



(12) 发明专利申请

(10) 申请公布号 CN 103473239 A

(43) 申请公布日 2013. 12. 25

(21) 申请号 201210187686. 0

(22) 申请日 2012. 06. 08

(71) 申请人 腾讯科技(深圳)有限公司

地址 518057 广东省深圳市福田区振兴路赛格科技园 2 栋东 403 室

(72) 发明人 钟艳春 刘锦标 江红英

(74) 专利代理机构 上海波拓知识产权代理有限公司 31264

代理人 杨波

(51) Int. Cl.

G06F 17/30(2006. 01)

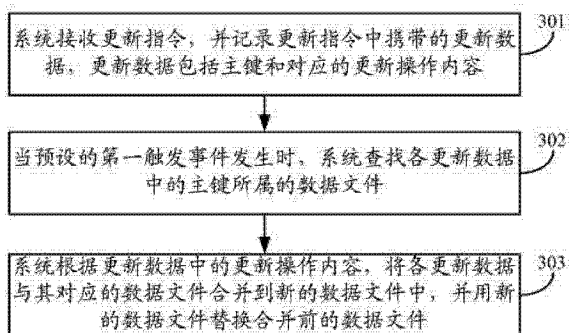
权利要求书3页 说明书7页 附图4页

(54) 发明名称

一种非关系型数据库数据更新方法和装置

(57) 摘要

本发明公开了一种非关系型数据库数据更新方法和装置,该方法包括:接收更新指令,并记录所述更新指令中携带的更新数据,所述更新数据包括主键和对应的更新操作内容;当预设的第一触发事件发生时,查找各更新数据中的主键所属的数据文件;根据更新数据中的所述更新操作内容,将各更新数据与其对应的数据文件合并到新的数据文件中,并用所述新的数据文件替换合并前的数据文件。采用本发明提供的方法,可以提高非关系型数据库历史数据更新的效率。



1. 一种非关系型数据库数据更新方法,其特征在于,包括:

接收更新指令,并记录所述更新指令中携带的更新数据,所述更新数据包括主键和对应的更新操作内容;

当预设的第一触发事件发生时,查找各更新数据中的主键所属的数据文件;

根据更新数据中的所述更新操作内容,将各更新数据与其对应的数据文件合并到新的数据文件中,并用所述新的数据文件替换合并前的数据文件。

2. 如权利要求1所述的方法,其特征在于,预先建立两个用于记录所述更新数据的目录;

将接收到的更新指令中携带的更新数据记录到其中一个目录中;

每当所述第一触发事件发生时,进行目录切换,将后续接收到的更新指令中携带的更新数据记录到另一个目录中,并将之前使用的目录中记录的各更新数据与对应的数据文件合并到新的数据文件中;

将之前使用的目录中记录的各更新数据与对应的数据文件合并到新的数据文件中之后,删除之前使用的目录中的更新数据。

3. 如权利要求1所述的方法,其特征在于,还包括:

当预设的第二触发事件发生时,判断相邻多个数据文件的存储容量之和是否超过预设存储容量阈值,如果没有超过,则将所述多个数据文件合并为一个数据文件。

4. 如权利要求3所述的方法,其特征在于,所述判断相邻多个数据文件的存储容量之和是否超过预设存储容量阈值,如果没有超过,则将所述多个数据文件合并为一个数据文件,具体为:

步骤A,根据数据文件中包含的主键的顺序对数据文件进行排序,将排序后的数据文件中的第一个数据文件作为基准文件;

步骤B,判断基准文件是否为排序后的数据文件中的最后一个数据文件;如果是,则结束流程;否则,确定基准文件与基准文件的下一个数据文件的存储空间之和,并执行步骤C;

步骤C,判断所述存储空间之和是否超过预设的空间阈值;如果是,则将所述基准文件的下一个数据文件作为基准文件,并转至步骤B;否则,将基准文件与基准文件的下一个数据文件合并,将合并后的数据文件作为基准文件,并转至步骤B。

5. 如权利要求3所述的方法,其特征在于,所述预设的第二触发事件,具体为达到预设数据文件合并周期。

6. 如权利要求1所述的方法,其特征在于,还包括:对应各数据文件建立索引文件,索引文件中记录其对应的数据文件中的各 SSTable 所存储的主键的范围;

当接收到查询指令时,确定所述查询指令中携带的主键所属的数据文件;

根据该数据文件对应的索引文件中记录的该数据文件中的各 SSTable 所存储的主键的范围,确定所述查询指令中携带的主键所属的 SSTable;

在该主键所属的 SSTable 中查找该主键对应的值。

7. 如权利要求1所述的方法,其特征在于,所述预设的第一触发事件,具体为达到预设更新数据合并周期。

8. 如权利要求1所述的方法,其特征在于,还包括:

为所述新的数据文件分配新的文件编号；
删除被替换的数据文件，并释放该数据文件的文件编号。

9. 一种非关系型数据库数据更新装置，其特征在于，包括：

记录模块，用于接收更新指令，并记录所述更新指令中携带的更新数据，所述更新数据包括主键和对应的更新操作内容；

查找模块，用于当预设的第一触发事件发生时，查找各更新数据中的主键所属的数据文件；

更新模块，用于根据更新数据中的所述更新操作内容，将各更新数据与其对应的数据文件合并到新的数据文件中，并用所述新的数据文件替换合并前的数据文件。

10. 如权利要求 9 所述的装置，其特征在于，还包括更新数据管理模块，用于：

预先建立两个用于记录所述更新数据的目录；

将接收到的更新指令中携带的更新数据记录到其中一个目录中；

每当所述第一触发事件发生时，进行目录切换，将后续接收到的更新指令中携带的更新数据记录到另一个目录中，并将之前使用的目录中记录的各更新数据与对应的数据文件合并到新的数据文件中；

将之前使用的目录中记录的各更新数据与对应的数据文件合并到新的数据文件中之后，删除之前使用的目录中的更新数据。

11. 如权利要求 9 所述的装置，其特征在于，还包括合并模块，用于：

当预设的第二触发事件发生时，判断相邻多个数据文件的存储容量之和是否超过预设存储容量阈值，如果没有超过，则将所述多个数据文件合并为一个数据文件。

12. 如权利要求 11 所述的装置，其特征在于，所述合并模块，具体用于：

步骤 A，根据数据文件中包含的主键的顺序对数据文件进行排序，将排序后的数据文件中的第一个数据文件作为基准文件；

步骤 B，判断基准文件是否为排序后的数据文件中的最后一个数据文件；如果是，则结束流程；否则，确定基准文件与基准文件的下一个数据文件的存储空间之和，并执行步骤 C；

步骤 C，判断所述存储空间之和是否超过预设的空间阈值；如果是，则将所述基准文件的下一个数据文件作为基准文件，并转至步骤 B；否则，将基准文件与基准文件的下一个数据文件合并，将合并后的数据文件作为基准文件，并转至步骤 B。

13. 如权利要求 11 所述的装置，其特征在于，所述预设的第二触发事件，具体为达到预设数据文件合并周期。

14. 如权利要求 9 所述的装置，其特征在于，还包括：

建立模块，用于对应各数据文件建立索引文件，索引文件中记录其对应的数据文件中的各 SSTable 所存储的主键的范围；

查询模块，用于当接收到查询指令时，确定所述查询指令中携带的主键所属的数据文件；根据该数据文件对应的索引文件中记录的该数据文件中的各 SSTable 所存储的主键的范围，确定所述查询指令中携带的主键所属的 SSTable；在该主键所属的 SSTable 中查找该主键对应的值。

15. 如权利要求 9 所述的装置,其特征在于,所述预设的第一触发事件,具体为达到预设更新数据合并周期。

16. 如权利要求 9 所述的装置,其特征在于,还包括文件编号管理模块,用于:

为所述新的数据文件分配新的文件编号;

删除被替换的数据文件,并释放该数据文件的文件编号。

一种非关系型数据库数据更新方法和装置

技术领域

[0001] 本发明涉及网络技术领域,特别涉及一种非关系型数据库数据更新方法和装置。

背景技术

[0002] 关系型数据库中的表都具有固定的数据结构,每个元组(数据库表中的每一条记录)的字段组成都一样,即使不是每个元组都需要所有的字段,但数据库会为每个元组的所有字段分配相应的数据。例如,某数据库表中包括姓名、年龄、籍贯三个字段,那么该表中的每条记录都需要包括姓名、年龄、籍贯三个字段,即使其中某个元组的某个字段的内容为空,系统也要为其分配一个默认值,如0。这样的结构可以便于表与表之间进行连接等操作,但从另一个角度来说它也是导致关系型数据库性能瓶颈的一个关键因素,使得关系型数据库不能满足对数据库高并发读写的需求,对海量数据的高效率存储和访问的需求,对数据库的高可扩展性和高可用性的需求。

[0003] 为了解决上述问题,非关系数据库(NoSQL)应运而生。Google 的 BigTable 与 Amazon 的 Dynamo 是很成功的商业 NoSQL 系统。一些开源的 NoSQL 系统,如 Membase, MongoDB, Cassandra, BeansDB, Redis 等,也得到了广泛认同。

[0004] 非关系型数据库,它以键值对存储,结构不固定,每一个元组可以有不一样的字段,每个元组可以根据需要增加一些自己的键值对,不局限于固定的结构,可以减少一些时间和空间的开销。KV (Key-Value, 键值) 存储,是 NoSQL 存储的一种方式,数据按照键值对的形式进行组织、索引和存储。KV 存储非常适合不涉及过多数据关系和业务关系的业务数据,同时能有效减少读写磁盘的次数,比关系型数据库存储拥有更好的读写性能。

[0005] 键值存储机制采用键值对形式存储,值可以是任意不定长数据,使用主键可作简单查询和复杂查询(如列表查询、范围查询等),同时定期进行更新数据和历史数据的合并(该合并操作可以包括添加、删除、修改等操作)以处理过期数据。如图1所示,更新数据和历史数据采用0、1目录管理,对于更新数据设置0、1两个目录,对历史数据也设置0、1两个目录,假设当前的更新数据目录和历史数据目录都为0目录,在进行更新数据和历史数据的合并时,将更新数据的0目录和历史数据的0目录合并,并将合并后的数据存储到历史数据的1目录中,同时在合并过程中,新的更新数据写入到更新数据的1目录中。在下一次进行更新数据和历史数据的合并时,将更新数据的1目录和历史数据的1目录合并到历史数据的0目录中,依此类推。

[0006] 然而,发明人发现,现有技术至少存在如下问题:

现有的数据更新方法中,每次更新数据和历史数据的合并,都要将所有的历史数据与更新数据进行合并,即没有发生变化的历史数据也要合并并迁移到另一个目录中,效率较低,大大增加了系统的开销。

发明内容

[0007] 本发明的目的在于提供一种非关系型数据库数据更新方法和装置,以提高非关系

型数据库历史数据更新的效率,为此,本发明实施例采用如下技术方案:

一种非关系型数据库数据更新方法,包括:

接收更新指令,并记录所述更新指令中携带的更新数据,所述更新数据包括主键和对应的更新操作内容;

当预设的第一触发事件发生时,查找各更新数据中的主键所属的数据文件;

根据更新数据中的所述更新操作内容,将各更新数据与其对应的数据文件合并到新的数据文件中,并用所述新的数据文件替换合并前的数据文件。

[0008] 一种非关系型数据库数据更新装置,包括:

记录模块,用于接收更新指令,并记录所述更新指令中携带的更新数据,所述更新数据包括主键和对应的更新操作内容;

查找模块,用于当预设的第一触发事件发生时,查找各更新数据中的主键所属的数据文件;

更新模块,用于根据更新数据中的所述更新操作内容,将各更新数据与其对应的数据文件合并到新的数据文件中,并用所述新的数据文件替换合并前的数据文件。

[0009] 本发明的上述实施例,接收更新指令,并记录更新指令中携带的更新数据,更新数据包括主键和对应的更新操作内容,当预设的第一触发事件发生时,查找记录的各更新数据中的主键所属的数据文件,根据更新数据中的所述更新操作内容,将各更新数据与其对应的数据文件合并到新的数据文件中,并用所述新的数据文件替换合并前的数据文件,从而,可以提高非关系型数据库历史数据更新的效率。

附图说明

[0010] 图 1 为现有技术中更新数据和历史数据的存储方式示意图;

图 2 为本发明实施例提供的非关系型数据库数据更新方法中 SSTable 的结构示意图;

图 3 为本发明实施例提供的非关系型数据库数据更新方法的流程示意图;

图 4 为本发明实施例提供的非关系型数据库数据更新方法中根据更新数据查找对应数据文件的示意图;

图 5 为本发明实施例提供的非关系型数据库数据更新方法中元数据文件存储内容的示意图;

图 6 为本发明实施例提供的非关系型数据库数据更新方法中数据查询方法的流程示意图;

图 7 为本发明实施例提供的非关系型数据库数据更新方法的流程示意图;

图 8 为本发明实施例提供的非关系型数据库数据更新装置的结构示意图。

具体实施方式

[0011] 下面将结合本发明中的附图,对本发明中的技术方案进行清楚、完整的描述,显然,所描述的实施例是本发明的一部分实施例,而不是全部的实施例。基于本发明中的实施例,本领域普通技术人员在没有做出创造性劳动的前提下所获得的所有其它实施例,都属于本发明保护的范围。

[0012] 实施例一

在本实施例中,更新数据以 Memtable 的存储结构存储在更新文件中,历史数据以 SSTable 的存储结构存储在一个或多个数据文件中。SSTable 结构是一种分布式的键值存储结构,每个数据文件中都可以包含多个 SSTable,每个 SSTable 中可以存储有多个 value (主键对应的值)和对应的块内索引,每个 SSTable 的结构可以如图 2 所示。在进行查询的时候,可以先根据 key (主键)查询其所属的数据文件,然后再进一步查询其在该数据文件中所属的 SSTable,由于历史数据的数量非常庞大,采用 SSTable 结构可以提升查询效率。Memtable 是一种非分布式的存储结构,由于更新数据一般数量比较少,所以采用 Memtable 结构可以提升存储效率。

[0013] 如图 3 所示,为本发明实施例提供的非关系型数据库数据更新方法的处理的流程,具体包括以下步骤:

步骤 301,系统接收更新指令,并记录更新指令中携带的更新数据,更新数据包括主键和对应的更新操作内容。

[0014] 数据库的维护人员可以通过服务器或管理终端发出更新指令,系统接收到更新指令后,可以先将更新指令中的更新数据存储到更新文件当中,具体可以采用列表的形式存储 key 和对应的更新操作内容。更新数据中可以包括其需要更新的 key,并包括对相应的 key-value 对进行的操作(即更新操作内容),该更新操作内容可以是删除 key-value 对、添加 key-value 对或修改 value 等。

[0015] 步骤 302,当预设的第一触发事件发生时,系统查找各更新数据中的主键所属的数据文件。

[0016] 具体的,该预设的第一触发事件可以是达到预设更新数据合并周期,即系统预先设置更新数据合并周期,按照周期进行更新数据与历史数据的合并。另外,预设的第一触发事件还可以是接收到更新数据合并指令,即可以由数据库维护人员发送更新数据合并指令,以指示系统将更新数据与历史数据进行合并。

[0017] 具体的,如图 4 所示,系统可以根据元数据文件中记录的元数据,查找各更新数据中的 key 对应的数据文件。元数据是一种用于描述数据的数据,元数据文件中记录的元数据的队列可以如图 5 所示,可以包括每个数据文件的文件编号、文件中 key 的范围、文件的存储容量等信息。另外,在元数据文件中还可以记录文件编号的信息,如已用文件编号、可用文件编号。

[0018] 步骤 303,系统根据更新数据中的更新操作内容,将各更新数据与其对应的数据文件合并到新的数据文件中,并用新的数据文件替换合并前的数据文件。

[0019] 系统针对每条更新数据,根据更新数据中的 key 找到其对应的数据文件,然后将更新数据与相应的数据文件进行合并。具体的,合并的操作是根据更新数据中的更新操作内容进行的,可以包括删除 key-value 对、添加 key-value 对或修改 value 等。

[0020] 当多个更新数据 key 对应到同一个数据文件时,此多个更新数据共同与该数据文件合并到新的数据文件中。合并过程会根据原数据文件中的各 key 和 value,以及更新数据中的 key 和对应的更新操作内容,按照 key 的顺序,在新的数据文件中重新顺序生成各 SSTable, SSTable 的存储容量为预先设置的数值。

[0021] 优选的,系统可以为合并得到的新的数据文件分配新的文件编号(在可用文件编号中选取)。在合并结束后,系统可以删除被替换的数据文件,并释放该数据文件的文件编

号,该文件编号由已用文件编号变为可用文件编号。

[0022] 在本实施例上述流程中,系统可以预先建立两个用于记录更新数据的目录,将接收到的更新指令中携带的更新数据记录到其中一个目录中;然后,每当第一触发事件发生时,进行目录切换,将后续接收到的更新指令中携带的更新数据记录到另一个目录中,并将之前使用的目录中记录的各更新数据与对应的数据文件合并到新的数据文件中;将之前使用的目录中记录的各更新数据与对应的数据文件合并到新的数据文件中之后,删除之前使用的目录中的更新数据。

[0023] 具体的,可以采用 0、1 目录的方式对更新数据进行进行存储,分别建立更新数据的 0 目录文件、1 目录文件,每当第一触发事件发生时,进行目录切换,假设当前使用的是 0 目录,当第一触发事件发生开始进行合并过程时,新接收到的更新数据将被存储到目录 1 中,将目录 0 中的更新数据与相应的数据文件进行合并,且在合并过程结束后,0 目录中的更新数据将被删除,然后,下一次发生第一触发事件时,再切换回到 0 目录进行更新数据存储,将目录 1 中的更新数据与相应的数据文件进行合并,且在合并过程结束后,删除 1 目录中的更新数据,依此类推。

[0024] 优选的,本发明实施例还提供了的非关系型数据库数据更新方法,还可以包括:当预设的第二触发事件发生时,判断相邻多个数据文件的存储容量之和是否超过预设存储容量阈值,如果没有超过,则将所述多个数据文件合并为一个数据文件。其中的第二触发事件可以是达到预设数据文件合并周期,也可以是各更新数据与相应的数据文件合并完毕,还可以是接收到数据文件合并指令。具体的数据文件的合并过程可以包括如下步骤:

步骤 A,根据数据文件中包含的主键的顺序对数据文件进行排序,将排序后的数据文件中的第一个数据文件作为基准文件。

[0025] 根据数据文件中存储的 key 的顺序,可以对数据文件进行排序。例如,数据文件 1 中包含 key1、key2、key3,数据文件 2 中包含 key4、key5、key6,数据文件 3 中包含 key7、key8、key9,则根据 key 的顺序可以确定数据文件的顺序为数据文件 1、数据文件 2、数据文件 3。

[0026] 步骤 B,判断基准文件是否为排序后的数据文件中的最后一个数据文件;如果是,则结束流程;否则,确定基准文件与基准文件的下一个数据文件的存储空间之和,并执行步骤 C。

[0027] 步骤 C,判断所述存储空间之和是否超过预设的空间阈值;如果是,则将所述基准文件的下一个数据文件作为基准文件,并转至步骤 B;否则,将基准文件与基准文件的下一个数据文件合并,将合并后的数据文件作为基准文件,并转至步骤 B。

[0028] 例如,对数据文件 1 到数据文件 n 进行数据文件合并过程,设置存储容量阈值为 1M (兆)。将数据文件 1 作为基准文件,判断基准文件与数据文件 2 的存储容量之和是否超过 1M,假设没有超过 1M,则将数据文件 1 与数据文件 2 合并为数据文件 1',将数据文件 1' 作为基准文件,并继续判断基准文件与数据文件 3 的存储容量之和是否超过 1M,假设超过 1M,则将数据文件 3 作为基准文件,判断基准文件与数据文件 4 的存储容量之和是否超过 1M,依此类推,直至基准文件是数据文件队列中的最后一个文件时,结束流程。

[0029] 本发明实施例还提供了一种数据查询的方法,系统预先对应各数据文件建立索引文件,索引文件中记录其对应的数据文件中的各 SSTable 所存储的主键的范围。具体的,索

引文件中可以记录 SSTable 中的开始 key, SSTable 中 key 的个数, SSTable 数据在其所属的数据文件中的偏移。在数据库启动时,系统可以预先将各索引文件的内容加载到内存中。

[0030] 如图 6 所示,该数据查询方法的流程可以包括如下步骤:

步骤 601,当接收到查询指令时,系统确定所述查询指令中携带的主键所属的数据文件。系统可以根据元数据文件中记录的元数据,查找各更新数据中的 key 对应的数据文件,具体方法在步骤 302 中已作阐述,在此不再累述。

[0031] 步骤 602,系统根据该数据文件对应的索引文件中记录的该数据文件中的各 SSTable 所存储的主键的范围,确定查询指令中携带的主键所属的 SSTable。

[0032] 在数据库启动时,系统可以预先将各索引文件的内容加载到内存中。然后,系统可以根据内存中加载的数据文件中各 SSTable 所存储的主键的范围,确定相应的主键所属的 SSTable。

[0033] 步骤 603,在该主键所属的 SSTable 中查找该主键对应的值。

[0034] 本发明实施例中,接收更新指令,并记录更新指令中携带的更新数据,更新数据包括主键和对应的更新操作内容,当预设的第一触发事件发生时,查找记录的各更新数据中的主键所属的数据文件,根据更新数据中的所述更新操作内容,将各更新数据与其对应的数据文件合并到新的数据文件中,并用所述新的数据文件替换合并前的数据文件,从而,可以提高非关系型数据库历史数据更新的效率。

[0035] 实施例二

如图 7 所示,为本发明实施例提供的非关系型数据库数据更新方法在具体应用场景中的处理流程,可以包括如下步骤:

步骤 701,系统切换到更新数据合并模式,获取当前数据正在使用的目录。假设当前使用的目录是 0 目录,后续再接收到的更新数据将存入到 1 目录中。

[0036] 步骤 702,获取该目录中所有更新数据对应的 key,并根据元数据查找各 key 对应的数据文件。

[0037] 步骤 703,分配可用文件编号,建立新的数据文件,并将各更新数据与其对应的数据文件合并到新的数据文件中。合并结束后,可以删除合并前的数据文件及其对应的索引文件,并将其文件编号释放为可用。

[0038] 步骤 704,建立新的数据文件的索引文件,并将该索引文件的内容更新到内存中。可以用该索引文件的内容替换合并前的数据文件对应的索引文件在内存中的相应内容。

[0039] 步骤 705,结束更新数据与历史数据的合并流程,系统切换到正常模式。在正常模式下,系统只接收更新数据并存储到相应的目录中,而不进行更新数据与历史数据的合并。

[0040] 本发明实施例中,接收更新指令,并记录更新指令中携带的更新数据,更新数据包括主键和对应的更新操作内容,当预设的第一触发事件发生时,查找记录的各更新数据中的主键所属的数据文件,根据更新数据中的所述更新操作内容,将各更新数据与其对应的数据文件合并到新的数据文件中,并用所述新的数据文件替换合并前的数据文件,从而,可以提高非关系型数据库历史数据更新的效率。

[0041] 实施例三

基于相同的技术构思,本发明实施例还提供了一种非关系型数据库数据更新装置,如图 8 所示,包括:

记录模块 810,用于接收更新指令,并记录所述更新指令中携带的更新数据,所述更新数据包括主键和对应的更新操作内容;

查找模块 820,用于当预设的第一触发事件发生时,查找各更新数据中的主键所属的数据文件;

更新模块 830,用于根据更新数据中的所述更新操作内容,将各更新数据与其对应的数据文件合并到新的数据文件中,并用所述新的数据文件替换合并前的数据文件。

[0042] 优选的,还包括更新数据管理模块,用于:

预先建立两个用于记录所述更新数据的目录;

将接收到的更新指令中携带的更新数据记录到其中一个目录中;

每当所述第一触发事件发生时,进行目录切换,将后续接收到的更新指令中携带的更新数据记录到另一个目录中,并将之前使用的目录中记录的各更新数据与对应的数据文件合并到新的数据文件中;

将之前使用的目录中记录的各更新数据与对应的数据文件合并到新的数据文件中之后,删除之前使用的目录中的更新数据。

[0043] 优选的,还包括合并模块,用于:

当预设的第二触发事件发生时,判断相邻多个数据文件的存储容量之和是否超过预设存储容量阈值,如果没有超过,则将所述多个数据文件合并为一个数据文件。

[0044] 优选的,所述合并模块,具体用于:

步骤 A,根据数据文件中包含的主键的顺序对数据文件进行排序,将排序后的数据文件中的第一个数据文件作为基准文件;

步骤 B,判断基准文件是否为排序后的数据文件中的最后一个数据文件;如果是,则结束流程;否则,确定基准文件与基准文件的下一个数据文件的存储空间之和,并执行步骤 C;

步骤 C,判断所述存储空间之和是否超过预设的空间阈值;如果是,则将所述基准文件的下一个数据文件作为基准文件,并转至步骤 B;否则,将基准文件与基准文件的下一个数据文件合并,将合并后的数据文件作为基准文件,并转至步骤 B。

[0045] 优选的,所述预设的第二触发事件,具体为达到预设数据文件合并周期。

[0046] 优选的,还包括:

建立模块,用于对应各数据文件建立索引文件,索引文件中记录其对应的数据文件中的各 SSTable 所存储的主键的范围;

查询模块,用于当接收到查询指令时,确定所述查询指令中携带的主键所属的数据文件;根据该数据文件对应的索引文件中记录的该数据文件中的各 SSTable 所存储的主键的范围,确定所述查询指令中携带的主键所属的 SSTable;在该主键所属的 SSTable 中查找该主键对应的值。

[0047] 优选的,所述预设的第一触发事件,具体为达到预设更新数据合并周期。

[0048] 优选的,还包括文件编号管理模块,用于:

为所述新的数据文件分配新的文件编号;

删除被替换的数据文件,并释放该数据文件的文件编号。

[0049] 本发明实施例中,接收更新指令,并记录更新指令中携带的更新数据,更新数据包

括主键和对应的更新操作内容,当预设的第一触发事件发生时,查找记录的各更新数据中的主键所属的数据文件,根据更新数据中的所述更新操作内容,将各更新数据与其对应的数据文件合并到新的数据文件中,并用所述新的数据文件替换合并前的数据文件,从而,可以提高非关系型数据库历史数据更新的效率。

[0050] 本领域技术人员可以理解实施例中的装置中的模块可以按照实施例描述进行分布于实施例的装置中,也可以进行相应变化位于不同于本实施例的一个或多个装置中。上述实施例的模块可以合并为一个模块,也可以进一步拆分成多个子模块。

[0051] 上述本发明实施例序号仅仅为了描述,不代表实施例的优劣。

[0052] 通过以上的实施方式的描述,本领域的技术人员可以清楚地了解到本发明可借助软件加必需的通用硬件平台的方式来实现,当然也可以通过硬件,但很多情况下前者是更佳的实施方式。基于这样的理解,本发明的技术方案本质上或者说对现有技术做出贡献的部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,包括若干指令用以使得一台终端设备(可以是手机,个人计算机,服务器,或者网络设备等)执行本发明各个实施例所述的方法。

[0053] 以上所述仅是本发明的优选实施方式,应当指出,对于本技术领域的普通技术人员来说,在不脱离本发明原理的前提下,还可以做出若干改进和润饰,这些改进和润饰也应视本发明的保护范围。

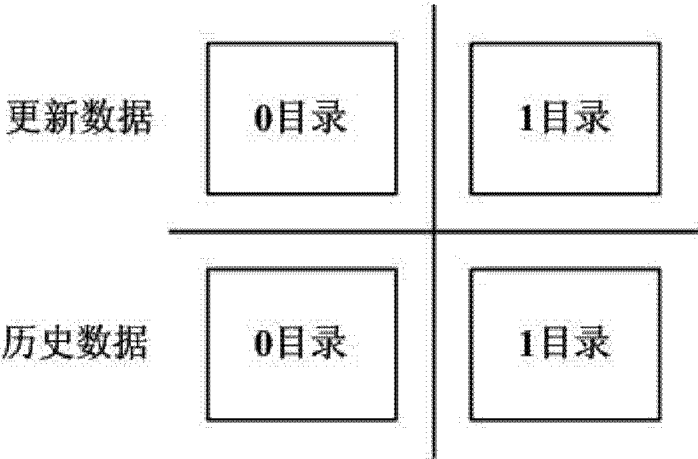


图 1



图 2

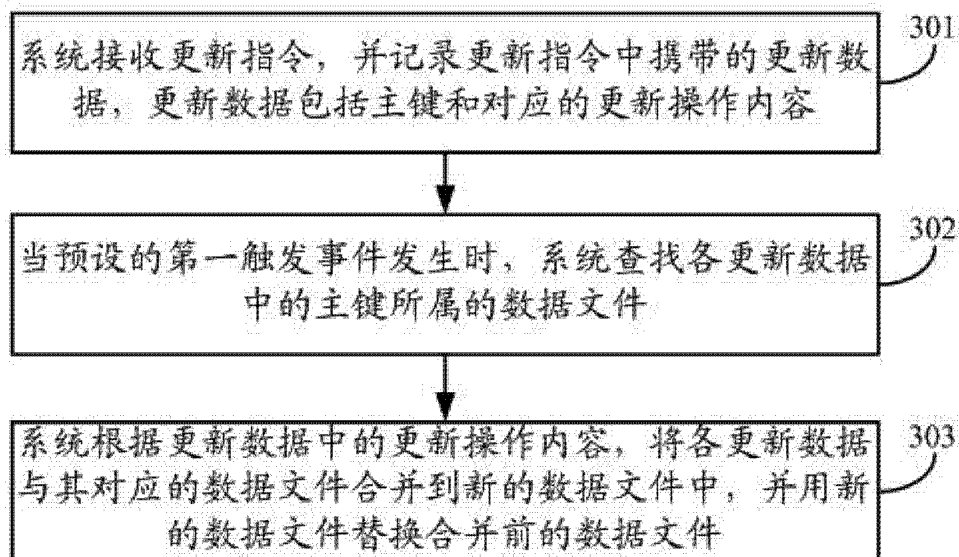


图 3

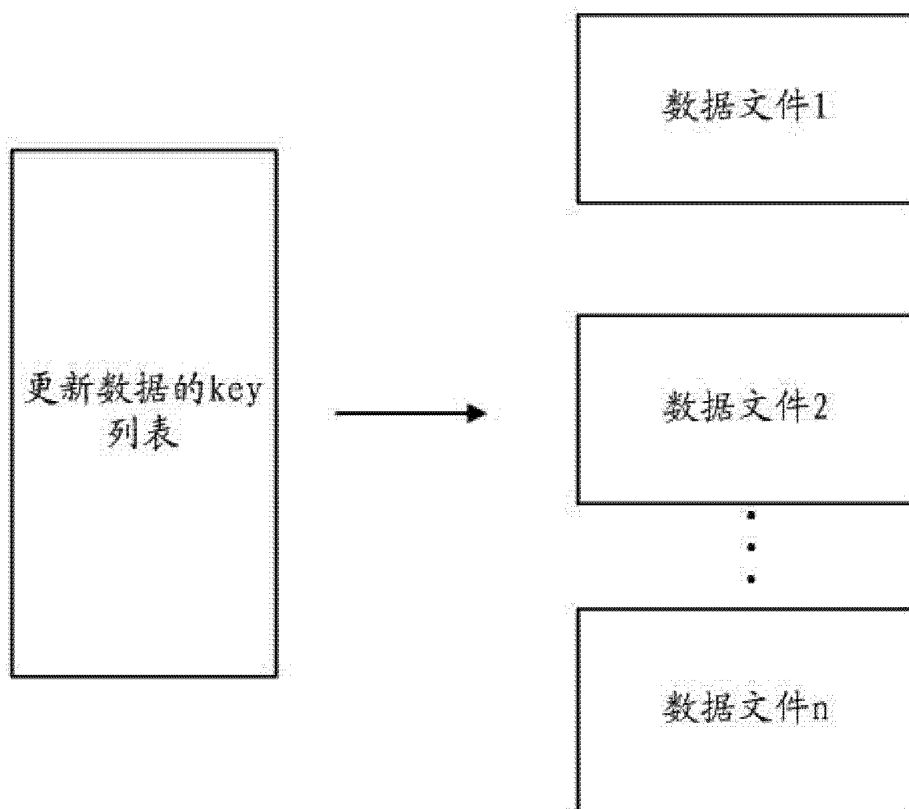


图 4

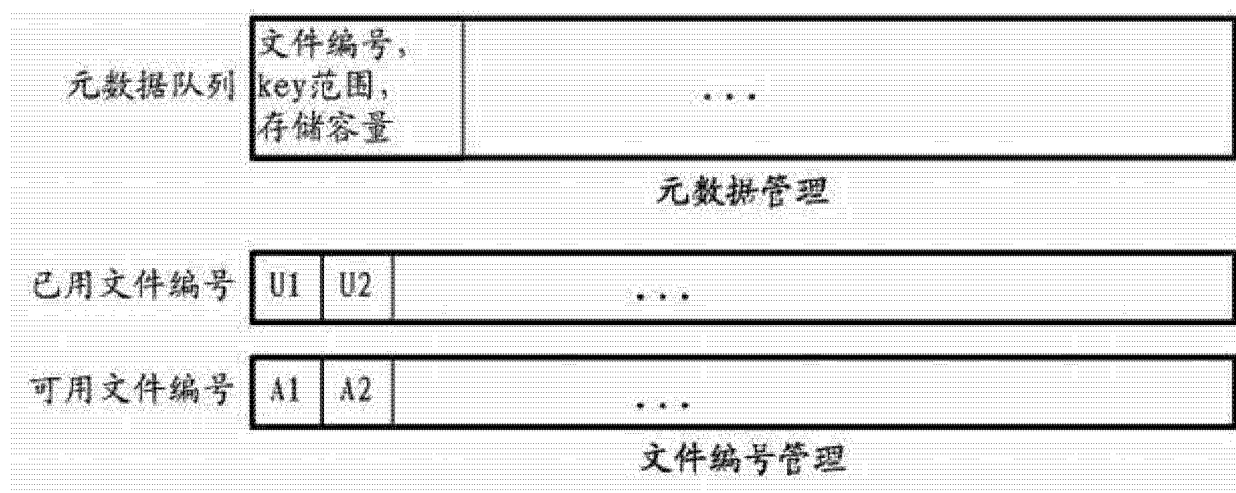


图 5

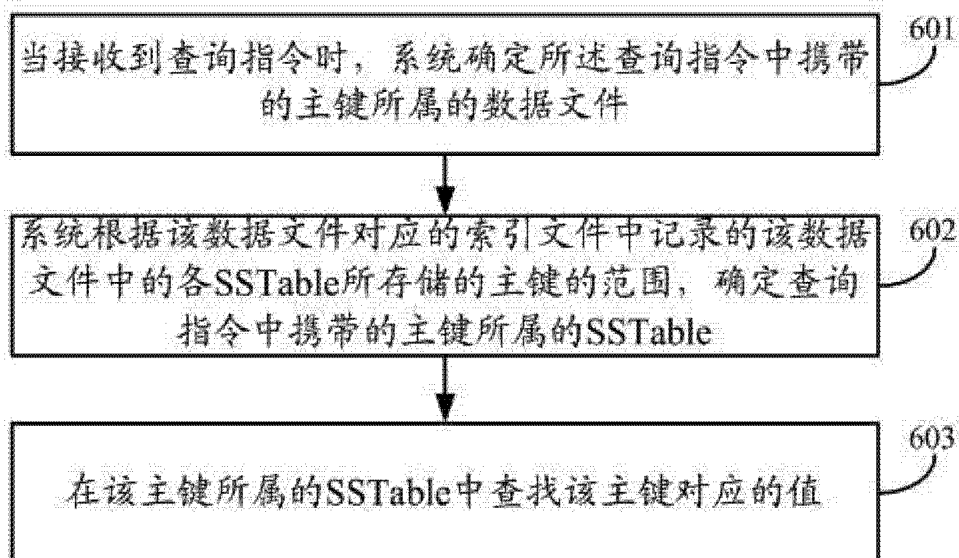


图 6

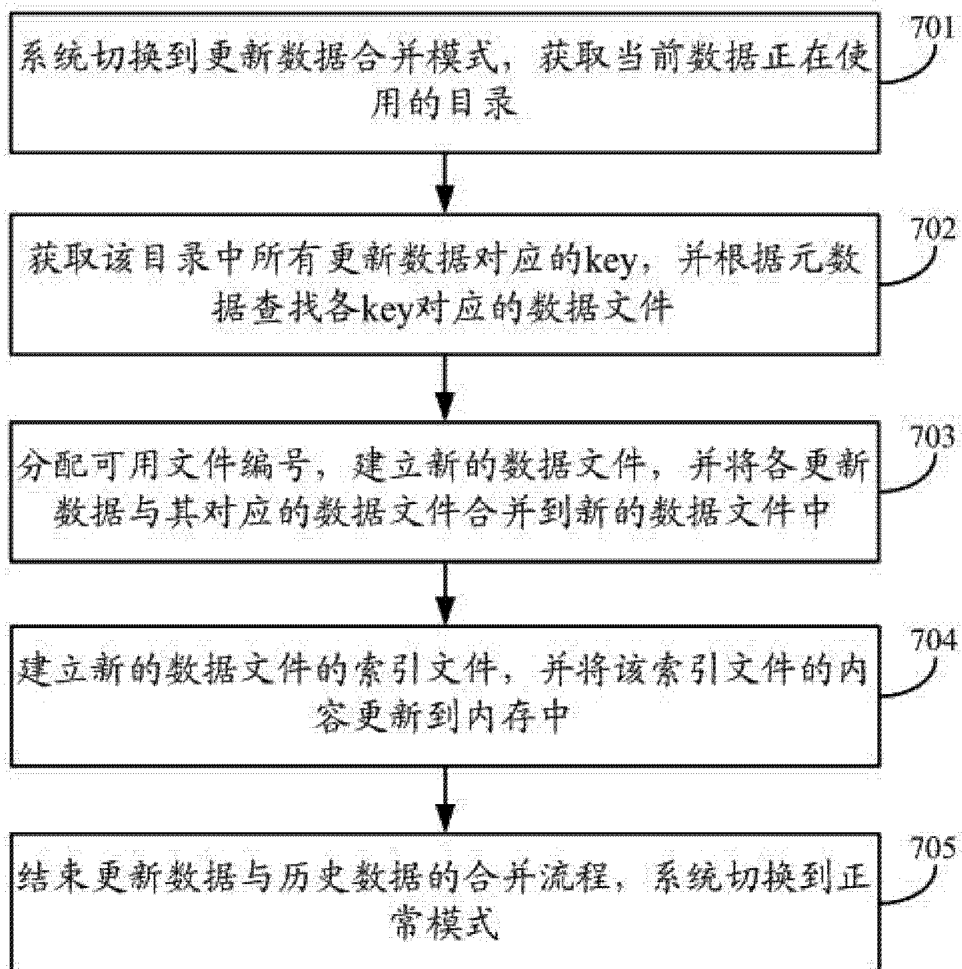


图 7

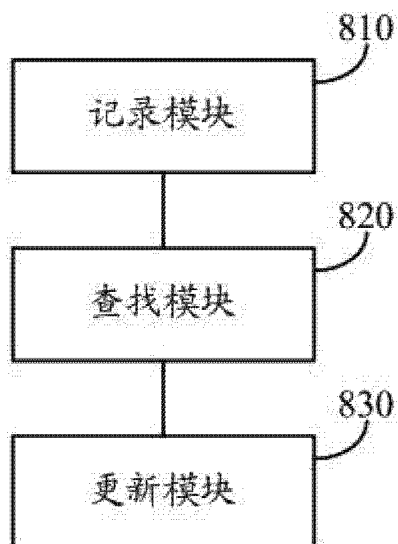


图 8