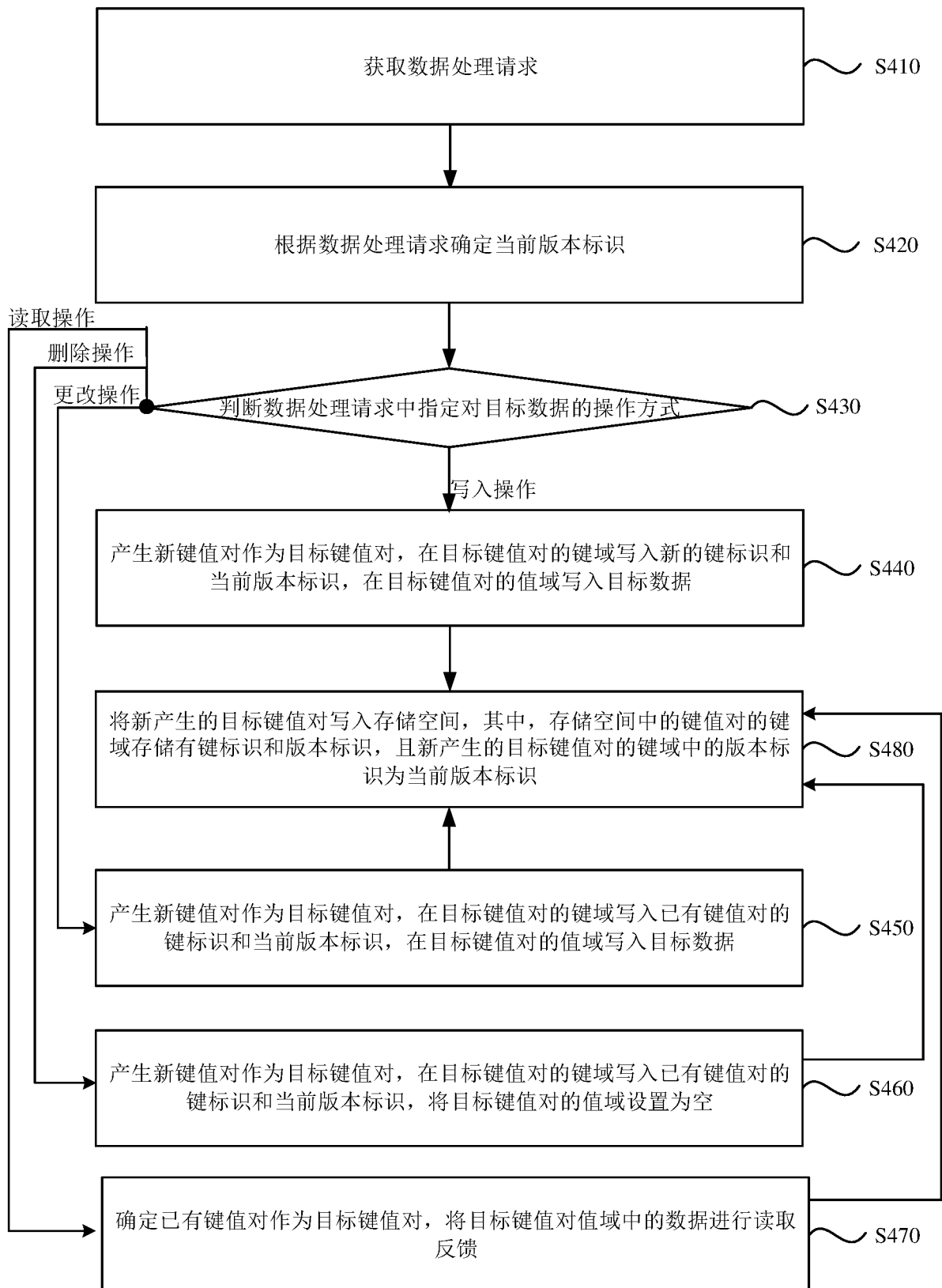


图 4

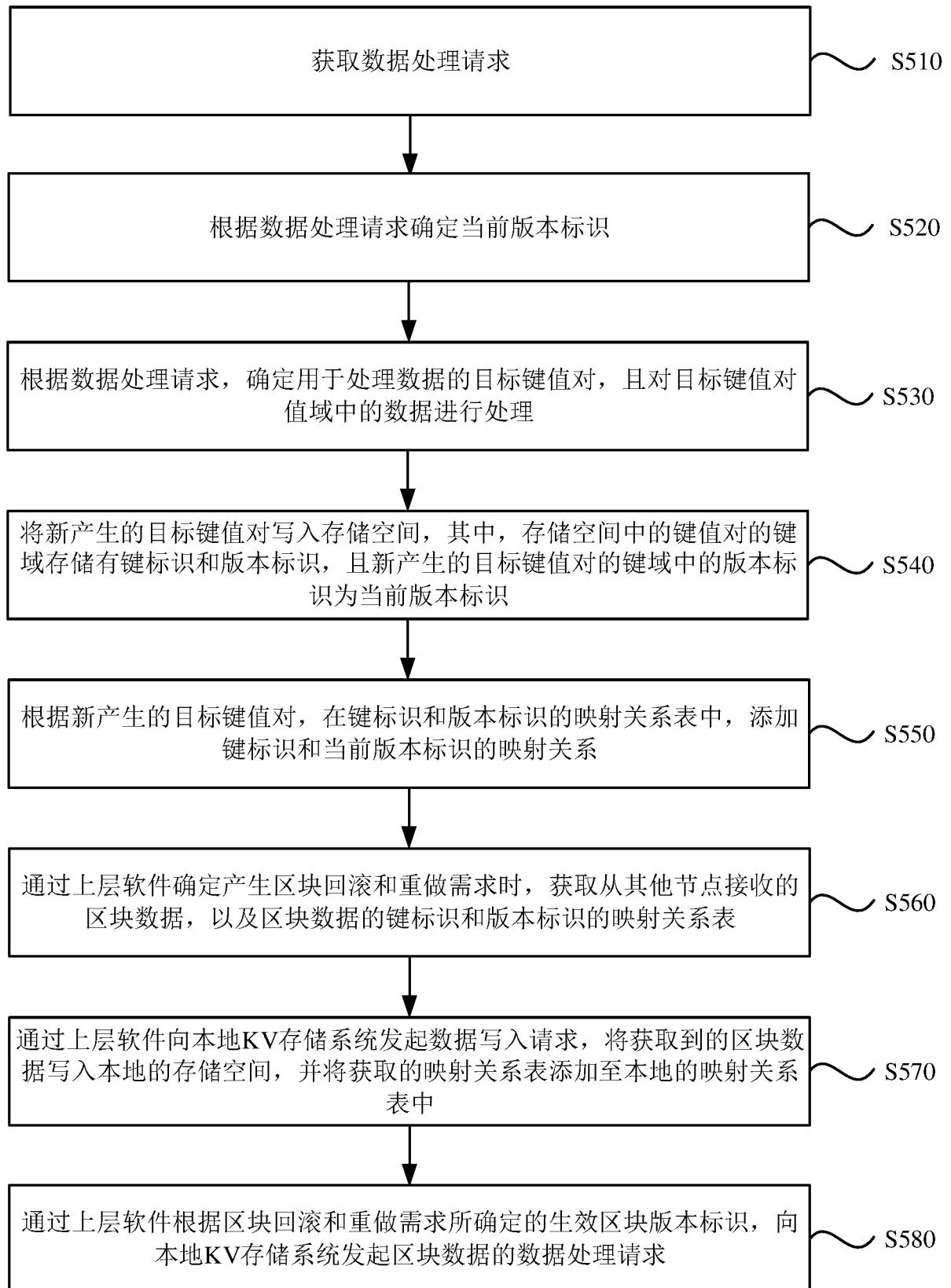


图 5A

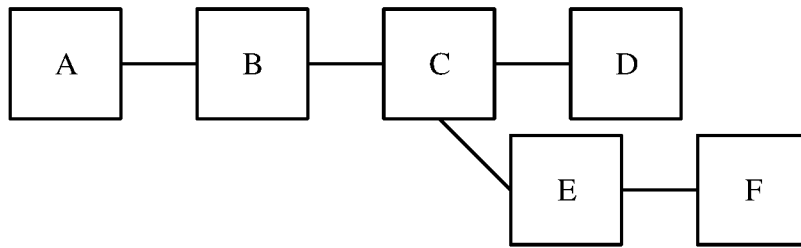


图 5B

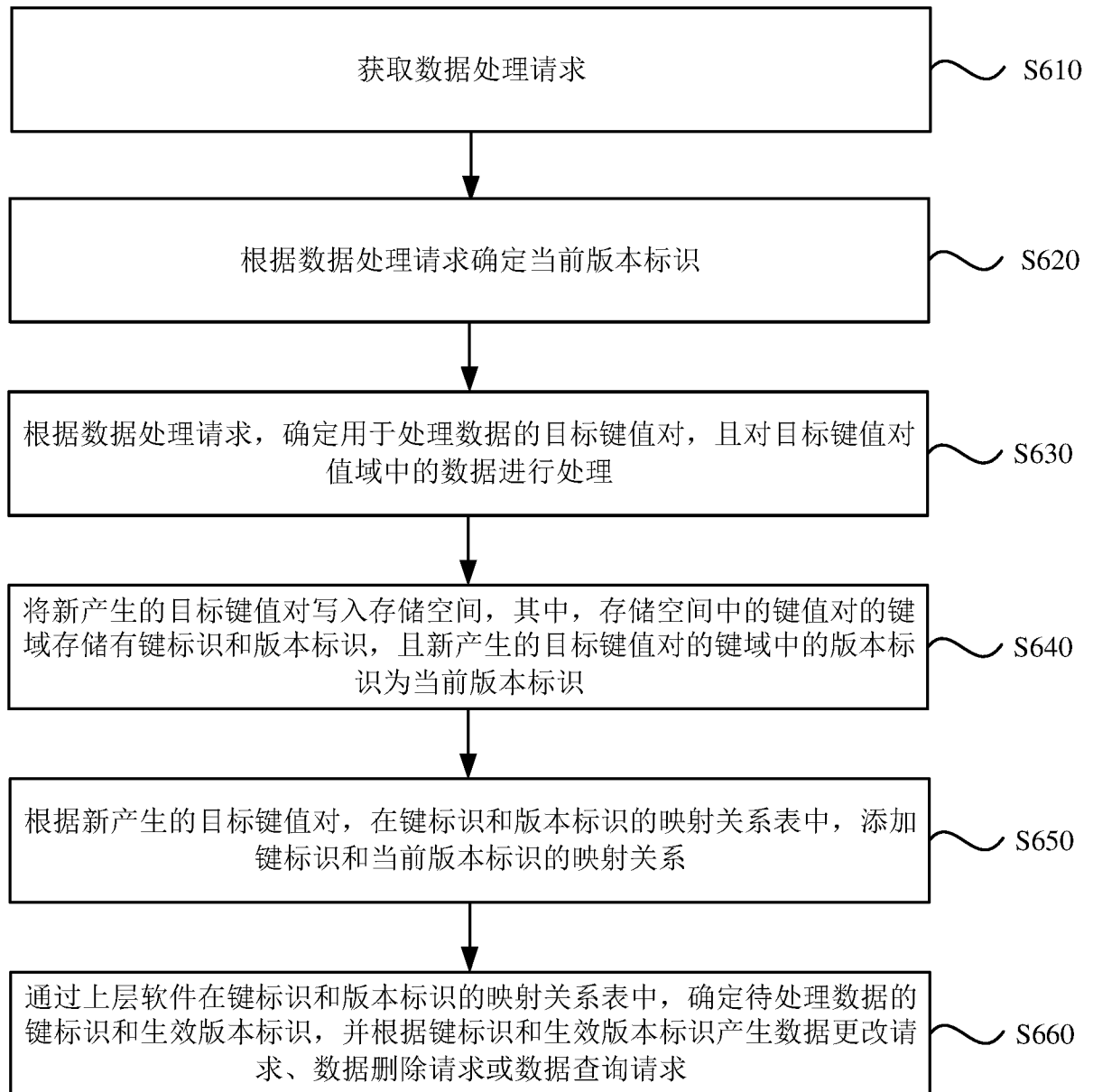


图 6

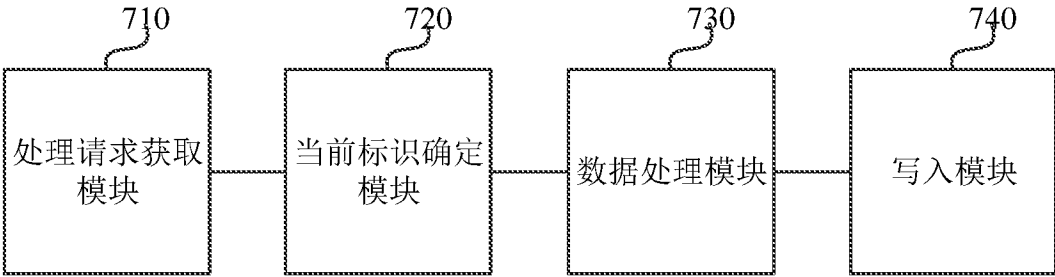


图 7

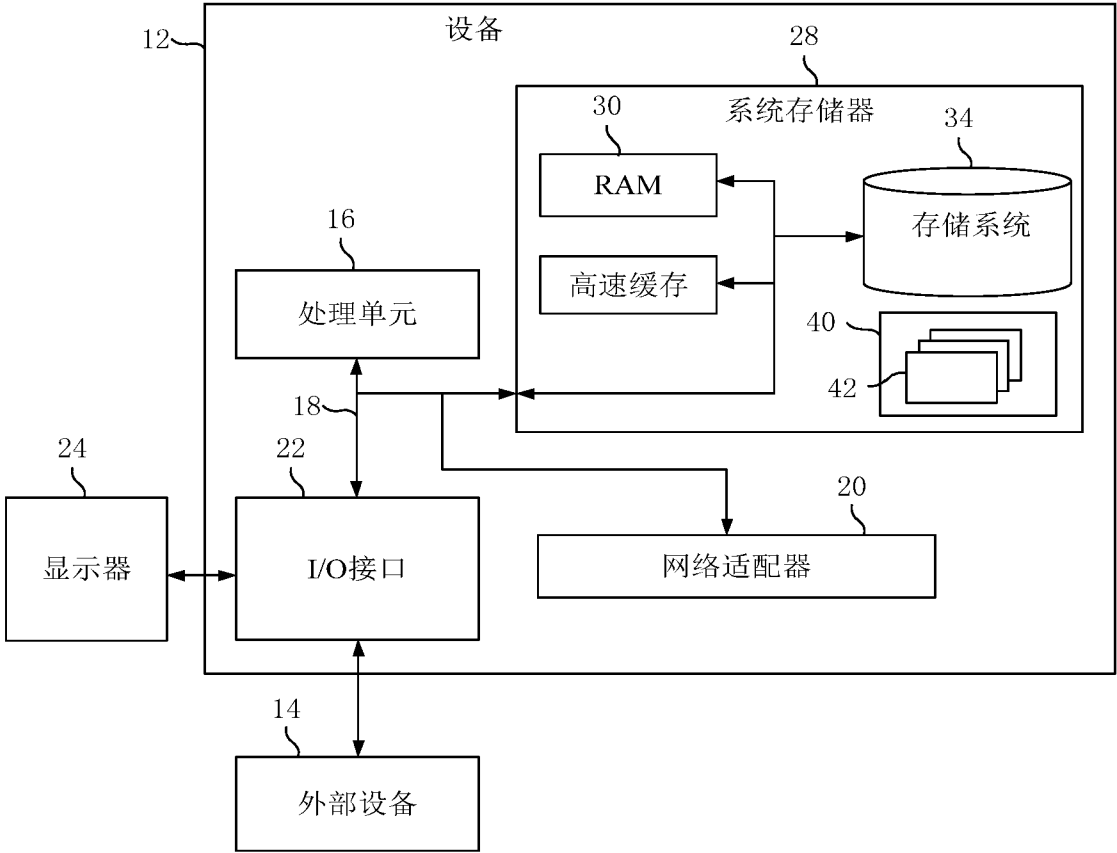


图 8

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2019/100495

A. CLASSIFICATION OF SUBJECT MATTER

G06F 16/22(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; 3GPP; IEEE: 标识, 键值对, 存储, 版本, 域, storage, version, identifier, key, value, pair, field, domain

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
PX	CN 109684307 A (BAIDU ONLINE NETWORK TECHNOLOGY (BEIJING) CO., LTD.) 26 April 2019 (2019-04-26) claims 1-24	1-24
PX	CN 109710190 A (BAIDU ONLINE NETWORK TECHNOLOGY (BEIJING) CO., LTD.) 03 May 2019 (2019-05-03) claims 1-25	1-24
X	CN 103902698 A (BEIJING CHESHANGHUI SOFTWARE CO., LTD.) 02 July 2014 (2014-07-02) description, paragraphs [0068]-[0091]	1-24
A	CN 108319661 A (LIANJIANG WANG (BEIJING) TECHNOLOGY CO., LTD.) 24 July 2018 (2018-07-24) entire document	1-24
A	US 2014172898 A1 (MICROSOFT CORPORATION) 19 June 2014 (2014-06-19) 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

15 October 2019

Date of mailing of the international search report

30 October 2019

Name and mailing address of the ISA/CN

China National Intellectual Property Administration
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/CN2019/100495

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	109684307	A	26 April 2019	None			
CN	109710190	A	03 May 2019	None			
CN	103902698	A	02 July 2014	None			
CN	108319661	A	24 July 2018	None			
US	2014172898	A1	19 June 2014	US	2016117363	A1	28 April 2016
				EP	2932410	A2	21 October 2015
				WO	2014093920	A2	19 June 2014
				BR	112015011884	A2	11 July 2017
				CN	105009111	A	28 October 2015

国际检索报告

国际申请号

PCT/CN2019/100495

A. 主题的分类 G06F 16/22 (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; 3GPP; IEEE: 标识, 键值对, 存储, 版本, 域, storage, version, identifier, key, value, pair, field, domain		
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 109684307 A (百度在线网络技术北京有限公司) 2019年 4月 26日 (2019 - 04 - 26) 权利要求1-24	1-24
PX	CN 109710190 A (百度在线网络技术北京有限公司) 2019年 5月 3日 (2019 - 05 - 03) 权利要求1-25	1-24
X	CN 103902698 A (北京车商汇软件有限公司) 2014年 7月 2日 (2014 - 07 - 02) 说明书第[0068]-[0091]段	1-24
A	CN 108319661 A (链家网北京科技有限公司) 2018年 7月 24日 (2018 - 07 - 24) 全文	1-24
A	US 2014172898 A1 (MICROSOFT CORPORATION) 2014年 6月 19日 (2014 - 06 - 19) 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		国际检索报告邮寄日期
2019年 10月 15日		2019年 10月 30日
ISA/CN的名称和邮寄地址		受权官员
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		宫磊 电话号码 86-(10)-53961773

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2019/100495

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	109684307	A	2019年 4月 26日	无	
CN	109710190	A	2019年 5月 3日	无	
CN	103902698	A	2014年 7月 2日	无	
CN	108319661	A	2018年 7月 24日	无	
US	2014172898	A1	2014年 6月 19日	US 2016117363 A1	2016年 4月 28日
				EP 2932410 A2	2015年 10月 21日
				WO 2014093920 A2	2014年 6月 19日
				BR 112015011884 A2	2017年 7月 11日
				CN 105009111 A	2015年 10月 28日