

(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
2018 年 4 月 12 日 (12.04.2018)



(10) 国际公布号
WO 2018/064962 A1

(51) 国际专利分类号:
G06F 17/30 (2006.01)

(21) 国际申请号: PCT/CN2017/104515

(22) 国际申请日: 2017 年 9 月 29 日 (29.09.2017)

(25) 申请语言: 中文

(26) 公布语言: 中文

(30) 优先权:
201610879174.9 2016 年 10 月 8 日 (08.10.2016) CN

(71) 申请人: 腾讯科技(深圳)有限公司 (TENCENT TECHNOLOGY (SHENZHEN) COMPANY LIMITED) [CN/CN]; 中国广东省深圳市南山区高新区科技中一路腾讯大厦 35 层, Guangdong 518057 (CN)。

(72) 发明人: 肖涛 (XIAO, Tao); 中国广东省深圳市南山区高新区科技中一路腾讯大厦 35 层, Guangdong 518057 (CN)。

(74) 代理人: 广州华进联合专利商标代理有限公司 (ADVANCE CHINA IP LAW OFFICE); 中国广东省广州市天河区花城大道 85 号 3901 房, Guangdong 510623 (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,

(54) Title: DATA STORAGE METHOD, ELECTRONIC DEVICE AND COMPUTER NON-VOLATILE STORAGE MEDIUM

(54) 发明名称: 数据存储方法、电子设备和计算机非易失性存储介质

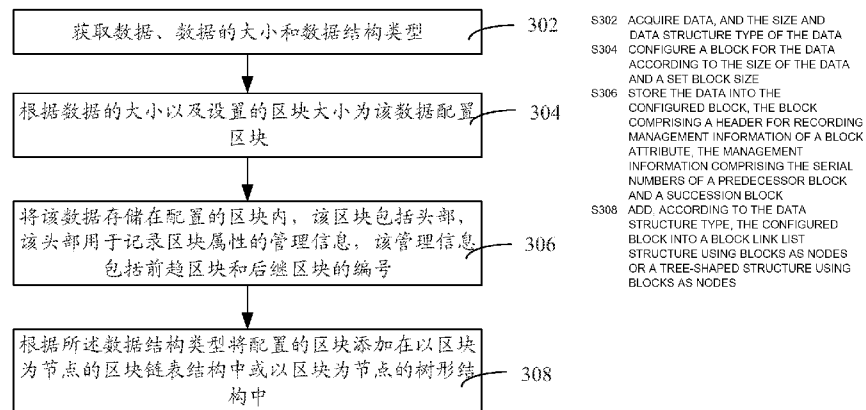


图 3

(57) Abstract: A data storage method, comprising: acquiring data, the size of the data and the data structure type of the data; configuring a block for the data according to the size of the data and a set block size; storing the data into the configured block, the block comprising a header for recording management information of a block attribute, the management information comprising the serial numbers of a predecessor block and a succession block; and adding, according to the data structure type, the configured block into a block link list structure using blocks as nodes or into a tree-shaped structure using blocks as nodes.

(57) 摘要: 一种数据存储方法包括: 获取数据, 以及所述数据的大小和所述数据的数据结构类型; 根据所述数据的大小以及设置的区块大小为所述数据配置区块; 将所述数据存储在所述配置的区块内, 所述区块包括头部, 所述头部用于记录区块属性的管理信息, 所述管理信息包括前趋和后继区块的编号; 根据所述数据结构类型将配置的区块添加在以区块为节点的区块链表结构或以区块为节点的树形结构中。

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))。

数据存储方法、电子设备和计算机非易失性存储介质

相关申请的交叉引用

本申请要求于 2016 年 10 月 08 日提交中国专利局、申请号为 201610879174.9、发明名称为“键值存储系统数据结构的实现方法和装置”
5 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

本发明涉及数据存储领域，特别是涉及一种数据存储方法、电子设备和计算机非易失性存储介质。

10

背景技术

键值存储系统是指以 key（键）-value（值）方式存储的数据存储系统，例如 redis 是一个高性能的 key-value 数据库。原生 redis 数据库通常采用私有内存存储数据，每个字符串元素和数据结构节点单独申请内存。由于采用私有内存存储和管理数据，使得原生 redis 的内存碎片问题比较严重，内存使用
15 率很低。

发明内容

根据本申请的各种实施例，提供一种数据存储方法、电子设备和计算机
20 非易失性存储介质。

一种数据存储方法，包括：

获取数据，以及所述数据的大小和所述数据的数据结构类型；

根据所述数据的大小以及设置的区块大小为所述数据配置区块；

将所述数据存储在所配置的区块内，所述区块包括头部，所述头部用于记录区块属性的管理信息，所述管理信息包括前趋和后继区块的编号；及
25 根据所述数据结构类型将配置的区块添加在以区块为节点的区块链表结

构中或以区块为节点的树形结构中。

一种电子设备，包括存储器及处理器，所述存储器中储存有计算机可读指令，所述指令被所述处理器执行时，使得所述处理器执行以下步骤：

获取数据，以及所述数据的大小和所述数据的数据结构类型；

5 根据所述数据的大小以及设置的区块大小为所述数据配置区块；；

将所述数据存储在所述配置的区块内，所述区块包括头部，所述头部用于记录区块属性的管理信息，所述管理信息包括前趋和后继区块的编号；及

根据所述数据结构类型将配置的区块添加在以区块为节点的区块链表结构中或以区块为节点的树形结构中。

10 一个或多个包含计算机可执行指令的非易失性计算机可读存储介质，当所述计算机可执行指令被一个或多个处理器执行时，使得所述处理器执行以下步骤：

获取数据，以及所述数据的大小和所述数据的数据结构类型；

根据所述数据的大小以及设置的区块大小为所述数据配置区块；；

15 将所述数据存储在所述配置的区块内，所述区块包括头部，所述头部用于记录区块属性的管理信息，所述管理信息包括前趋和后继区块的编号；及

根据所述数据结构类型将配置的区块添加在以区块为节点的区块链表结构中或以区块为节点的树形结构中。

20 本发明的一个或多个实施例的细节在下面的附图和描述中提出。本发明的其它特征、目的和优点将从说明书、附图以及权利要求书变得明显。

附图说明

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

图 1 为一个实施例中数据存储方法的应用环境示意图；

图 2 为一个实施例中电子设备的内部结构示意图；

图 3 为一个实施例中数据存储方法的流程图；

图 4 为字符串类型的数据存储在区块链表的示意图；

图 5 为链表数据结构类型存储数据的示意图；

5 图 6 为插入元素的流程示意图；

图 7 为删除元素的流程示意图；

图 8 为哈希值类型的数据存储示意图；

图 9 为字符串集合类型的数据存储示意图；

图 10 为排序字符串集合类型的数据存储示意图；

10 图 11 为一个实施例中数据存储装置的结构框图；

图 12 为另一个实施例中数据存储装置的结构框图。

具体实施方式

为了使本发明的目的、技术方案及优点更加清楚明白，以下结合附图及
15 实施例，对本发明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发明，并不用于限定本发明。

可以理解，本发明所使用的术语“第一”、“第二”等可在本文中用于描述各种元件，但这些元件不受这些术语限制。这些术语仅用于将第一个元件与另一个元件区分。举例来说，在不脱离本发明的范围的情况下，可以将第一判断模块称为第二判断模块，且类似地，可将第二判断模块称为第一判断模块。第一判断模块和第二判断模块两者都是判断模块，但其不是同一判断模块。

图 1 为一个实施例中数据存储方法的应用环境示意图。如图 1 所示，该应用环境包括存储设备等。该数据存储方法可在存储设备上实现，主要用于
25 在共享内存或磁盘等非私有内存的存储空间上实现键值存储系统数据结构。该存储设备可为各种存储数据的设备，如磁盘、U 盘、存储卡等。

图2为一个实施例中电子设备的内部结构示意图。如图2所示，该电子设

备包括通过系统总线连接的处理器、非易失性存储介质、内存储器 and 网络接口。该处理器用于提供计算和控制能力，支撑整个电子设备的运行。非易失性存储介质存储有操作系统和计算机可读指令，该计算机可读指令被处理器执行时以实现一种数据存储方法。内存储器为非易失性存储介质中的计算机可读指令的运行提供环境。该电子设备可以是带存储设备的手机、平板电脑或者个人数字助理或穿戴式设备或磁盘或硬盘或个人计算机等。本领域技术人员可以理解，图2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电子设备的限定，具体的电子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。

如图3所示，一种数据存储方法，可应用于键值存储系统实现数据存储，运行于图1的存储设备上，包括：

步骤302，获取数据、数据的大小和数据结构类型。

本实施例中，数据是指需要存储的数据。数据可为从网络中获取或通过设备采集的。数据的大小可为任意大小，如1字节、1KB（千字节）等。数据结构类型可为string（字符串类型）、list（链表类型）、hash（哈希值类型）、set（字符串集合类型）、zset（排序字符串集合类型）。

该键值存储系统可为redis等。

步骤304，根据数据的大小以及设置的区块大小为该数据配置区块。

具体地，每次配置的区块的大小可以是固定的。

本实施例中，每次申请新的区块时，配置的区块大小是相同的，且为固定大小，例如固定大小为128字节、256字节等，不限于此。

根据数据的大小以及设置的区块大小可以计算得到需要该为数据配置的区块的数量。

步骤306，将该数据存储于配置的区块内，该区块包括头部，该头部用于记录区块属性的管理信息，该管理信息包括前趋区块和后继区块的编号。

本实施例中，将数据的大小与区块的大小进行比较，若数据的大小小于

或等于区块的大小，可将数据直接存储在该区块内；若数据的大小大于区块的大小，则将数据拆分出与区块大小相等的部分数据存储在区块内，然后再配置新的区块，再将剩余的数据拆分出与区块大小相等的部分数据存储在新的区块中，依此类推，直到将数据全部存储在配置的全部区块内。

5 区块链表中每个区块包括头部 Head。头部用于记录区块链表和区块属性的管理信息。区块链表是指由所需的区块 block 构成的链表。每个区块的大小固定。管理信息可包括区块中存储的数据属于全部数据的哪个节点、存储的部分数据的大小、在区块链表中的前趋和后继区块的编号等一种或多种。区块链表中的前趋区块是指某一区块链接到的前一个区块。区块链表中的后
10 继区块是指某一区块链接到的后一个区块。区块编号是指用于唯一标识区块的字符串，例如 001、002 等。

平衡树中的各个区块的头部用于记录区块属性的管理信息。区块属性的管理信息中存储有父节点的区块编号、左子节点区块编号和右子节点区块编号，以及平衡系数等。平衡系数是指左子树树高与右子树树高的差值。父节点
15 的区块编号即为前趋区块编号，左子节点区块编号和右子节点区块编号即为后继区块编号。步骤 308，根据所述该数据结构类型将配置的区块添加在以区块为节点的区块链表结构中或以区块为节点的树形结构中。

在一个区块链表结构或树形结构中存在于一个区块用于记录元数据 (Metadata)，元数据可包括创建时间、最后修改时间、总大小、耗费区块 block 数量等。创建时间是指区块链表结构或树形结构创建的时刻。例如，2016 年 1
20 月 1 日 10 时 20 分 30 秒创建的区块链表结构或树形结构。最后修改时间是指区块链表结构或树形结构最后修改的时刻。例如，区块链表结构或树形结构被修改的最后时刻为 2016 年 1 月 5 日 10 时 20 分 30 秒。总大小是指数据的总量。耗费区块 block 数量是指所用的区块数量。例如，用 3 个区块存储了
25 20M 数据，耗费的区块数量是 3，数据总大小为 20M。

将配置的区块添加在以区块为节点的区块链表结构中或以区块为节点的树形结构中包括：获取配置的区块的编号，将该区块的编号添加到前一区块

的头部中，将该区块作为前一区块的后继区块，以及将前一区块的编号记录在该区块的头部中，将前一区块作为该区块的前趋区块。

以区块为节点的区块链表结构是指将配置的区块以链表形式关联起来形成的结构。以区块为节点的树形结构是指将配置的区块以平衡树形式关联起来形成的结构。平衡树是指一颗空树或它的左右两个子树的高度差的绝对值不超过 1，并且左右两个子树都是一颗平衡二叉树。平衡二叉树常用算法有红黑树、AVL（由 G.M. Adelson-Velsky 和 E.M. Landis 两人提出的平衡二叉树算法）、Treap（树堆）、伸展树、SBT（Size Balanced Tree）等。

上述数据存储方法，通过获取数据、数据大小和数据结构类型，配置固定大小的区块，根据数据大小将数据存储到配置的区块内，根据数据结构类型将配置的区块添加到区块链表结构或树形结构中，区块包括头部，在头部中记录了区块链表中的前趋区块和后继区块的编号，将存储数据的区块通过链表或树形结构关联起来，以固定大小的区块进行配置，使得数据存储更加紧凑，极大减少了管理的内存开销，提高了内存使用率。

在一个实施例中，上述数据存储方法还包括：若该数据结构类型为字符串 string 类型或链表类型 list，则根据该字符串 string 类型或链表类型 list 将配置的区块添加在以区块为节点的区块链表结构；若该数据结构类型为哈希值类型 hash 或字符串集合类型 set 或排序字符串集合类型 zset，则根据哈希值类型 hash 或字符串集合类型 set 或排序字符串集合类型 zset 将配置的各区块添加在以区块为节点的树形结构。

本实施例中，若待存储的数据的数据结构类型为字符串类型 string，该方法具体可如下：

获取字符串的大小，将字符串的大小与区块大小进行比较，若字符串的大小小于或等于区块大小，则将该字符串存储在该区块中。若字符串的大小大于区块大小，则将字符串以区块大小为单位拆成多段，根据字符串大小配置足够数量的区块，并通过区块的头部将区块链接为链表结构，再将字符串的每段对应存储在一个区块内。

图 4 为字符串类型的数据存储在区块链表的示意图。如图 4 所示，数据库 db (database) 中字符串数据结构类型的第一个区块存储有 Meta data (元数据) 和 String part 1 (同一字符串的第一部分)，第二区块包括头部 Head 和存储有 String part 2 (同一字符串的第一部分)，第三区块包括头部 Head 和存储有 String part 3 (同一字符串的第一部分)。一个完整的字符串即为一个元素。Meta data 中存储有任意元数据，包括创建时间、最后修改时间、总长度、耗费区块数量等。Head 中记录区块链表和区块属性的管理信息，该管理信息可包括区块属于字符串的一个节点、存储的 part 的长度、在区块链表中的前趋区块和后继区块的编号等。例如，第一个区块的编号为 block1，第二个区块的编号为 block2，第三个区块的编号为 block3，第二个区块的头部 Head 存储有第一个区块的编号 block1 和第三个区块的编号 block3。

(2) 数据结构类型为链表类型 List。

具体地，链表类型是一种物理存储单元上非连续、非顺序的存储结构，数据元素的逻辑顺序是通过链表中的指针链接次序实现的。链表中头部存储有指向下一个存储地址的指针。将链表的元素存储在区块链表中，每个区块存储若干元素，每个元素的 string 可放在当前区块，或只放一个索引，将字符串记录在外部的区块链表中。每个区块中存储的元素数量与区块的大小与元素的大小有关，例如，一个区块大小为 10M，4 个元素大小分别为 1M、3M、5M、4M，则该区块可存储前 3 个元素，第 4 个元素无法存储在该区块中。

在一个实施例中，上述数据存储方法还包括：若数据结构类型为链表 List 类型，获取该数据结构类型中元素的字符串的大小；判断该元素的字符串的大小是否小于或等于当前要存储的区块的剩余空间大小，若是，则将该元素的字符串存储在该当前要存储的区块内，若否，则获取该元素的字符串的索引大小，判断该元素的字符串的索引是否小于或等于当前要存储的区块的剩余空间大小，若是，则将该元素的字符串的索引存储在该当前要存储的区块内，并将该元素的字符串存储在该索引对应的外部区块链表中。

具体地，例如，当前区块的空间大小为 100 字节，去掉 list 节点的头部

Head 后剩余 80 字节空间，若一个元素的字符串 string 的大小比 80 字节短，则可直接存储在当前区块中，否则将该元素的字符串的索引存储在当前区块中，将元素的字符串存储在外部区块链表中。外部区块链表是指在链表数据结构类型的区块节点外的区块构成的区块链表，用于存储一个字符串 string 的区块链表。区块节点外的区块是指与分配的区块节点不在一个区块链表中的区块。

进一步的，将该元素的字符串存储在该索引对应的外部区块链表中包括：判断元素的字符串大小是否小于或等于外部区块的大小，若是，则将该元素的字符串存储在外部区块中，若否，则将元素的字符串按照外部区块的大小拆成多段，配置对应数量的外部区块，对应数量的外部区块关联起来构成外部区块链表，将元素的字符串存储在对应数量的外部区块链表中。

图 5 为链表数据结构类型存储数据的示意图。如图 5 所示，数据库 db (database) 中链表数据结构类型的第一个区块存储有 Meta data (元数据) 和 List Object，第二区块包括头部 Head 和存储有 Elem 1、Elem 2...，第三区块包括头部 Head 和存储有 Elem i、Elem i+1...。Meta data 中存储有任意元数据，包括创建时间、最后修改时间、总长度、耗费区块数量等。Head 中记录区块链表和区块属性的管理信息，该管理信息可包括区块属于字符串的一个节点、存储的 part 的长度、在区块链表中的前趋区块和后继区块的编号等。第二区块中存储的第一个元素 Elem 1 是一个足够短的字符串，比如“abc”，加上一个长度管理，采用明文+冒号表示长度，则元素的字符串为“3:abc”共 5 个字节，能够存储在第二区块中，不需要做外部链接索引。第二个元素 Elem 2 非常长，有几百字节，不可能存储在第二区块中，则用外部区块链表存储，在链表数据类型的区块节点中存储第二个元素的字符串的索引，即带方框的 Elem 2。在外部区块链表存储该第二个元素的字符串。

在一个实施例中，上述数据存储方法还包括：获取数据访问请求；根据该数据访问请求从区块链表中定位到对应的区块，遍历该区块得到所需的数据。

在一个实施例中，当一个区块 block 中插入元素而导致数据存储不下时，将区块中的数据分裂，存储到其他的区块中。应用层需要将元素插入到某个区块中时，将元素插入到对应的区块中。

如图 6 所示，上述键值存储系统数据结构的实现方法还包括：

5 步骤 602，获取要插入的元素、第一目标区块和在该第一目标区块内的插入位置。

若为区块链表，则根据第一目标区块的编号查找到对应的区块。若为平衡树，则采用平衡二叉树的查找方式，将第一目标区块的区块编号与平衡树中根节点的区块编号比较，若第一目标区块的区块编号小于平衡树中根节点的区块编号，则从左子树节点查找，若第一目标区块的区块编号大于平衡树中根节点的区块编号，则从右子树节点查找，依次类推，直到查找到与第一目标区块编号相同的区块。

步骤 604，判断该要插入的元素的大小是否超过该第一目标区块的剩余空间大小。

15 步骤 606，若该要插入的元素的大小超过该第一目标区块的剩余空间大小，则将该要插入的元素存储在该第一目标区块内的插入位置处，并将该第一目标区块内的插入位置处的后续元素向该插入位置处的后续位置移动，配置新的固定大小的区块，将该后续位置中超过该第一目标区块的元素存储在该新的区块内。将第一目标区块中存储的后继区块编号修改为新的区块，将新的区块中 Head 中记录的后继区块编号配置为第一目标区块中记录的原来的后继区块的编号，将新的区块中 Head 中记录的前趋区块编号配置为第一目标区块的编号。

例如，假设一个区块 A 存放了“abc”、“def”和“ghi”三个字符串，若需要在“def”和“ghi”中插入一个“123”字符串，插入后区块 A 已满，无法存储“ghi”，则配置一个新的空闲区块 B，和原有区块 A 分别存放“abc”、“def”和“123”，“ghi”。

在一个实施例中，当一个区块因为删除元素而空出多余空间时，检查该

区块的前趋和后继区块，进行合并。

如图 7 所示，上述键值存储系统数据结构的实现方法还包括：

步骤 702，获取删除元素指令，该删除元素指令中包括指定元素及该指定元素所处的第二目标区块。

5 步骤 704，根据该删除元素指令从该第二目标区块中删除该指定元素。

步骤 706，判断该第二目标区块的后继区块中元素的字符串的大小是否小于或等于该第二目标区块的剩余空间大小，若是，执行步骤 708，若否，执行步骤 710。

10 步骤 708，将该第二目标区块的后继区块中元素的字符串合并到该第二目标区块中。

步骤 710，不作处理。

通过删除元素后，将删除元素的区块与其后继区块内的数据合并成一个区块，可以释放一个区块空间，避免浪费。

(3) 数据结构类型为哈希值类型 hash。

15 利用平衡树来组织各区块，每个区块内部可存储多个键值对。在平衡树中，key-value（键值）对的分布是按 key 的内容有序排列，比如一个元素的 key 是 abc，则必定在另一个 key 为 def 的元素之前。

平衡树组织各区块即采用平衡二叉树原理组织区块，其是一颗空树或它的左右两个子树的高度差的绝对值不超过 1，并且左右两个子树都是一颗平衡二叉树。例如，首先选择一个区块编号为 1 的区块为根节点，大于父节点的区块编号的位于左子树，小于父节点区别编号的位于右子树。区块编号 2 和区块编号 3 添加到区块编号 1 后，均位于右子树，导致平衡树不平衡，则进行旋转，将区块编号 2 变为父节点，区块编号 1 添加到区块编号 2 的左子树，区块编号 3 添加到区块编号 2 的右子树，后续添加多个区块采用同样的
20 处理方式。平衡树中的各个区块的 Head 中存储有父节点的区块编号、左子节点区块编号和右子节点区块编号，以及平衡系数。平衡系数是指左子树树高与右子树树高的差值。父节点的区块编号即为前趋区块编号，左子节点区块
25

编号和右子节点区块编号即为后继区块编号。

图 8 为哈希值类型的数据存储示意图。如图 8 所示，数据库 db(database) 中哈希值数据结构类型的第一个区块存储有 Meta data(元数据)和 Hash Object。第二区块包括 Head 和存储有 Key M Value M、Key N Value N。第二区块链接到第三区块和第四区块。第三区块包括 Head 和存储有 Key G Value G、Key H Value H。第三区块链接到第五区块和第六区块。第五区块包括 Head 和存储有 Key D Value D、Key F Value F。第六区块包括 Head 和存储有 Key J Value J、Key K Value K。第四区块包括 Head 和存储有 Key X Value X。第四区块链接到第七区块。第七区块包括 Head 和存储有 Key Z Value Z。

在一个实施例中，上述数据存储方法还包括：若该数据结构类型为哈希值类型 hash，获取该数据结构类型中元素的字符串的大小；判断该元素的字符串的大小是否小于或等于当前要存储的区块的剩余空间大小，若是，则将该元素的字符串存储在该当前要存储的区块内，若否，则获取该元素的字符串的索引大小，判断该元素的字符串的索引是否小于或等于当前要存储的区块的剩余空间大小，若是，则将该元素的字符串的索引存储在该当前要存储的区块内，并将该元素的字符串存储在该索引对应的外部区块链表中。

在一个实施例中，上述数据存储方法还包括：若该数据结构类型为哈希值类型 hash，获取要插入的元素、第一目标区块和在该第一目标区块内的插入位置；判断该要插入的元素的大小是否超过该第一目标区块的剩余空间大小；若该要插入的元素的大小超过该第一目标区块的剩余空间大小，则将该要插入的元素存储在该第一目标区块内的插入位置处，并将该第一目标区块内的插入位置处的后续元素向该插入位置处的后续位置移动，配置新的固定大小的区块，将该后续位置中超过该第一目标区块的元素存储在该新的区块内。

在一个实施例中，上述数据存储方法还包括：若该数据结构类型为哈希值类型 hash，获取删除元素指令，该删除元素指令中包括指定元素及该指定元素所处的第二目标区块；根据该删除元素指令从该第二目标区块中删除该

指定元素；判断该第二目标区块的后继区块中元素的字符串的大小是否小于或等于该第二目标区块的剩余空间大小，若是，则将该第二目标区块的后继区块中元素的字符串合并到该第二目标区块中。

在插入元素或删除元素时，采用平衡树算法查找区块的前趋和后继区块。

5 在一个实施例中，上述数据存储方法还包括：获取数据访问请求；根据该数据访问请求从树形结构中定位到对应的区块，遍历该区块得到所需的数据。

具体地，通过平衡树算法定位到对应的区块。

(4) 数据结构类型为字符串集合类型 set。

10 利用平衡树来组织各区块，每个区块内部包括 key。

图 9 为字符串集合类型的数据存储示意图。如图 9 所示，数据库 db (database) 中字符串集合数据结构类型的第一个区块存储有 Meta data (元数据) 和 set Object。第二区块包括 Head 和存储有 Key M、Key N。第二区块链接到第三区块和第四区块。第三区块包括 Head 和存储有 Key G、Key H。第三区块链接到第五区块和第六区块。第五区块包括 Head 和存储有 Key D、Key F。第六区块包括 Head 和存储有 Key J、Key K。第四区块包括 Head 和存储有 Key X。第四区块链接到第七区块。第七区块包括 Head 和存储有 Key Z。

15 在一个实施例中，上述数据存储方法还包括：若该数据结构类型为字符串集合类型 set，获取该数据结构类型中元素的字符串的大小；判断该元素的字符串的大小是否小于或等于当前要存储的区块的剩余空间大小，若是，则将该元素的字符串存储在该当前要存储的区块内，若否，则获取该元素的字符串的索引大小，判断该元素的字符串的索引是否小于或等于当前要存储的区块的剩余空间大小，若是，则将该元素的字符串的索引存储在该当前要存储的区块内，并将该元素的字符串存储在该索引对应的外部区块链表中。

25 在一个实施例中，上述数据存储方法还包括：若该数据结构类型为字符串集合类型 set，获取要插入的元素，第一目标区块和在该第一目标区块内的

插入位置；判断该要插入的元素的大小是否超过该第一目标区块的剩余空间大小；若该要插入的元素的大小超过该第一目标区块的剩余空间大小，则将该要插入的元素存储在该第一目标区块内的插入位置处，并将该第一目标区块内的插入位置处的后续元素向该插入位置处的后续位置移动，配置新的固定大小的区块，将该后续位置中超过该第一目标区块的元素存储在该新的区块内。

在一个实施例中，上述数据存储方法还包括：若该数据结构类型为字符串集合 set 类型，获取删除元素指令，该删除元素指令中包括指定元素及该指定元素所处的第二目标区块；根据该删除元素指令从该第二目标区块中删除该指定元素；判断该第二目标区块的后继区块中元素的字符串的大小是否小于或等于该第二目标区块的剩余空间大小，若是，则将该第二目标区块的后继区块中元素的字符串合并到该第二目标区块中。

在插入元素或删除元素时，采用平衡树算法查找区块的前趋和后继区块。

(5) 数据结构类型为排序字符串集合类型 zset。

具体地，每个 zset 包括两部分，一个是数据集合，结构同 set，另一个是根据 score 的排序，采用平衡树实现。

图 10 为排序字符串集合类型的数据存储示意图。如图 10 所示，数据库 db (database) 中排序字符串集合数据结构类型的第一个区块存储有 Meta data (元数据) 和 Zset Object。第二区块包括 Head 和存储有 Score M、Elem M、Score N、Elem N。第二区块链接到第三区块和第四区块。第三区块包括 Head 和存储有 Score G、Elem G、Score H、Elem H。第三区块链接到第五区块和第六区块。第五区块包括 Head 和存储有 Score D、Elem D、Score F、Elem F。第六区块包括 Head 和存储有 Score J、Elem J、Score K、Elem K。第四区块包括 Head 和存储有 Score X、Elem X。第四区块链接到第七区块。第七区块包括 Head 和存储有 Score Z、Elem Z。

在一个实施例中，上述数据存储方法还包括：若该数据结构类型为排序字符串集合类型 zset，获取该数据结构类型中元素的字符串的大小；判断该

元素的字符串的大小是否小于或等于当前要存储的区块的剩余空间大小，若是，则将该元素的字符串存储在该当前要存储的区块内，若否，则获取该元素的字符串的索引大小，判断该元素的字符串的索引是否小于或等于当前要存储的区块的剩余空间大小，若是，则将该元素的字符串的索引存储在该当前要存储的区块内，并将该元素的字符串存储在该索引对应的外部区块链表中。

在一个实施例中，上述数据存储方法还包括：若该数据结构类型为排序字符串集合类型 `zset`，获取要插入的元素、第一目标区块和在该第一目标区块内的插入位置；判断该要插入的元素的大小是否超过该第一目标区块的剩余空间大小；若该要插入的元素的大小超过该第一目标区块的剩余空间大小，则将该要插入的元素存储在该第一目标区块内的插入位置处，并将该第一目标区块内的插入位置处的后续元素向该插入位置处的后续位置移动，配置新的固定大小的区块，将该后续位置中超过该第一目标区块的元素存储在该新的区块内。

在一个实施例中，上述数据存储方法还包括：若该数据结构类型为排序字符串集合类型 `zset`，获取删除元素指令，该删除元素指令中包括指定元素及该指定元素所处的第二目标区块；根据该删除元素指令从该第二目标区块中删除该指定元素；判断该第二目标区块的后继区块中元素的字符串的大小是否小于或等于该第二目标区块的剩余空间大小，若是，则将该第二目标区块的后继区块中元素的字符串合并到该第二目标区块中。

在插入元素或删除元素时，采用平衡树算法查找区块的前趋和后继区块。

在一个实施例中，上述数据存储方法还包括：若该数据结构类型为哈希值类型 `hash` 或字符串集合类型 `set` 或排序字符串集合类型 `zset` 时，获取树形结构的各区块中的键的哈希值，根据该键的哈希值将该键分成多份，将每份的键采用独立的平衡树进行存储；获取键的查询指令；根据该查询指令从独立的平衡树中查找该键。

因将键分成了 N 份，每一份采用独立的平衡树进行存储，查找一个键 `key`

的工作量是先计算在哪颗树，然后在该树中查找，由于每棵树的数据量只有整体的 $1/N$ ，所以速度会提升。其中， N 为自然数。

需要说明的是，相对于原生的 redis，通过本发明的键值存储系统数据结构的实现方法，可将数据紧凑的存储，极大减少了数据管理的内存开销，同时，由于支持固定大小内存申请释放的管理方式，可以方便地在共享内存、磁盘等非私有内存的存储空间上实现 redis 数据结构。

图 11 为一个实施例中数据存储装置的结构框图。如图 11 所示，一种数据存储装置 1100，包括数据获取模块 1102、配置模块 1104、存储模块 1106 和添加模块 1108。其中：

数据获取模块 1102 用于获取数据、数据的大小和数据结构类型。

配置模块 1104 用于根据所述数据的大小以及设置的区块大小为所述数据配置区块。每次配置固定大小的区块。

存储模块 1106 用于将该数据存储在该区块内，该区块包括头部，该头部用于记录区块属性的管理信息，该管理信息包括前趋和后继区块的编号。区块链表中的区块包括头部，该头部用于记录区块链表和区块属性的管理信息，该管理信息包括在该区块链表中的前趋和后继区块的编号。

添加模块 1108 用于根据该数据结构类型将配置的区块添加在以区块为节点的区块链表结构中或以区块为节点的树形结构中。

上述数据存储装置，通过获取数据、数据大小和数据结构类型，配置固定大小的区块，根据数据大小将数据存储 in 配置的区块内，根据数据结构类型将配置的区块添加到区块链表结构或树形结构中，区块包括头部，在头部中记录了区块链表中的前趋区块和后继区块的编号，将存储数据的区块通过链表或树形结构关联起来，以固定大小的区块进行配置，使得数据存储更加紧凑，极大减少了管理的内存开销，提高了内存使用率。

在一个实施例中，添加模块 1108 还用于若该数据结构类型为字符串 string 类型或链表类型 list，则根据该字符串 string 类型或链表类型 list 将配置的区

块添加在以区块为节点的区块链表结构；以及

若该数据结构类型为哈希值类型 hash 或字符串集合类型 set 或排序字符串集合类型 zset，则根据哈希值类型 hash 或字符串集合类型 set 或排序字符串集合类型 zset 将配置的各区块添加在以区块为节点的树形结构。

5 图 12 为另一个实施例中数据存储装置的结构框图。如图 12 所示，一种数据存储装置 1200，包括数据获取模块 1202、配置模块 1204、存储模块 1206、添加模块 1208、大小获取模块 1210、第一判断模块 1212、插入元素获取模块 1214、第二判断模块 1216、删除指令获取模块 1218、删除模块 1220、第三判断模块 1222、合并模块 1224、分类模块 1226、查询模块 1228 和查找模
10 块 1230。其中：

大小获取模块 1210 用于若该数据结构类型为链表类型 list 或哈希值类型 hash 或字符串集合类型 set 或排序字符串集合类型 zset 时，则获取该数据结构类型中元素的字符串的大小。

第一判断模块 1212 用于判断该元素的字符串的大小是否小于或等于当前要存储的区块的剩余空间大小，若是，则该存储模块 1206 还用于将该元素的字符串存储在该当前要存储的区块内，若否，则该大小获取模块 1210 还用于获取该元素的字符串的索引大小，该第一判断模块 1212 还用于判断该元素的字符串的索引是否小于或等于当前要存储的区块的剩余空间大小，若是，
15 则该存储模块 1206 还用于将该元素的字符串的索引存储在该当前要存储的区块内，并将该元素的字符串存储在该索引对应的外部区块链表中。

插入元素获取模块 1214 用于获取要插入的元素、第一目标区块和在该第一目标区块内的插入位置。

第二判断模块 1216 用于判断该要插入的元素的大小是否超过该第一目标区块的剩余空间大小。

25 存储模块 1206 还用于若该要插入的元素的大小超过该第一目标区块的剩余空间大小，则将该要插入的元素存储在该第一目标区块内的插入位置处，并将该第一目标区块内的插入位置处的后续元素向该插入位置处的后续位置

移动，配置新的固定大小的区块，将该后续位置中超过该第一目标区块的元素存储在该新的区块内。

删除指令获取模块 1218 用于获取删除元素指令，该删除元素指令中包括指定元素及该指定元素所处的第二目标区块。

5 删除模块 1220 用于根据该删除元素指令从该第二目标区块中删除该指定元素。

第三判断模块 1222 用于判断该第二目标区块的后继区块中元素的字符串的大小是否小于或等于该第二目标区块的剩余空间大小。

10 合并模块 1224 用于当该第二目标区块的后继区块中元素的字符串的大小小于或等于该第二目标区块的剩余空间大小时，将该第二目标区块的后继区块中元素的字符串合并到该第二目标区块中。

15 分类模块 1226 用于若该数据结构类型为哈希值类型 hash 或字符串集合类型 set 或排序字符串集合类型 zset 时，获取树形结构的各区块中的键的哈希值，根据该键的哈希值将该键分成多份，将每份的键采用独立的平衡树进行存储。

查询模块 1228 用于获取键的查询指令。

查找模块 1230 用于根据该查询指令从独立的平衡树中查找该键。

20 在其他实施例中，一种数据存储装置，可包括数据获取模块 1202、配置模块 1204、存储模块 1206、添加模块 1208、大小获取模块 1210、第一判断模块 1212、插入元素获取模块 1214、第二判断模块 1216、删除指令获取模块 1218、删除模块 1220、第三判断模块 1222、合并模块 1224、分类模块 1226、查询模块 1228 和查找模块 1230 中任意可能的组合，不限于所列举的实施例。

25 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，所述的存储介质可为磁碟、光盘、只读存储记忆体（Read-Only Memory, ROM）等。

以上所述实施例仅表达了本发明的几种实施方式，其描述较为具体和详细，但并不能因此而理解为对本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，在不脱离本发明构思的前提下，还可以做出若干变形和改进，这些都属于本发明的保护范围。因此，本发明专利的保护范围应以所附权利要求为准。

权利要求书

1、一种键值存储系统的数据存储方法，包括：

获取数据，以及所述数据的大小和所述数据的数据结构类型；

根据所述数据的大小以及设置的区块大小为所述数据配置区块；

5 将所述数据存储在所配置的区块内，所述区块包括头部，所述头部用于记录区块属性的管理信息，所述管理信息包括前趋和后继区块的编号；

根据所述数据结构类型将配置的区块添加在以区块为节点的区块链表结构中或以区块为节点的树形结构中。

2、根据权利要求1所述的方法，其特征在于，所述数据结构类型为字符串 string 类型或链表类型 list，所述方法还包括根据所述字符串 string 类型或
10 链表类型 list 将配置的区块添加在以区块为节点的区块链表结构。

3、根据权利要求1所述的方法，其特征在于，所述数据结构类型为哈希值类型 hash 或字符串集合类型 set 或排序字符串集合类型 zset，所述方法还包括根据哈希值类型 hash 或字符串集合类型 set 或排序字符串集合类型 zset 将配置
15 的各区块添加在以区块为节点的树形结构。

4、根据权利要求1所述的方法，其特征在于，所述方法还包括：

若所述数据结构类型为链表类型 list 或哈希值类型 hash 或字符串集合类型 set 或排序字符串集合类型 zset 时，则获取所述数据结构类型中元素的字符串的大小；

20 判断所述元素的字符串的大小是否小于或等于当前要存储的区块的剩余空间大小，若是，则将所述元素的字符串存储在所述当前要存储的区块内，若否，则获取所述元素的字符串的索引大小，判断所述元素的字符串的索引是否小于或等于当前要存储的区块的剩余空间大小，若是，则将所述元素的字符串的索引存储在所述当前要存储的区块内，并将所述元素的字符串存储在所述索引对应的外部区块链表中。

25 5、根据权利要求4所述的方法，其特征在于，所述将所述元素的字符串存储在所述索引对应的外部区块链表中包括：

判断所述元素的字符串大小是否小于或等于外部区块的大小，若是，则将所述元素的字符串存储在外部区块中，若否，则将所述元素的字符串按照外部区块的大小拆成多段，配置对应数量的外部区块，对应数量的外部区块关联起来构成外部区块链表，将元素的字符串存储在对应数量的外部区块链表中。

6、根据权利要求 1 所述的方法，其特征在于，还包括：

获取要插入的元素、第一目标区块和在所述第一目标区块内的插入位置；判断所述要插入的元素的大小是否超过所述第一目标区块的剩余空间大小；

若所述要插入的元素的大小超过所述第一目标区块的剩余空间大小，则将所述要插入的元素存储在所述第一目标区块内的插入位置处，并将所述第一目标区块内的插入位置处的后续元素向所述插入位置处的后续位置移动，配置新的固定大小的区块，将所述后续位置中超过所述第一目标区块的元素存储在所述新的区块内。

7、根据权利要求 1 所述的方法，其特征在于，还包括：

获取删除元素指令，所述删除元素指令中包括指定元素及所述指定元素所处的第二目标区块；

根据所述删除元素指令从所述第二目标区块中删除所述指定元素；以及

判断所述第二目标区块的后继区块中元素的字符串的大小是否小于或等于所述第二目标区块的剩余空间大小，若是，则将所述第二目标区块的后继区块中元素的字符串合并到所述第二目标区块中。

8、根据权利要求 1 所述的方法，其特征在于，还包括：

若所述数据结构类型为哈希值类型 hash 或字符串集合类型 set 或排序字符串集合类型 zset 时，获取树形结构的各区块中的键的哈希值，根据所述键的哈希值将所述键分成多份，将每份的键采用独立的平衡树进行存储；

获取键的查询指令；以及

根据所述查询指令从独立的平衡树中查找所述键。

9、根据权利要求 1 所述的方法，其特征在于，所述将配置的区块添加在以区块为节点的区块链表结构中或以区块为节点的树形结构中包括：

获取配置的区块的编号，将所述区块的编号添加到前一区块的头部中，将所述区块作为前一区块的后继区块，以及将所述前一区块的编号记录在所述区块的头部中，将所述前一区块作为所述区块的前趋区块。

10、根据权利要求 1 所述的方法，其特征在于，还包括：

获取数据访问请求；以及

根据所述数据访问请求从所述区块链表中定位到对应的区块，遍历所述区块得到所需的数据。

11、一种电子设备，包括存储器及处理器，所述存储器中储存有计算机可读指令，所述指令被所述处理器执行时，使得所述处理器执行以下步骤：

获取数据、数据的大小和数据结构类型；

根据所述数据的大小以及设置的区块大小为所述数据配置区块；

将所述数据存储在该配置的区块内，所述区块包括头部，所述头部用于记录区块属性的管理信息，所述管理信息包括前趋和后继区块的编号；

根据所述数据结构类型将配置的区块添加在以区块为节点的区块链表结构中或以区块为节点的树形结构中。

12、根据权利要求 11 所述的电子设备，其特征在于，所述处理器还被用于执行以下步骤：若所述数据结构类型为字符串 string 类型或链表类型 list，根据所述字符串 string 类型或链表类型 list 将配置的区块添加在以区块为节点的区块链表结构；

若所述数据结构类型为哈希值类型 hash 或字符串集合类型 set 或排序字符串集合类型 zset，根据哈希值类型 hash 或字符串集合类型 set 或排序字符串集合类型 zset 将配置的各区块添加在以区块为节点的树形结构。

13、根据权利要求 11 所述的电子设备，其特征在于，所述处理器还被用于执行以下步骤：

若所述数据结构类型为链表类型 list 或哈希值类型 hash 或字符串集合类

型 set 或排序字符串集合类型 zset 时,则获取所述数据结构类型中元素的字符串的大小;

判断所述元素的字符串的大小是否小于或等于当前要存储的区块的剩余空间大小,若是,则将所述元素的字符串存储在所述当前要存储的区块内, 5 若否,则获取所述元素的字符串的索引大小,判断所述元素的字符串的索引是否小于或等于当前要存储的区块的剩余空间大小,若是,则将所述元素的字符串的索引存储在所述当前要存储的区块内,并将所述元素的字符串存储在所述索引对应的外部区块链表中。

14、根据权利要求 13 所述的电子设备,其特征在于,所述将所述元素的 10 字符串存储在所述索引对应的外部区块链表中包括:

判断所述元素的字符串大小是否小于或等于外部区块的大小,若是,则将所述元素的字符串存储在外部区块中,若否,则将所述元素的字符串按照外部区块的大小拆成多段,配置对应数量的外部区块,对应数量的外部区块 15 关联起来构成外部区块链表,将元素的字符串存储在对应数量的外部区块链表中。

15、根据权利要求 11 所述的电子设备,其特征在于,所述处理器还被用于执行以下步骤:

获取要插入的元素、第一目标区块和在所述第一目标区块内的插入位置;

判断所述要插入的元素的大小是否超过所述第一目标区块的剩余空间大 20 小;

若所述要插入的元素的大小超过所述第一目标区块的剩余空间大小,则将所述要插入的元素存储在所述第一目标区块内的插入位置处,并将所述第一目标区块内的插入位置处的后续元素向所述插入位置处的后续位置移动,配置新的固定大小的区块,将所述后续位置中超过所述第一目标区块的元素 25 存储在所述新的区块内。

16、根据权利要求 11 所述的电子设备,其特征在于,所述处理器还被用于执行以下步骤:

获取删除元素指令，所述删除元素指令中包括指定元素及所述指定元素所处的第二目标区块；

根据所述删除元素指令从所述第二目标区块中删除所述指定元素；以及

判断所述第二目标区块的后继区块中元素的字符串的大小是否小于或等于所述第二目标区块的剩余空间大小，若是，则将所述第二目标区块的后继区块中元素的字符串合并到所述第二目标区块中。

17、根据权利要求 11 所述的电子设备，其特征在于，所述处理器还被用于执行以下步骤：

若所述数据结构类型为哈希值类型 hash 或字符串集合类型 set 或排序字符串集合类型 zset 时，获取树形结构的各区块中的键的哈希值，根据所述键的哈希值将所述键分成多份，将每份的键采用独立的平衡树进行存储；

获取键的查询指令；以及

根据所述查询指令从独立的平衡树中查找所述键。

18、根据权利要求 11 所述的电子设备，其特征在于，所述将配置的区块添加在以区块为节点的区块链表结构中或以区块为节点的树形结构中包括：

获取配置的区块的编号，将所述区块的编号添加到前一区块的头部中，将所述区块作为前一区块的后继区块，以及将所述前一区块的编号记录在所述区块的头部中，将所述前一区块作为所述区块的前趋区块。

19、根据权利要求 11 所述的电子设备，其特征在于，所述处理器还被用于执行以下步骤：

获取数据访问请求；以及

根据所述数据访问请求从所述区块链表中定位到对应的区块，遍历所述区块得到所需的数据。

20、一个或多个包含计算机可执行指令的非易失性计算机可读存储介质，当所述计算机可执行指令被一个或多个处理器执行时，使得所述处理器执行以下步骤：

获取数据，以及所述数据的大小和所述数据的数据结构类型；

根据所述数据的大小以及设置的区块大小为所述数据配置区块；

将所述数据存储在该所述配置的区块内，所述区块包括头部，所述头部用于记录区块属性的管理信息，所述管理信息包括前趋和后继区块的编号；及

根据所述数据结构类型将配置的区块添加在以区块为节点的区块链表结构5 中或以区块为节点的树形结构中。

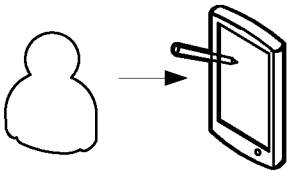


图 1

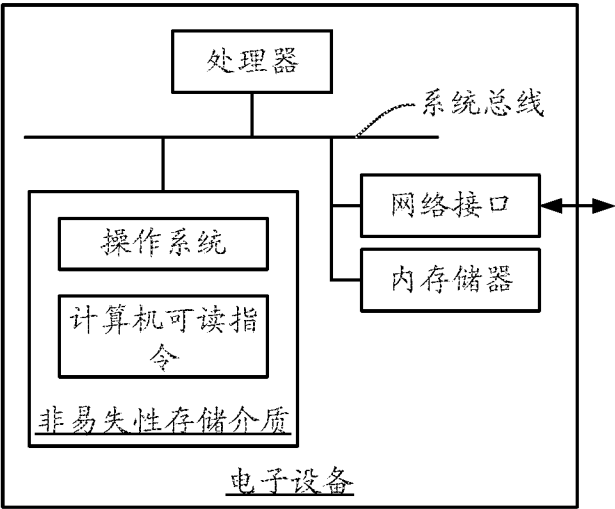


图 2

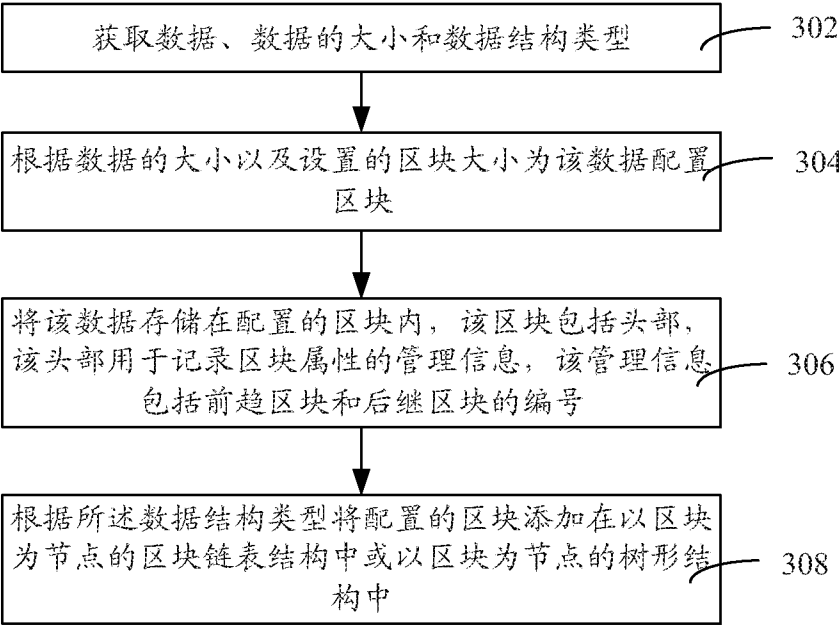


图 3

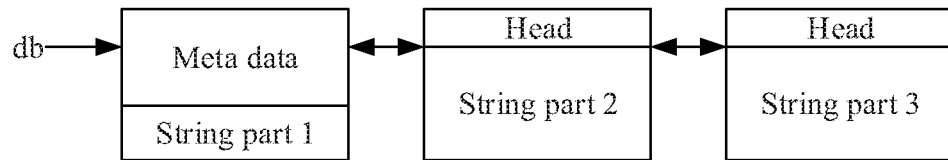


图 4

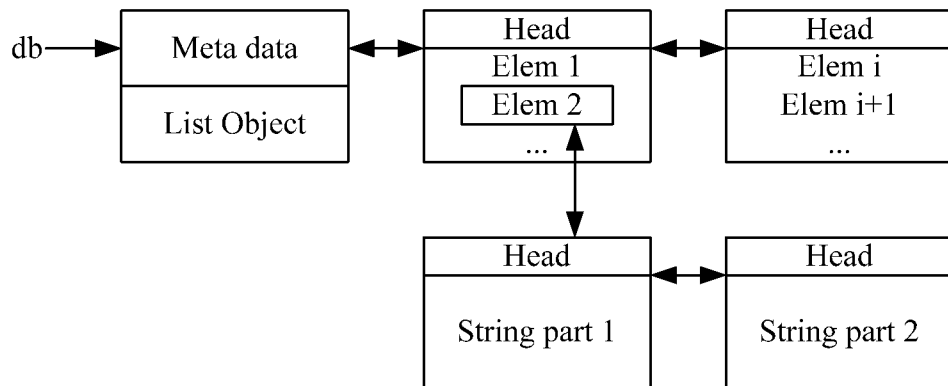


图 5

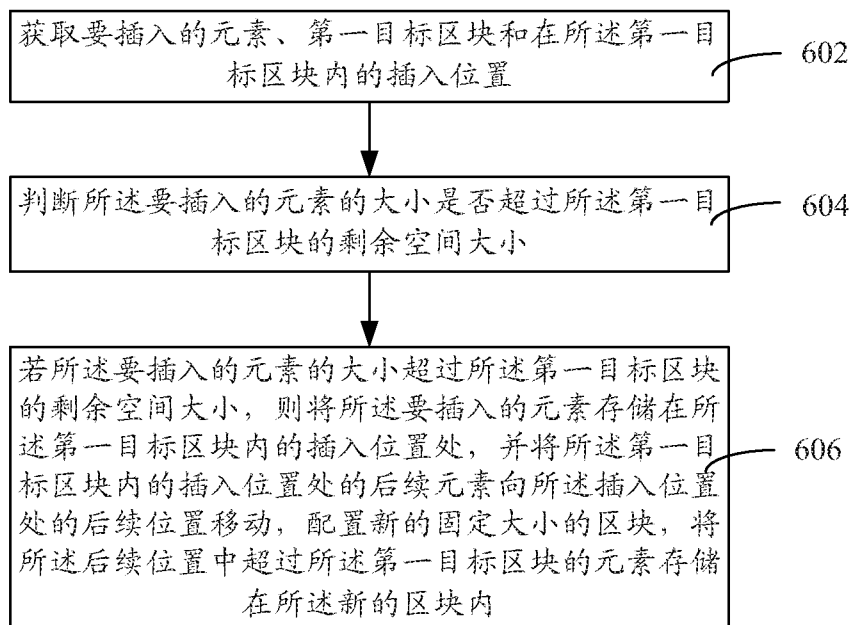


图 6

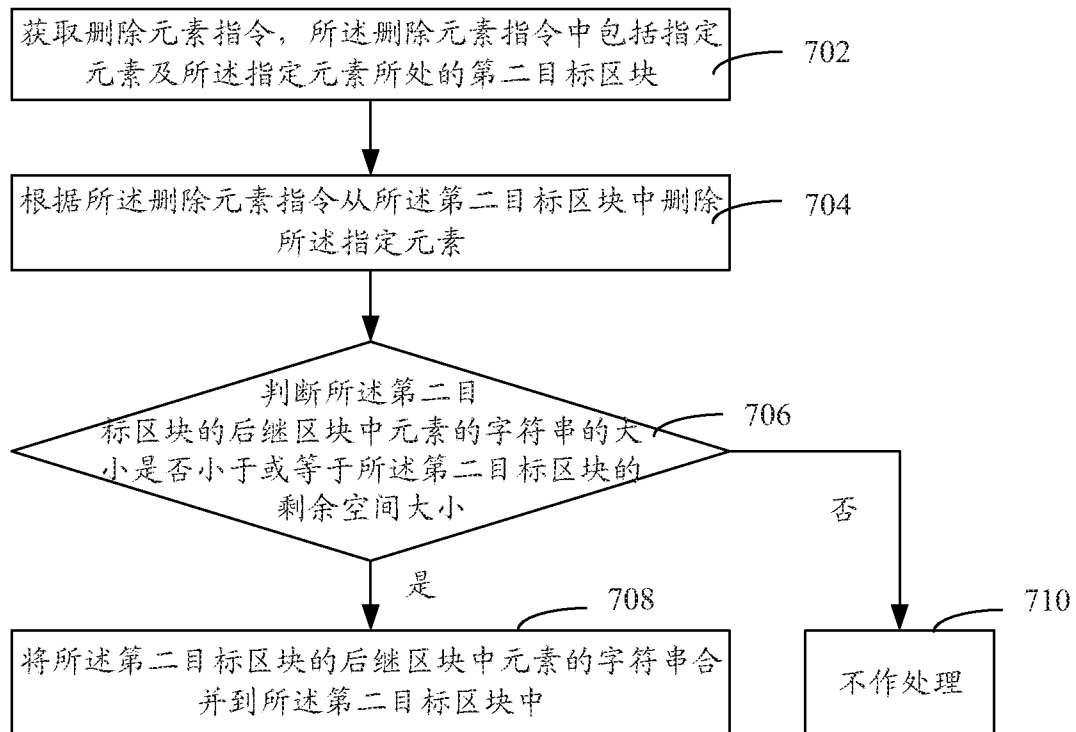


图 7

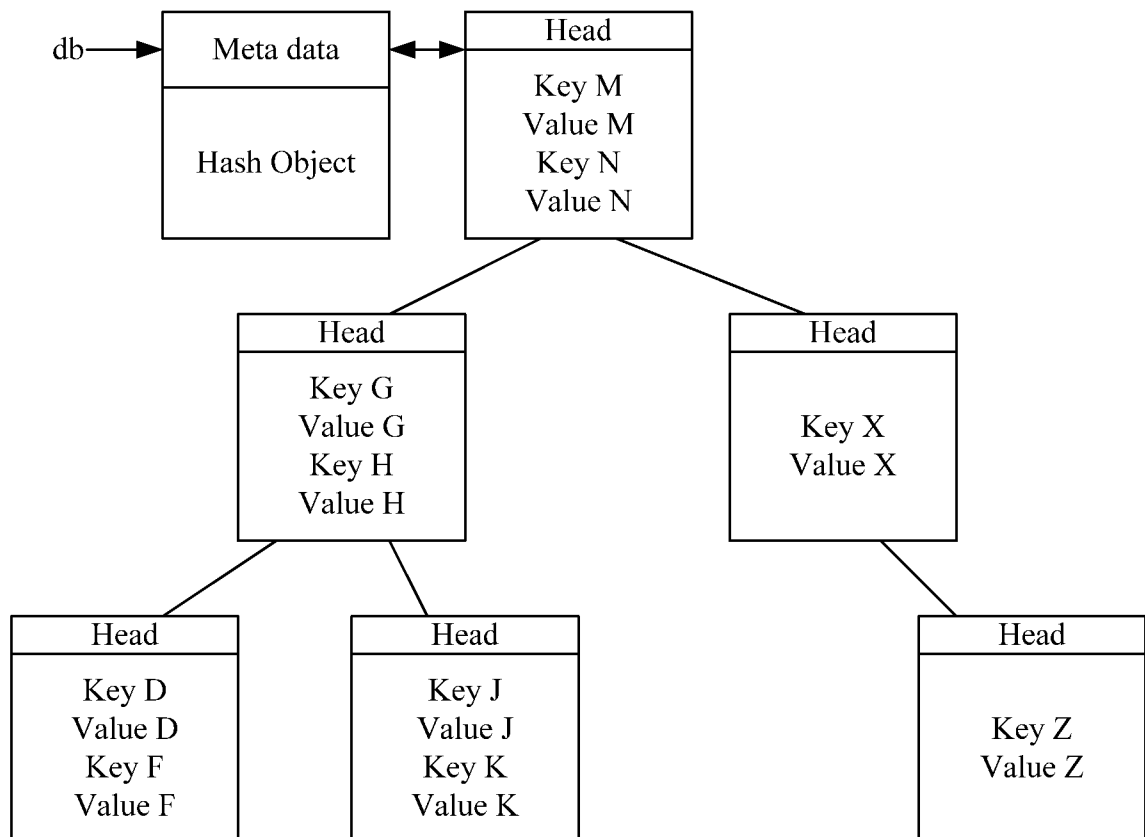


图 8

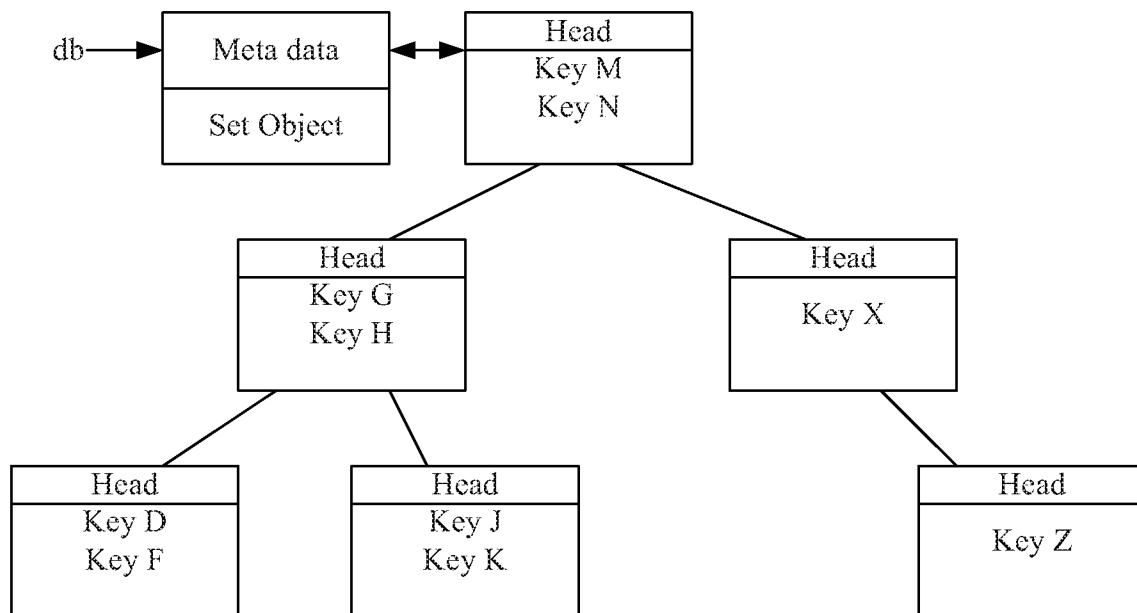


图 9

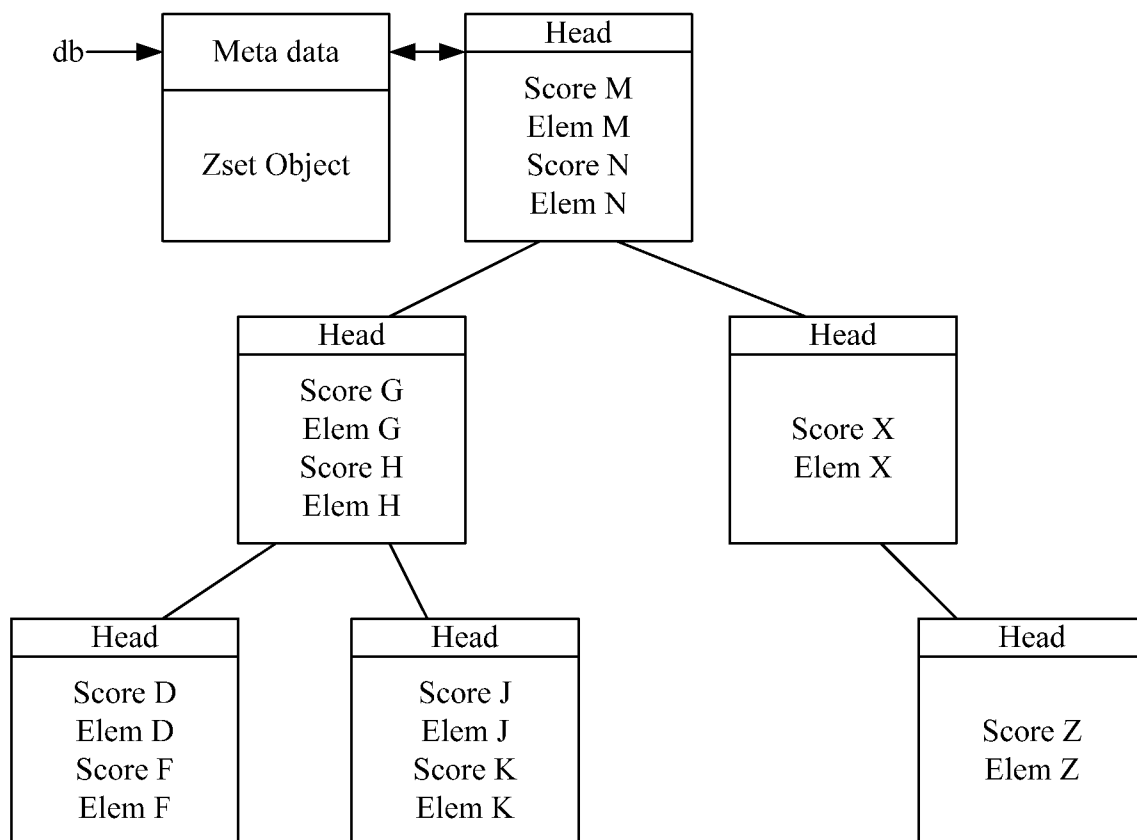


图 10

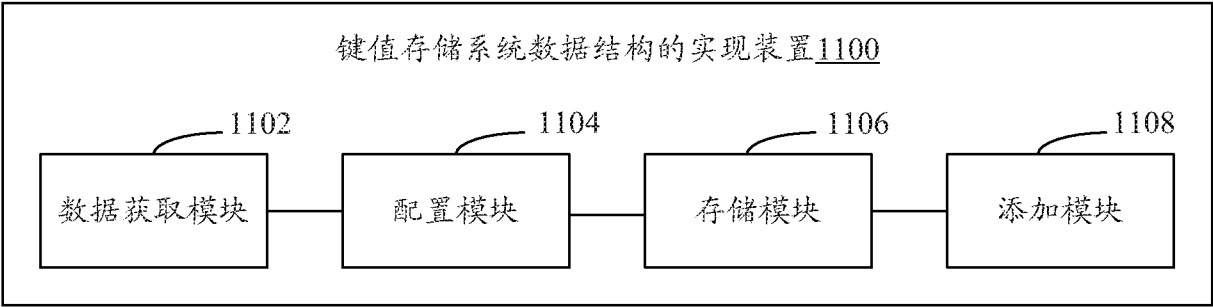


图 11

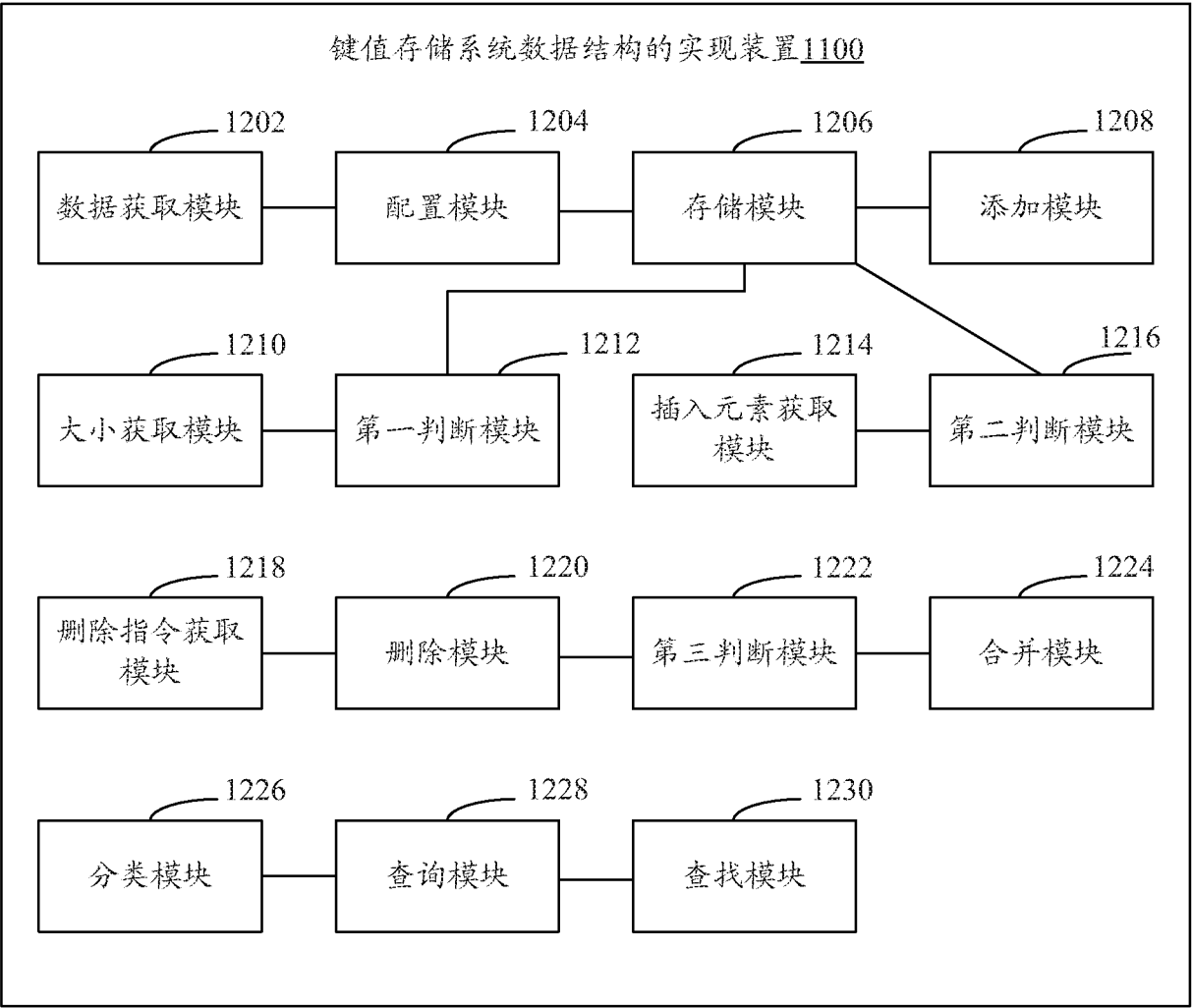


图 12

INTERNATIONAL SEARCH REPORT

International application No.
PCT/CN2017/104515

A. CLASSIFICATION OF SUBJECT MATTER

G06F 17/30 (2006.01) 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: 数据, 存储, 缓存, 插入, 新增, 增加, 数据结构, 类型, 类别, 分类, 区, 块, 大小, 头部, 前, 后, 上, 下, 地址, 编号, 链表, 树, 平衡树, 键值, 内存数据库, 哈希; data, stor+, memory, structure, type, unit, block, key, value, head, chain, tree, B tree, length, pre+, next, hash.

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	CN 103678160 A (TENCENT TECHNOLOGY (SHENZHEN) CO., LTD.), 26 March 2014 (26.03.2014), description, paragraphs [0007]-[0082]	1-20
Y	CN 101276334 A (SHANGHAI XINYUE METER FACTORY), 01 October 2008 (01.10.2008), description, page 2, paragraph 3 to page 4, paragraph 4, and figure 1	1-20
A	CN 103870259 A (TENCENT TECHNOLOGY (SHENZHEN) CO., LTD.), 18 June 2014 (18.06.2014), entire document	1-20
A	WO 2015078194 A1 (HUAWEI TECHNOLOGIES CO., LTD.), 04 June 2015 (04.06.2015), entire document	1-20

☐ Further documents are listed in the continuation of Box C.

☒ See patent family annex.

* Special categories of cited documents:	“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
“A” document defining the general state of the art which is not considered to be of particular relevance	“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
“E” earlier application or patent but published on or after the international filing date	“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
“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)	“&” document member of the same patent family
“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	

Date of the actual completion of the international search 26 December 2017	Date of mailing of the international search report 04 January 2018
Name and mailing address of the ISA State Intellectual Property Office of the P. R. China No. 6, Xitucheng Road, Jimenqiao Haidian District, Beijing 100088, China Facsimile No. (86-10) 62019451	Authorized officer DU, Jingzi Telephone No. (86-10) 62413658

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/CN2017/104515

Patent Documents referred in the Report	Publication Date	Patent Family	Publication Date
CN 103678160 A	26 March 2014	None	
CN 101276334 A	01 October 2008	None	
CN 103870259 A	18 June 2014	None	
WO 2015078194 A1	04 June 2015	CN 103593477 A	19 February 2014
		EP 3037988 A1	29 June 2016
		US 2016217167 A1	28 July 2016

国际检索报告

国际申请号

PCT/CN2017/104515

A. 主题的分类

G06F 17/30(2006.01)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: 数据, 存储, 缓存, 插入, 新增, 增加, 数据结构, 类型, 类别, 分类, 区, 块, 大小, 头部, 前, 后, 上, 下, 地址, 编号, 链表, 树, 平衡树, 键值, 内存数据库, 哈希; data, stor+, memory, structure, type, unit, block, key, value, head, chain, tree, B tree, length, pre+, next, hash.

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
Y	CN 103678160 A (腾讯科技深圳有限公司) 2014年 3月 26日 (2014 - 03 - 26) 说明书第[0007]-[0082]段	1-20
Y	CN 101276334 A (上海新跃仪表厂) 2008年 10月 1日 (2008 - 10 - 01) 说明书第2页第3段-第4页第4段, 图1	1-20
A	CN 103870259 A (腾讯科技深圳有限公司) 2014年 6月 18日 (2014 - 06 - 18) 全文	1-20
A	WO 2015078194 A1 (华为技术有限公司) 2015年 6月 4日 (2015 - 06 - 04) 全文	1-20

☐ 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

2017年 12月 26日

国际检索报告邮寄日期

2018年 1月 4日

ISA/CN的名称和邮寄地址

中华人民共和国国家知识产权局(ISA/CN)
中国北京市海淀区蓟门桥西土城路6号 100088

受权官员

杜婧子

传真号 (86-10)62019451

电话号码 (86-10)62413658

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2017/104515

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	103678160	A	2014年 3月 26日	无			
CN	101276334	A	2008年 10月 1日	无			
CN	103870259	A	2014年 6月 18日	无			
WO	2015078194	A1	2015年 6月 4日	CN	103593477	A	2014年 2月 19日
				EP	3037988	A1	2016年 6月 29日
				US	2016217167	A1	2016年 7月 28日

表 PCT/ISA/210 (同族专利附件) (2009年7月)