



(12)发明专利申请

(10)申请公布号 CN 106294768 A

(43)申请公布日 2017.01.04

(21)申请号 201610658261.1

(22)申请日 2016.08.11

(71)申请人 深圳市宜搜科技发展有限公司
地址 518000 广东省深圳市南山区软件产业基地5栋C座403

(72)发明人 季强 张世侠 张宗世 陈兆卿

(74)专利代理机构 深圳市舜立知识产权代理事务所(普通合伙) 44335

代理人 李亚萍

(51) Int. Cl.
G06F 17/30(2006.01)

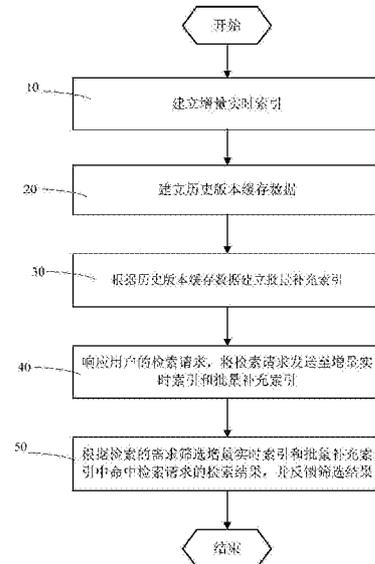
权利要求书2页 说明书6页 附图4页

(54)发明名称

信息搜索方法及信息搜索引擎

(57)摘要

本发明提供一种数据实时更新并且提供历史版本查询的信息搜索方法及信息搜索引擎。所述信息搜索方法及搜索引擎通过建立增量实时索引结合同步有历史版本缓存的批量补充索引的检索架构模式。增量实时索引通过编号递增的数据标码来完成一份数据多个历史版本的存储,在增量实时索引的数据被删除时,批量补充索引中包含的这份数据较早的历史版本,依然能够被检索到,避免数据丢失。



1. 一种信息搜索方法,其特征在于,包括:

建立增量实时索引,所述增量实时索引包括正排索引和倒排索引,其中,正排索引的正排数据包括唯一的用于标识一份数据及其历史版本状态的数据标码和数据属性,倒排索引的倒排数据包括数据标码、检索词及其对应关系;

建立历史版本缓存数据,其中,该历史版本缓存数据存储于独立的存储模块中,每一份历史版本缓存数据包括为一标识的数据键,该数据键对应于该数据的所有历史版本;

根据历史版本缓存数据建立批量补充索引;

响应用户的检索请求,将检索请求发送至增量实时索引和批量补充索引;

根据检索的需求筛选增量实时索引和批量补充索引中命中检索请求的检索结果,并反馈筛选结果。

2. 如权利要求1所述的信息搜索方法,其特征在于:

所述倒排数据采用哈希表附加链表的方式存储;

每一份历史版本缓存数据包括均包含该历史版本缓存数据的所有历史版本的正排数据和倒排数据,其中,每一份历史版本缓存数据中的多个不同历史版本通过版本编号来标识。

3. 如权利要求1所述的信息搜索方法,其特征在于,还包括步骤:在历史版本缓存数据超过存储模块的最大容量时,删除最旧的历史版本缓存数据。

4. 如权利要求1所述的信息搜索方法,其特征在于:所述批量补充索引的换库方式采用双库轮换的方式。

5. 如权利要求1所述的信息搜索方法,其特征在于,还包括步骤:

根据接收到的数据更新内容,将原数据对应的数据标码编号增加一个单位作为更新后的数据的数据标码,分配内存,并复制更新的数据属性,写入正排索引中;

根据接收到的数据更新内容,将该更新的数据的全部检索词逐个添加到倒排索引数据库中;

修改全局数据标码。

6. 如权利要求1所述的信息搜索方法,其特征在于,还包括步骤:

在增量实时索引数据库所存储的数据容量达到内存阈值时,或者数据标码编号超出最大范围时,删除一个或多个子索引数据库。

7. 一种信息搜索引擎,其特征在于,包括:

增量实时索引单元,包括索引数据模块用于建立增量实时索引,所述增量实时索引包括正排索引和倒排索引,其中,正排索引的正排数据包括唯一的用于标识一份数据及其历史版本状态的数据标码和数据属性,倒排索引的倒排数据包括数据标码、检索词及其对应关系;

历史版本缓存单元,用于建立历史版本缓存数据,其中,该历史版本缓存数据存储于一独立的存储模块中,每一份历史版本缓存数据包括为一标识的数据键,该数据键对应于该数据的所有历史版本;

批量补充索引单元,根据历史版本缓存数据建立批量补充索引;

检索单元,响应用户的检索请求,将检索请求发送至增量实时索引和批量补充索引;

结果版本控制单元,根据检索的需求筛选增量实时索引和批量补充索引中命中检索请

求的检索结果,并反馈筛选结果。

8.如权利要求7所述的信息搜索引擎,其特征在于:

所述倒排数据采用哈希表附加链表的方式存储;

每一份历史版本缓存数据包括均包含该历史版本缓存数据的所有历史版本的正排数据和倒排数据,其中,每一份历史版本缓存数据中的多个不同历史版本通过版本编号来标识。

9.如权利要求7所述的信息搜索引擎,其特征在于:所述历史版本缓存单元还用于在历史版本缓存数据超过存储模块的最大容量时,删除最旧的历史版本缓存数据。

10.如权利要求7所述的信息搜索引擎,其特征在于,所述增量实时索引单元还包括:

数据更新模块,用于根据接收到的数据更新内容,将原数据对应的数据标码编号增加一个单位作为更新后的数据的数据标码,分配内存,并复制更新的数据属性,写入正排索引中;根据接收到的数据更新内容,将该更新的数据的全部检索词逐个添加到倒排索引数据库中;以及修改全局数据标码;

索引库容量控制模块,用于在增量实时索引数据库所存储的数据容量达到内存阈值时,或者数据标码编号超出最大范围时,删除一个或多个子索引数据库。

信息搜索方法及信息搜索引擎

技术领域

[0001] 本发明涉及信息搜索领域,特别涉及一种数据实时更新并且提供历史版本查询的信息搜索方法及搜索引擎。

背景技术

[0002] 现有的搜索引擎普遍采用倒排的方式组织索引,倒排索引本质上是面向读取的高效率的信息组织形式。倒排索引通常为了提高存储利用率,采用高效率的压缩形式,从而导致了在更新索引中的数据时较为费时费力,影响到检索效率。

[0003] 在现有技术中,信息搜索领域中更新数据的方式主要有全量磁盘索引和增量内存索引两种。其中,全量磁盘索引容量大,检索效率高,但是索引中的数据无法更新,只能整体替换;增量内存索引容量小,但可以提供新数据的添加和已有数据的变更。

[0004] 现有的搜索引擎应对数据实时更新的普遍做法是采用全量磁盘索引和增量内存索引相结合的方式,来达到数据实时更新和索引容量、检索效率兼顾的目的。全量磁盘索引附加增量内存索引的方式虽然提供了一种解决数据更新的方法,但是对于数据多个历史版本的处理方式存在缺陷,往往只能提供数据的最新版本。在特定应用场景下,数据历史版本的检索也是有意义的,搜索引擎在满足数据实时更新的同时,也需要考虑数据多个历史版本的检索问题。

[0005] 另外,在内存索引结构上,众多现有的实现方法往往只关注存储空间的效率和更新效率,对于检索效率关注不足,数据的更新会排斥检索的并发进行,无法应对搜索引擎相对低频次更新和高频次检索的应用场景。

发明内容

[0006] 有鉴于此,本发明旨在解决相关技术中的上述技术问题,提供一种数据实时更新并且提供历史版本查询的信息搜索方法及信息搜索引擎。

[0007] 一种信息搜索方法,包括:

[0008] 建立增量实时索引,所述增量实时索引包括正排索引和倒排索引,其中,正排索引的正排数据包括唯一的用于标识一份数据及其历史版本状态的数据标码和数据属性,倒排索引的倒排数据包括数据标码、检索词及其对应关系;

[0009] 建立历史版本缓存数据,其中,该历史版本缓存数据存储于独立的存储模块中,每一份历史版本缓存数据包括为一标识的数据键,该数据键对应于该数据的所有历史版本;

[0010] 根据历史版本缓存数据建立批量补充索引;

[0011] 响应用户的检索请求,将检索请求发送至增量实时索引和批量补充索引;

[0012] 根据检索的需求筛选增量实时索引和批量补充索引中命中检索请求的检索结果,并反馈筛选结果。

[0013] 进一步的,所述倒排数据采用哈希表附加链表的方式存储;

[0014] 每一份历史版本缓存数据包括均包含该历史版本缓存数据的所有历史版本的正

排数据和倒排数据,其中,每一份历史版本缓存数据中的多个不同历史版本通过版本编号来标识。

[0015] 进一步的,还包括步骤:在历史版本缓存数据超过存储模块的最大容量时,删除最旧的历史版本缓存数据。

[0016] 进一步的,所述批量补充索引的换库方式采用双库轮换的方式。

[0017] 进一步的,还包括步骤:

[0018] 进一步的,根据接收到的数据更新内容,将原数据对应的数据标码编号增加一个单位作为更新后的数据的数据标码,分配内存,并复制更新的数据属性,写入正排索引中;

[0019] 根据接收到的数据更新内容,将该更新的数据的全部检索词逐个添加到倒排索引数据库中;

[0020] 修改全局数据标码。

[0021] 进一步的,还包括步骤:

[0022] 在增量实时索引数据库所存储的数据容量达到内存阈值时,或者数据标码编号超出最大范围时,删除一个或多个子索引数据库。

[0023] 一种信息搜索引擎,包括:

[0024] 增量实时索引单元,包括索引数据模块用于建立增量实时索引,所述增量实时索引包括正排索引和倒排索引,其中,正排索引的正排数据包括唯一的用于标识一份数据及其历史版本状态的数据标码和数据属性,倒排索引的倒排数据包括数据标码、检索词及其对应关系;

[0025] 历史版本缓存单元,用于建立历史版本缓存数据,其中,该历史版本缓存数据存储于一独立的存储模块中,每一份历史版本缓存数据包括为一标识的数据键,该数据键对应于该数据的所有历史版本;

[0026] 批量补充索引单元,根据历史版本缓存数据建立批量补充索引;

[0027] 检索单元,响应用户的检索请求,将检索请求发送至增量实时索引和批量补充索引;

[0028] 结果版本控制单元,根据检索的需求筛选增量实时索引和批量补充索引中命中检索请求的检索结果,并反馈筛选结果。

[0029] 进一步的,所述倒排数据采用哈希表附加链表的方式存储;

[0030] 每一份历史版本缓存数据包括均包含该历史版本缓存数据的所有历史版本的正排数据和倒排数据,其中,每一份历史版本缓存数据中的多个不同历史版本通过版本编号来标识。

[0031] 进一步的,所述历史版本缓存单元还用于在历史版本缓存数据超过存储模块的最大容量时,删除最旧的历史版本缓存数据。

[0032] 进一步的,所述增量实时索引单元还包括:

[0033] 数据更新模块,用于根据接收到的数据更新内容,将原数据对应的数据标码编号增加一个单位作为更新后的数据的数据标码,分配内存,并复制更新的数据属性,写入正排索引中;根据接收到的数据更新内容,将该更新的数据的全部检索词逐个添加到倒排索引数据库中;以及修改全局数据标码;

[0034] 索引库容量控制模块,用于在增量实时索引数据库所存储的数据容量达到内存阈

值时,或者数据标码编号超出最大范围时,删除一个或多个子索引数据库。

[0035] 在本发明中,搜索引擎采用增量实时索引结合同步有历史版本缓存的批量补充索引的检索架构模式,增量实时索引通过编号递增的数据标码来完成一份数据多个历史版本的存储,通过纯增量的内存组织形式,可以在数据更新的同时,照常提供检索服务,并且不会显降低检索服务的效率。通过将历史版本缓存的全部数据定时同步到批量补充索引中,可以保证增量实时索引的数据被删除时,批量补充索引中包含的这份数据较早的历史版本,依然能够被检索到,避免数据丢失。同时,按照数据的更新频次和规模的实际需要,通过设定增量实时索引和批量补充索引的库容量和同步时间间隔,可以最大限度的保持数据的完整性,提升了搜索引擎的时效等级,满足历史版本检索的应用场景。

附图说明

[0036] 图1是本发明一实施方式中的搜索引擎的模块架构示意图。

[0037] 图2是正排索引的示意图。

[0038] 图3是倒排索引的示意图。

[0039] 图4是历史版本缓存数据的示意图。

[0040] 图5是本发明一实施方式中的信息搜索方法步骤流程图。

[0041] 图6是图5中的信息搜索方法的增量实时索引更新控制方法的步骤流程图。

具体实施方式

[0042] 下面结合附图1~6,详细说明本发明的具体实施方式。

[0043] 请参考图1,在本实施方式中,搜索引擎100包括增量实时索引单元101,历史版本缓存单元102,批量补充索引单元103,检索单元104以及结果版本控制单元105。

[0044] 其中,增量实时索引单元101包括实时索引数据模块111以及数据更新模块114。

[0045] 在本实施方式中,增量实时索引单元101选用全内存的形式,便于数据的实时添加。实时索引数据模块111用于建立增量实时索引,增量实时索引包括正排索引112和倒排索引113。

[0046] 请一并结合图2,所述正排索引112包括存储有正排数据的一个或多个正排索引数据库。正排数据包括数据标码(Data ID)以及数据属性。其中,Data ID是检索的最小单位,每一个Data ID对应唯一的一份数据,该数据可以是一个网页,或者是某种垂直类别的数据,小说、APP应用、影视作品等。Data ID用于标识一份数据及其历史版本状态,同一份数据,内容变更后,添加到正排索引数据库中,会获得一个新的Data ID,新的Data ID中的编号会增加一个单位,用于标识其历史版本状态的变化。其中,数据属性可以是该数据的网页URL、文章的标题、内容等。例如,图2中一网页数据的第一版本网页数据的Data ID为“0”,数据属性为“网页URL1”,当该网页数据更新后,在正排索引112中,更新后的更新版本网页数据的Data ID编号增加一个单位为“1”,数据属性为“网页URL2”。在本发明中,图2仅作为一个简单的示例模型,在其他实施方式中,Data ID作为检索的最小单位,可以包括标识一份数据的编号及其历史版本状态的编号,内容变更后,新的Data ID中用于标识其历史版本状态的编号增加一个单位,以标识其历史版本状态的变化。

[0047] 请一并结合图3,倒排索引113包括存储有倒排数据的一个或多个倒排索引数据

库。倒排数据包括数据标码(Data ID)、检索词及其对应关系。在本实施方式中,倒排数据采用哈希表(Hash table)附加链表的方式存储,例如,hash表中检索词对应的第一个Data ID1的倒排数据中这个Data ID1作为头指针,该倒排数据的最后包含指向下一个倒排数据的Data ID2指针,该是Data ID2尾指针,最后一个倒排数据最后的尾指针填空。

[0048] 数据更新模块114用于在更新数据时,根据接收到的数据更新内容更新正排数据,并将更新后的正排数据写入正排索引112中。其中,在更新正排数据时,首先在不更新全局Data ID的条件下,将原数据对应的Data ID编号增加一个单位作为更新后的数据的Data ID,分配内存,并复制更新的数据属性,写入正排索引112中作为新一条索引。

[0049] 数据更新模块114还用于更新倒排索引,即将更新后的倒排数据写入倒排索引113中。在更新倒排索引113时,将该更新的数据的全部检索词逐个添加到倒排索引数据库中,每添加一个检索词,对应互斥修改hash表和链表的尾指针。全部检索词添加完毕后,修改全局Data ID,使得新添加的数据生效,对检索进程可见。

[0050] 在本实施方式中,所述一个或多个倒排索引数据库以及所述一个或多个正排索引数据库均设置有最大字节容量,按照字节容量预先分配单一区域的大块内存,添加数据时,全部正排数据、倒排数据都从预先分配的单一区域内存中顺序获取,避免长时间使用的内存碎片化。所述一个或多个倒排索引数据库以及所述一个或多个正排索引数据库均设定有单个索引数据库的最大数据容量,也就是Data ID的最大编号范围。正排数据的索引采用数组方式存储,预先分配。

[0051] 在本实施方式中,增量实时索引单元101还包括索引库容量控制模块115。该索引库容量控制模块115用于在整体增量实时索引数据库,即正排索引112以及倒排索引113所存储的数据容量达到内存阈值时,或者Data ID编号超出最大范围时,删除一个或多个子索引数据库。例如,单个索引数据库为100万数据,10个索引数据库支撑1000万的索引总量,当全部索引数据库满时,索引库容量控制模块115清除最旧的索引数据库中的全部数据,库容下降到900万,该被清空的索引数据库用于接收新数据直到再次充满。

[0052] 在本实施方式中,增量实时索引单元101通过编号递增的Data ID来完成一份数据多个历史版本的存储,通过纯增量的内存组织形式,可以在数据更新的同时,照常提供检索服务,并且不会显降低检索服务的效率;通过多个索引数据库循环替换的方式,既满足了过期数据的清除问题,又保持了索引数据库容量的稳定。

[0053] 历史版本缓存单元102用于存储和管理数据的历史版本缓存。历史版本缓存单元102包括独立的存储模块121以及历史版本缓存控制模块122。存储模块121用于存储历史版本缓存数据库,该历史版本缓存数据采用键-值(key-value)对附加链表的方式进行存储和检索,如图4所示,历史版本缓存数据库中每一份历史版本缓存数据包括唯一标识的数据键(Data Key),该Data Key对应有该数据的所有历史版本。每一份历史版本缓存数据包括均包含该历史版本缓存数据的所有历史版本的正排数据和倒排数据,其中,每一份历史版本缓存数据中的多个不同历史版本通过版本编号(Time ver)来标识。数据在添加到增量实时索引单元101的数据库中时,历史版本缓存控制模块122根据接收到的数据更新内容更新历史版本缓存数据库。历史版本缓存控制模块122还用于在历史版本缓存数据库超过存储模块121的最大容量时,删除最旧的历史版本缓存数据。其中,历史版本缓存控制模块122可以使用LRU(Least Recent Used)替换方法删除最旧的历史版本缓存数据。在其他实施方式

中,也可以通过设定单份历史版本缓存数据最多保存的历史版本个数,当历史版本个数超过设定值时,历史版本缓存控制模块122删除最旧的历史版本数据的。

[0054] 批量补充索引单元103用于保存从历史版本缓存单元102同步过来的历史版本缓存数据,并创建新的批量补充索引数据库。批量补充索引数据库的换库方式采用双库轮换的方式,从而始终保持一个索引数据库处于检索状态提供检索服务。具体的,批量补充索引单元103从历史版本缓存单元102接收历史版本缓存数据完成后,创建新的批量补充索引数据库,当新的批量补充索引数据库创建完成后,在此之后的全部检索进程切换到该新的批量补充索引数据库;新的批量补充索引数据库创建完成之前已经发起、未完成的检索进程依然在旧的索引数据库上检索,直到使用旧索引数据库全部检索进程完成后,释放旧的索引数据库,准备下次重建。批量补充索引单元103采用传统的文件索引和批量建库、整库替换的方式,提高库容量和检索效率。

[0055] 通过将历史版本缓存的全部数据定时同步到批量补充索引中,可以保证增量实时索引的数据被删除时,批量补充索引中包含的这份数据较早的历史版本,依然能够被检索到,避免数据丢失。按照数据的更新频次和规模的实际需要,合理的设定增量实时索引和批量补充索引的库容量和同步时间间隔,可以最大限度的保持数据的完整性。

[0056] 检索单元104用于响应用户的检索请求,将检索请求发送至增量实时索引单元101和批量补充索引单元103。

[0057] 结果版本控制单元105用于在检索时,根据检索的需求筛选增量实时索引单元101和批量补充索引单元103中命中检索请求的检索结果。具体的,结果版本控制单元105根据检索需求以及历史版本屏蔽表判断增量实时索引单元101和批量补充索引单元103中命中检索请求的检索结果是否有效。历史版本屏蔽表用于标识增量实时索引单元101和批量补充索引单元103中的数据的历史版本是否有效。具体的,例如,在最常用和最简单的应用场景中,设定最新的版本为有效,历史版本屏蔽表中最新添加的版本为有效,其他历史版本均标识为无效,结果版本控制单元105对比增量实时索引单元101和批量补充索引单元103中命中检索请求的检索结果中每一份历史版本缓存数据的Data Key和版本编号(Time ver)即可判断该历史版本是否有效。

[0058] 请一并参考图5,为本发明一实施方式中的信息搜索方法步骤流程图,根据具体的情况,该流程图步骤的顺序可以改变,某些步骤可以省略。该信息搜索方法包括:

[0059] 步骤10,建立增量实时索引。所述增量实时索引包括正排索引和倒排索引,其中,正排索引的正排数据包括唯一的用于标识一份数据及其历史版本状态的数据标码(Data ID)和数据属性,倒排索引的倒排数据包括数据标码(Data ID)、检索词及其对应关系。

[0060] 步骤20,建立历史版本缓存数据。其中,该历史版本缓存数据存储于独立的存储模块中,每一份历史版本缓存数据包括为一标识的Data Key,该Data Key对应于该数据的所有历史版本。每一份历史版本缓存数据包括均包含该历史版本缓存数据的所有历史版本的正排数据和倒排数据,其中,每一份历史版本缓存数据中的多个不同历史版本通过版本编号(Time ver)来标识。

[0061] 其中,该步骤20还可以包括步骤:在历史版本缓存数据超过存储模块的最大容量时,删除最旧的历史版本缓存数据。

[0062] 步骤30,根据历史版本缓存数据建立批量补充索引。其中,批量补充索引采用传统

的文件索引和批量建库的方式,且换库方式采用双库轮换的方式。

[0063] 步骤40,响应用户的检索请求,将检索请求发送至增量实时索引和批量补充索引。

[0064] 步骤50,根据检索的需求筛选增量实时索引和批量补充索引中命中检索请求的检索结果,并反馈筛选结果。

[0065] 请一并参考图6,为本发明一实施方式中的信息搜索方法的增量实时索引更新控制方法的步骤流程图,根据具体的情况,该流程图步骤的顺序可以改变,某些步骤可以省略。该增量实时索引更新控制方法包括:

[0066] 步骤11,根据接收到的数据更新内容,将原数据对应的Data ID编号增加一个单位作为更新后的数据的Data ID,分配内存,并复制更新的数据属性,写入正排索引中。

[0067] 步骤21,根据接收到的数据更新内容,将该更新的数据的全部检索词逐个添加到倒排索引数据库中。

[0068] 步骤31,修改全局Data ID。

[0069] 步骤41,在增量实时索引数据库所存储的数据容量达到内存阈值时,或者Data ID编号超出最大范围时,删除一个或多个子索引数据库。

[0070] 在本发明的上述实施方式中,搜索引擎采用增量实时索引结合同步有历史版本缓存的批量补充索引的检索架构模式,增量实时索引通过编号递增的Data ID来完成一份数据多个历史版本的存储,通过纯增量的内存组织形式,可以在数据更新的同时,照常提供检索服务,并且不会显降低检索服务的效率。通过将历史版本缓存的全部数据定时同步到批量补充索引中,可以保证增量实时索引的数据被删除时,批量补充索引中包含的这份数据较早的历史版本,依然能够被检索到,避免数据丢失。同时,按照数据的更新频次和规模的实际需要,通过设定增量实时索引和批量补充索引的库容量和同步时间间隔,可以最大限度的保持数据的完整性,提升了搜索引擎的时效等级,满足历史版本检索的应用场景。

[0071] 本技术领域的普通技术人员应当认识到,以上的实施方式仅是用来说明本发明,而并非用作为对本发明的限定,只要在本发明的实质精神范围之内,对以上实施例所作的适当改变和变化都落在本发明要求保护的范围之内。

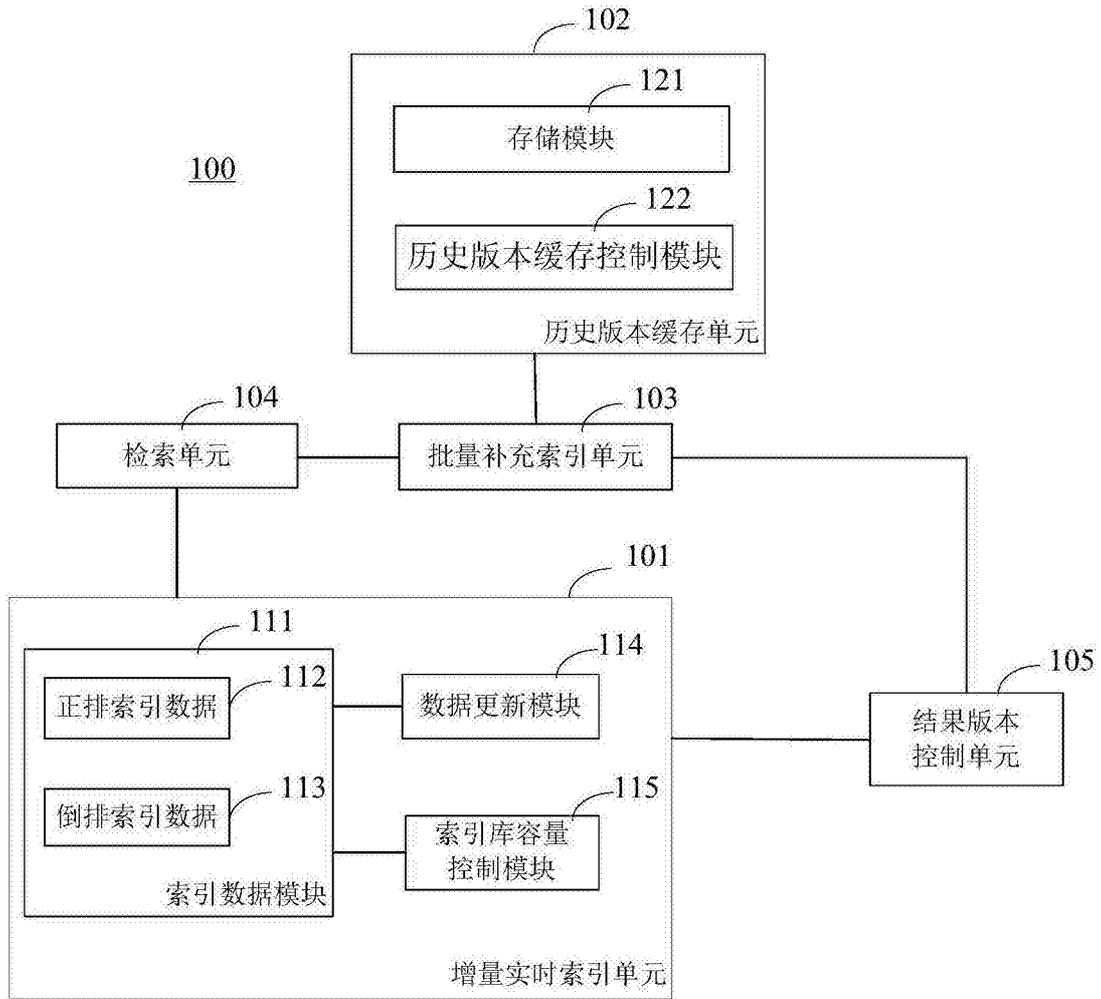


图1

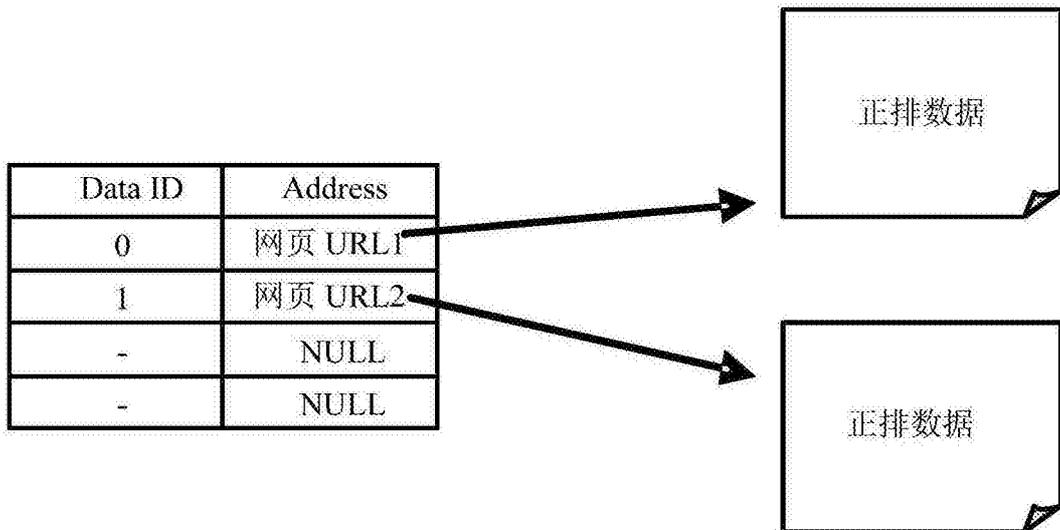


图2

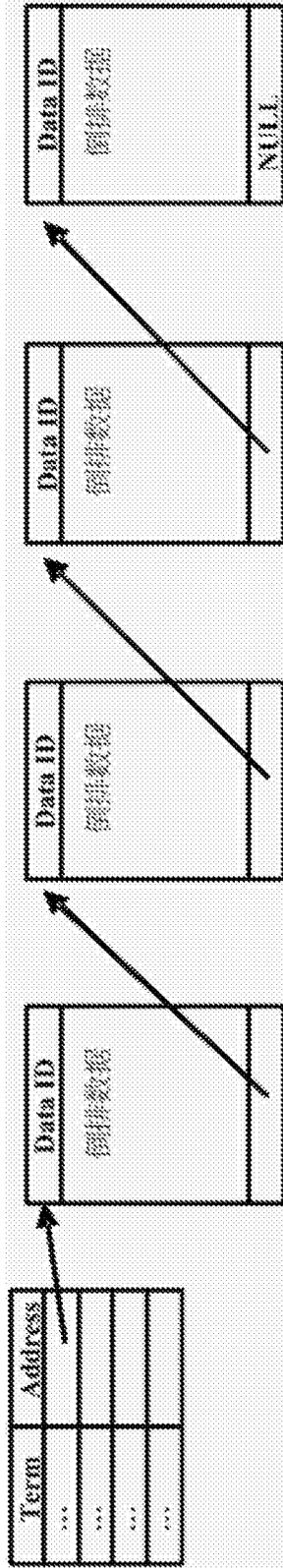


图3

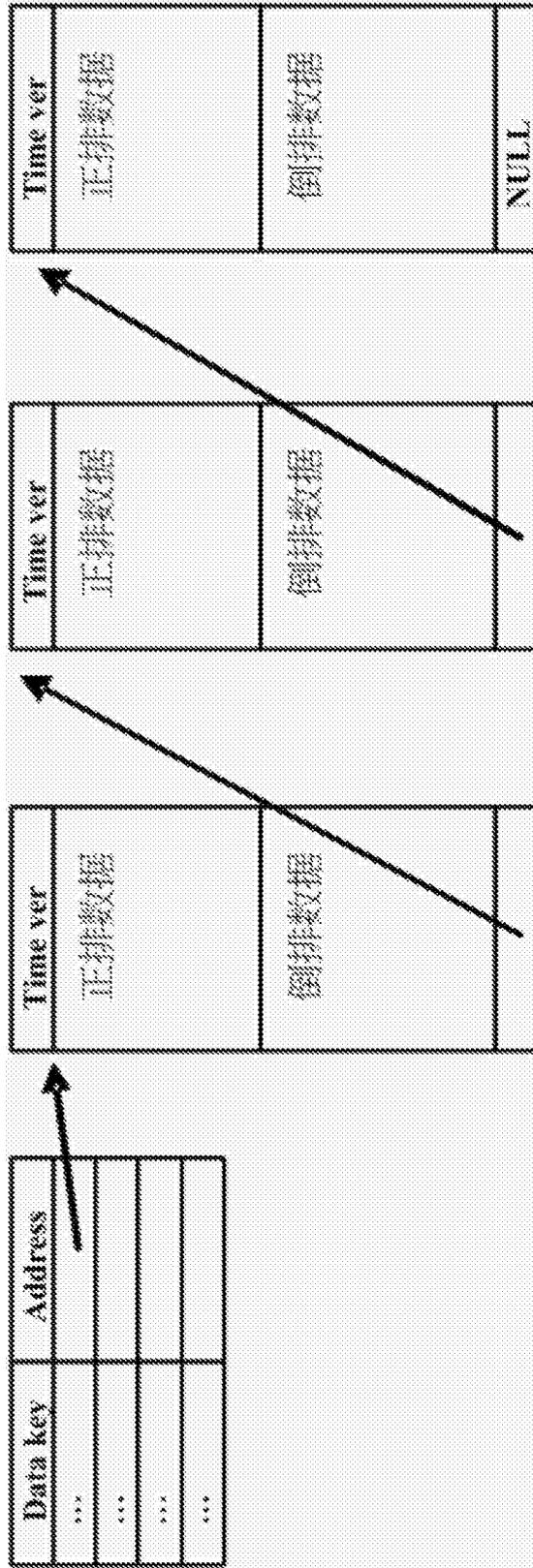


图4

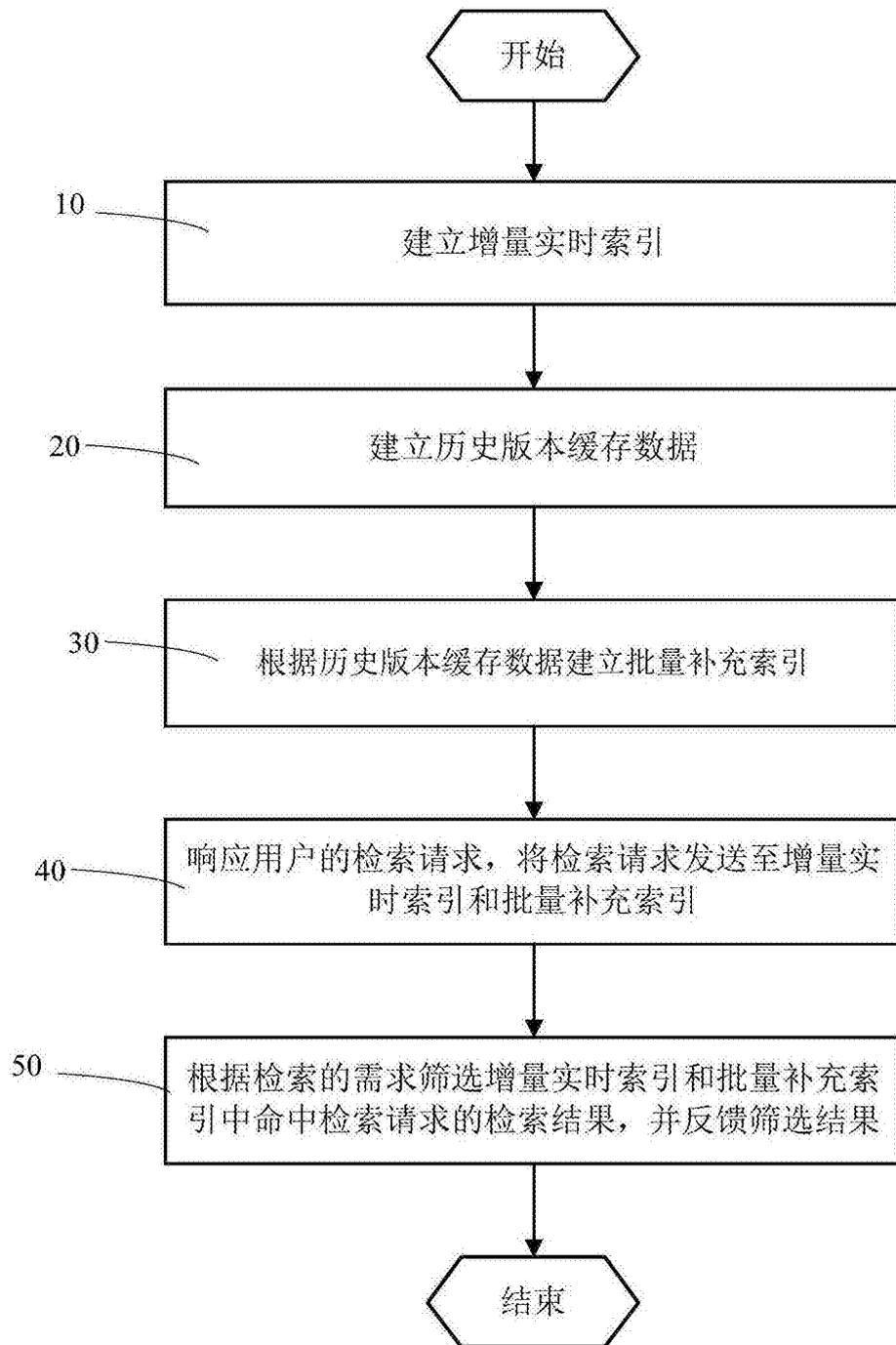


图5

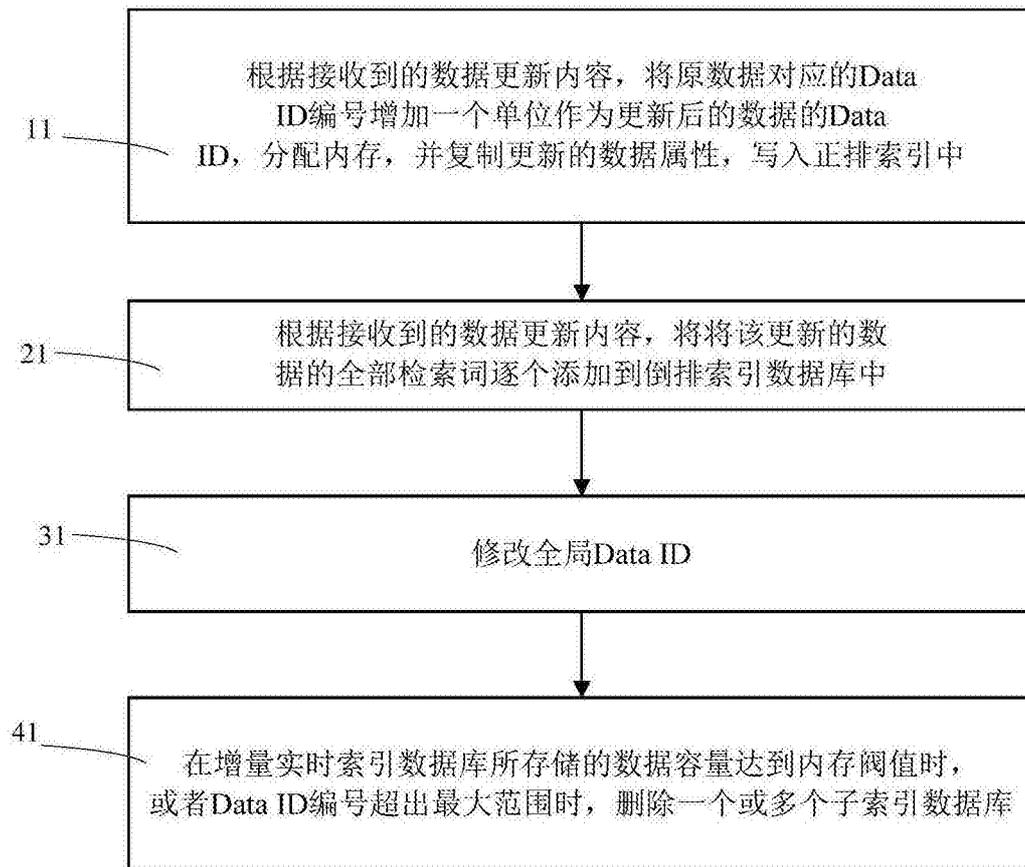


图6