



## (12) 发明专利申请

(10) 申请公布号 CN 105574093 A

(43) 申请公布日 2016. 05. 11

(21) 申请号 201510918956. 4

(22) 申请日 2015. 12. 10

(71) 申请人 深圳市华讯方舟软件技术有限公司

地址 518102 广东省深圳市宝安区西乡街道  
宝田一路臣田工业区第 37 栋 3 楼

申请人 深圳市华讯方舟科技有限公司

(72) 发明人 张鋆 冯骏

(74) 专利代理机构 北京律和信知识产权代理事  
务所(普通合伙) 11446

代理人 刘国伟 武玉琴

(51) Int. Cl.

G06F 17/30(2006. 01)

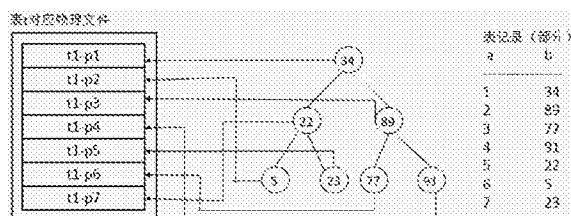
权利要求书1页 说明书6页 附图2页

## (54) 发明名称

一种在基于 HDFS 的 spark-sql 大数据处理系  
统上建立索引的方法

## (57) 摘要

本发明公开了一种在基于 HDFS 的 spark-sql 大数据处理系统上建立索引的方法,通过 SQL 语句在基于 HDFS 的 spark-sql 大数据处理系统上增加索引,删除索引,插入数据,删除数据,在数据查询的时候,自动判断查询列是否存在索引,如果存在,则查找索引包含的文件块,过滤不需要查询的文件块。本发明对在给 spark-sql 增加索引功能后,能有效增加查询速度,例如一个典型的 spark-sql 数据表,大小为 1000GB,1GB 一个文件存放,分为 1000 个文件,如果查询单条记录,原先做法需要扫描 1000 个文件,建立索引后,只需要扫描 1 个文件即可,效率提高 1000 倍。按照一般情况估算,结合传统的关系型数据库经验,建立索引的 spark-sql 数据库比没有索引的 sql 语句查询速度执行要快 100-10000 倍或更多。



1. 一种在基于HDFS的spark-sql大数据处理系统上建立索引的方法,其特征在于:通过SQL语句在基于HDFS的spark-sql大数据处理系统上增加索引,删除索引,插入数据,删除数据,在数据查询的时候,自动判断查询列是否存在索引,如果存在,则查找索引包含的文件块,过滤不需要查询的文件块。

2. 根据权利要求1所述在基于HDFS的spark-sql大数据处理系统上建立索引的方法,其特征在于:增加索引时,首先需要新增一个索引文件,索引文件的格式可以根据配置和其它指令设置,常用的有B-树、Hash索引等格式,然后遍历原有表中所有记录,确定每条记录所需要索引的列的值位于HDFS或其他文件系统中位置,再记录该记录的列值和对应的文件信息,写入索引树结构,循环遍历所有记录,以文件形式保存索引结构,最后更新表元数据信息,将新的索引信息写入表的元数据中,以备后续查询使用。

3. 根据权利要求1所述在基于HDFS的spark-sql大数据处理系统上建立索引的方法,其特征在于:删除某表某列的索引时,只需要定位到相应的索引文件将其删除,并更新表元数据信息,同时删除元数据中的索引信息即可。

4. 根据权利要求1所述在基于HDFS的spark-sql大数据处理系统上建立索引的方法,其特征在于:插入一条数据后,判断插入的数据是否涉及到索引,如果涉及索引,则需要调整相应的索引结构,将这条数据和它相关联的文件信息也加入到索引结构中去。

5. 根据权利要求1所述在基于HDFS的spark-sql大数据处理系统上建立索引的方法,其特征在于:在整个流程中,数据表增加数据流程沿用原流程不变,只在数据增加完成后,记录数据所在的文件名,根据此返回的文件名构造索引节点。

6. 根据权利要求1所述在基于HDFS的spark-sql大数据处理系统上建立索引的方法,其特征在于:删除一条数据后,判断删除的数据是否涉及到索引,如果涉及索引,则需要调整相应的索引结构,将这条数据关联的索引节点删除。

7. 根据权利要求1所述在基于HDFS的spark-sql大数据处理系统上建立索引的方法,其特征在于:在整个流程中,其中数据表删除数据流程沿用原流程不变,只在数据删除完成后,增加删除数据对应的索引信息。

8. 根据权利要求1所述在基于HDFS的spark-sql大数据处理系统上建立索引的方法,其特征在于:查询数据时,根据数据列值,查询索引文件中相应的节点元素,然后读取元素值,从而获取该数据所对应文件所在的文件名,然后继续原查询流程,原查询流程最后将读取这个表中所有的数据文件进行查询,在这之前根据上一步获取的文件名,过滤掉无效文件,然后对剩下的文件继续执行查询流程,然后根据查询的数据执行SQL操作,最后返回查询结果。

9. 根据权利要求1所述在基于HDFS的spark-sql大数据处理系统上建立索引的方法,其特征在于:将一条记录中某个字段的索引定位到某个文件,即记录了这条记录包含在哪个文件中,后续查找这条记录时,只需要根据索引直接定位到某个文件,而不用扫描这个表所包含的所有文件。

## 一种在基于HDFS的spark-sql大数据处理系统上建立索引的方法

### 技术领域

[0001] 本发明涉及一种在数据处理系统上建立索引的方法,尤其涉及一种在基于HDFS的spark-sql大数据处理系统上建立索引的方法。

### 背景技术

[0002] Spark-sql是在大数据处理平台spark的基础上,增加了支持标准sql查询语句的功能。

[0003] Spark大数据处理平台是Berkeley所开源的类Hadoop MapReduce的通用并行框架,它拥有Hadoop MapReduce所具有的优点;但是Spark大数据处理程序的编写需要掌握scala语言,并且基于开放的API函数接口进行代码编写,繁琐而且复杂,并且大量的传统数据库开发人员所掌握的SQL语言无法在spark上使用。spark-sql的诞生解决了上述问题,它将传统的数据库表概念应用到spark处理框架,用户可以像操作传统数据库表那样,用sql语句建表和查询,spark-sql自动将相应操作转化为spark内部操作,屏蔽了复杂的处理细节。

[0004] 但是由于spark大数据处理平台的特殊性,spark-sql不支持在数据表上建立索引,即不支持类似于传统数据库的建立索引语句,例如:

[0005] `create index myindex on table t(b);`

[0006] 意为:在表t的b列上建立名为myindex的一般索引。

[0007] 传统关系型数据库在接收到上述命令后,即开始为a表的c列建立索引。

[0008] 索引的类型有很多种,例如B-树索引,Hash索引,GiST索引,GIN索引等等。以B-树索引为例,关系型数据库建立索引原理如下:

[0009] 数据库开辟一块单独的存储区域,用来存储索引树。

[0010] 根据需要索引的列(例子中为名为b的列)中的字段生成B-树。并将此树保存到指定存储区域。其中B-树的每个节点对应b列中每个元素,另外每个节点中还包含一个指针,该指针记录了这个节点对应元素保存在数据库文件中的相应位置。

[0011] 当b列插入新元素时,也要将新元素插入B-树(B-树会自动调整),同时该树节点记录该元素在数据库文件中的位置。

[0012] 当b列删除元素时,也要将元素从B-树中删除(B-树会自动调整)。

[0013] 数据库索引基于数据库存储的实体文件,即上面所说的“数据库文件中的相应位置”,数据库文件可以根据需要自定义格式,所以数据在文件中的相应位置可以有不同表示方法,但总体思路都是记录了一个元素在文件中的确切位置,后续查找该元素时,不需要遍历文件,而可以借助这个已经记录的位置通过某种方法快速定位记录,从而达到加快查找的目的。

[0014] 如图1所示,数据文件对应表t,有a,b两列,其中b列建立了一个索引。即右边树结构的索引,索引中每个节点中的元素有指向数据文件相应元素位置的指针,其中索引本身

也作为一个文件存储起来。

[0015] 当查询数据时,例如查询语句

[0016] `Select*from t where b=22;`

[0017] 表示查询表t中b列等于22的所有行。首先数据库先解析sql语句,然后发现b列存在索引。

[0018] 然后,直接从索引树中快速找到22元素,根据22元素的指针,定位到元素值为22的b列所在的行,其地址为0x90,然后直接根据地址,取出这一行,返回结果为“5 22”。

[0019] 当表t插入元素时,根据b列的值,会动态修改相应的索引树;相应的,当表t删除元素时,也会动态删减索引树中的内容。

[0020] 传统关系数据库的索引技术已经比较成熟,这里表述的是其一般原理,其实现方式多种多样,表现形式通常不固定,但基本原理都是相通的,其它索引的建立方法和实现步骤这里不再一一赘述。

[0021] 对于spark-sql,由于其底层的文件存储方式不同于传统关系型数据库(通常采用HDFS,而不是一般的Linux或Windows文件系统),而且通常一个表容量很大,一个表甚至会关联到成千上万个物理文件,所以spark-sql设计之时并没有创建索引的功能。其设计重点在于强调数据处理的并发能力而忽略了处理的效率问题。

[0022] 通常spark-sql进行数据查询时,会搜索整个数据表,通常spark-sql处理的数据量很大,一个数据库可能对应于多个物理文件,spark-sql会通过并发技术,对所有文件进行搜索。

[0023] 如图2所示,以同样的sql查询语句为例:

[0024] `Select*from t where b=22;`

[0025] 首先Spark-sql解析sql语句,然后定位到表t的数据文件(这里有4个文件)。然后将这几个文件根据特定大小拆分成多个块,分配给不同的工作进程处理,工作进程顺序扫描整个表包含的所有文件分块,找到b列值为22的行,找到之后,返回结果。

[0026] 可以看出,spark-sql在没有索引的情况下,进行表搜索的方式比较简单,存在效率低下,需要扫描全部的数据文件的所有行。

[0027] 除了简单的select语句,传统关系数据库在所有涉及到查询的地方包括复杂查询,子查询,嵌套查询等等都会应用索引技术,来减少查询量,加快查询速度,spark-sql没有这样的机制。

[0028] 综上所述,当前的spark-sql由于没有数据索引机制,无法使得查询速度达到最优,相比于传统关系型数据库,存在效率低下的问题。

## 发明内容

[0029] 本发明要解决的技术问题是提供一种在基于HDFS的spark-sql大数据处理系统上建立索引的方法,该方法可以使spark-sql适应于更多、更灵活的应用场景,加快spark-sql执行sql语句进行查询的速度,提高spark-sql的执行效率,更充分地发挥spark-sql处理大数据能力的优势。

[0030] 为了解决上述技术问题,本发明提供了一种在基于HDFS的spark-sql大数据处理系统上建立索引的方法,通过SQL语句在基于HDFS的spark-sql大数据处理系统上增加索

引,删除索引,插入数据,删除数据,在数据查询的时候,自动判断查询列是否存在索引,如果存在,则查找索引包含的文件块,过滤不需要查询的文件块。

[0031] 增加索引时,首先需要新增一个索引文件,索引文件的格式可以根据配置和其它指令设置,常用的有B-树、Hash索引等格式,然后遍历原有表中所有记录,确定每条记录所需要索引的列的值位于HDFS或其他文件系统中位置,再记录该记录的列值和对应的文件信息,写入索引树结构。循环遍历所有记录,以文件形式保存索引结构,最后更新表元数据信息,将新的索引信息写入表的元数据中,以备后续查询使用。

[0032] 删除某表某列的索引时,只需要定位到相应的索引文件将其删除,并更新表元数据信息,同时删除元数据中的索引信息即可。

[0033] 插入一条数据后,判断插入的数据是否涉及到索引,如果涉及索引,则需要调整相应的索引结构,将这条数据和它相关联的文件信息也加入到索引结构中去。

[0034] 在整个流程中,数据表增加数据流程沿用原流程不变,只在数据增加完成后,记录数据所在的文件名,根据此返回的文件名构造索引节点。

[0035] 删除一条数据后,判断删除的数据是否涉及到索引,如果涉及索引,则需要调整相应的索引结构,将这条数据关联的索引节点删除。

[0036] 在整个流程中,其中数据表删除数据流程沿用原流程不变,只在数据删除完成后,增加删除数据对应的索引信息

[0037] 查询数据时,根据数据列值,查询索引文件中相应的节点元素,然后读取元素值,从而获取该数据所对应文件所在的文件名,然后继续原查询流程,原查询流程最后将读取这个表中所有的数据文件进行查询,在这之前根据上一步获取的文件名,过滤掉无效文件,然后对剩下的文件继续执行查询流程,然后根据查询的数据执行SQL操作,最后返回查询结果。

[0038] 将一条记录中某个字段的索引定位到某个文件,即记录了这条记录包含在哪个文件中,后续查找这条记录时,只需要根据索引直接定位到某个文件,而不用扫描这个表所包含的所有文件。

[0039] 本发明在基于HDFS的spark-sql大数据处理系统上建立索引的方法与现有技术相比具有以下有益效果。

[0040] 对spark-sql增加索引功能后,能有效增加查询速度,例如一个典型的spark-sql数据表,大小为1000GB,1GB一个文件存放,分为1000个文件,如果查询单条记录,原先做法需要扫描1000个文件,建立索引后,只需要扫描1个文件即可,效率提高1000倍。按照一般情况估算,结合传统的关系型数据库经验,建立索引的spark-sql数据库比没有索引的sql语句查询速度执行要快100-10000倍或更多。

## 附图说明

[0041] 以下结合附图和具体实施方式对本发明在基于HDFS的spark-sql大数据处理系统上建立索引的方法作进一步的详细描述。

[0042] 图1是现有技术中普通数据表与索引树结构示意图。

[0043] 图2是现有技术中在HDFS的spark-sql大数据处理系统上没有索引时的查询原理图。

- [0044] 图3是本发明的增加索引流程图。
- [0045] 图4是本发明的删除索引流程图。
- [0046] 图5是本发明的增加数据流程图。
- [0047] 图6是本发明的删除数据流程图。
- [0048] 图7是本发明的查询数据流程图。
- [0049] 图8是HDF分布式存储系统结构示意图。
- [0050] 图9是本发明分布式存储系统中数据表与索引树结构示意图。

### 具体实施方式

[0051] 如图3至图9所示,本实施方式在基于HDFS的spark-sql大数据处理系统上建立索引的方法实现了spark-sql增加支持索引功能,能够像传统关系型数据库那样,通过SQL语句增加索引,删除索引,插入数据,删除数据,在数据查询的时候,自动判断查询列是否存在索引,如果存在,则查找索引包含的文件块,过滤不需要查询的文件块,达到加快查询速度的目的。

[0052] 1)如图3所示,增加索引流程。

[0053] 增加索引指在原有数据表的基础上,为某一列增加索引,后续针对此列的查询可以通过索引加速。

[0054] 增加索引时,首先需要新增一个索引文件,索引文件的格式可以根据配置和其它指令设置,常用的有B-树、Hash索引等格式,然后遍历原有表中所有记录,确定每条记录所需要索引的列的值位于HDFS或其他文件系统中位置,再记录该记录的列值和对应的文件信息,写入索引树结构。循环遍历所有记录,以文件形式保存索引结构,最后更新表元数据信息,将新的索引信息写入表的元数据中,以备后续查询使用。

[0055] 2)如图4所示,删除索引流程。

[0056] 删除某表某列的索引流程较为简单,只需要定位到相应的索引文件将其删除,并更新表元数据信息,同时删除元数据中的索引信息即可。

[0057] 3)如图5所示,插入数据流程(表已有索引)。

[0058] 插入一条数据后(包括批量插入,实际为连续单条数据插入),会判断插入的数据是否涉及到索引,如果涉及索引,则需要调整相应的索引结构,将这条数据和它相关联的文件信息也加入到索引结构中去。

[0059] 在整个流程中,其中数据表增加数据流程沿用原流程不变,只在数据增加完成后,记录数据所在的文件名,根据此返回的文件名构造索引节点。

[0060] 4)如图6所示,删除数据流程(表已有索引)。

[0061] 删除一条数据后(包括批量删除,实际为连续单条数据删除),会判断删除的数据是否涉及到索引,如果涉及索引,则需要调整相应的索引结构,将这条数据关联的索引节点删除。

[0062] 在整个流程中,其中数据表删除数据流程沿用原流程不变,只在数据删除完成后,增加删除数据对应的索引信息。

[0063] 5)如图7所示,查询数据流程(表有索引)。

[0064] 查询数据时,根据数据列值,查询索引文件中相应的节点元素,然后读取元素值,

从而获取该数据所对应文件所在的文件名,然后继续原查询流程,原查询流程最后将读取这个表中所有的数据文件进行查询,在这之前根据上一步获取的文件名,过滤掉无效文件,然后对剩下的文件继续执行查询流程。经过过滤的文件数量会大大减少,减少查询负担,然后根据查询的数据执行SQL操作,最后返回查询结果。

[0065] 在此特别强调的是,本实施方式基于spark-sql的索引,不同于传统数据库的索引,其设计目的是为了处理大数据量的。传统数据库容量以10GB为例,spark-sql可以做到1PB,即10万倍普通传统数据库容量。

[0066] 普通数据库一张数据表一般对应若干个文件系统上的物理文件,而spark-sql典型部署方式与HDFS相结合,是以一种分布式存储的方式来存储文件,其一张数据表可以对应于数千个乃至上万个存储在HDFS上的文件,如图8所示。

[0067] 通常一个spark-sql节点由若干个spark节点组成,其底层存储采用 HDFS分布式存储系统。即数据文件存在于HDFS。图中,t1-p1表示表t1的part1部分,它是一个物理文件,同理t1-p2表示表t1的part2文件,整个表t1由p1-p7共7个文件组成;同样地,表t2由3个文件组成。

[0068] 原查询流程在进行sql语句查询时,会扫描所有表文件。

[0069] 例如Select\*from t where b=22

[0070] Spark-sql解析上述sql语句,然后查找表t对应的数据库文件,结果为t1-p1,t1-p2,t1-p3,t1-p4,t1-p5,t1-p6,t1-p7一共7个文件,在不考虑文件过大切分的情况下,spark-sql会建立7个查询任务,分别对应这7个文件开始扫描查询,扫描所有文件,直到找到符合条件的记录行。

[0071] 本发明参考一般索引的原理并在此基础上针对spark-sql存储特点加以改进。

[0072] 本发明的索引粒度不同于传统数据库索引,传统数据库索引一般指向某条记录在文件内的地址,由于spark-sql一个数据库表文件通常由很多文件组成,所以本法采取的思路为,将一条记录中某个字段的索引定位到某个文件,即记录了这条记录包含在哪个文件中,后续查找这条记录时,只需要根据索引直接定位到某个文件,而不用扫描这个表所包含的所有文件。

[0073] 引用以上述例子,表t有t1-p1,t1-p2,t1-p3,t1-p4,t1-p5,t1-p6,t1-p7一共7个文件组成,表t有2个字段a和b,其中b字段建立了索引,假设其中有若干条记录(这里没有显示全部记录),建立如图9所示的索引。

[0074] 其中表记录为插入到表中的原始记录,插入的同时,在b列上建立B-树索引。索引树中每个节点记录了该节点对应的数据库记录中的值以及该记录所在HDFS文件系统中对应的物理文件。

[0075] 当查询数据时,例如查询语句。

[0076] Select\*from t where b=22;

[0077] 表示查询表t中b列等于22的所有行。首先数据库先解析sql语句,然后发现b列存在索引。

[0078] 然后直接从索引树中快速找到22元素,根据22元素的指针,定位到元素值为22的b列所在的物理文件为t1-p7,然后仅读取这一个文件内容进行查找,找到记录后返回。

[0079] 当表t插入元素时,根据b列的值,会动态修改相应的索引树;对应的,当表t删除

元素时,也会动态删减索引树中的内容。

[0080] 可以看出,本发明中spark-sql的索引概念与传统数据库虽然类似,但有着根本的不同,本发明是根据spark-sql处理大数据特点,将索引粒度从传统数据库的文件中某位置改为spark数据库中某个文件,从而避免扫描大量无效文件,避免浪费系统资源。

[0081] 本发明中的索引适用于所有sql语句,即在无论简单或者复杂sql查询中,但凡涉及到索引列的查询操作,都会先根据索引定位文件,然后在定位的文件中进行sql查询操作,这与传统关系数据库做法具有根本区别。

[0082] 本发明的关键点

[0083] 1、在spark-sql上增加支持索引的机制,例如支持以下sql语句:

[0084] 建立索引:create index myindex on t(b);其中关键字为create index on

[0085] 查看索引:show index from t;其中关键字为show index from

[0086] 删除索引:drop index myindex on t;其中关键字为drop index on

[0087] 2、基于文件的索引机制

[0088] Spark-sql与传统关系型数据库不同,本发明的关键点之一在于,索引建立在文件基础上,即索引指向一个HDFS或者其它文件系统上的一个具体文件,而不是文件中的内容,粒度要比传统数据库大。在数据库表根据本发明建立索引的前提下,能够有效过滤无效查询文件,能够缩小所查询的文件范围,从而提高查询效率。

[0089] 3、建立的索引包含但不限于唯一索引,主键索引,多属性索引,部分索引,表达式索引。这些索引类型与传统数据库中的概念一致;建立索引所使用的数据结构包含但不限于B-树,Hash,GiST,GIN等,这些数据结构与传统数据库中的概念一致。

[0090] 本发明的优点如下。

[0091] 目前没有公开的对spark-sql支持索引技术的实时方案和方法。

[0092] 所以目前公开技术中,spark-sql中建立的数据库表都是没有索引的,其查询速度和查询效率被限制,通过对spark-sql建立索引机制,可以提高查询速度若干数量级。可以做到海量数据情况下,查询效率和查询速度与传统关系型数据库不相上下。

[0093] 需要说明的是,以上参照附图所描述的各个实施例仅用以说明本发明 而非限制本发明的范围,本领域的普通技术人员应当理解,在不脱离本发明的精神和范围的前提下对本发明进行的修改或者等同替换,均应涵盖在本发明的范围之内。此外,除上下文另有所指外,以单数形式出现的词包括复数形式,反之亦然。另外,除非特别说明,那么任何实施例的全部或一部分可结合任何其它实施例的全部或一部分来使用。



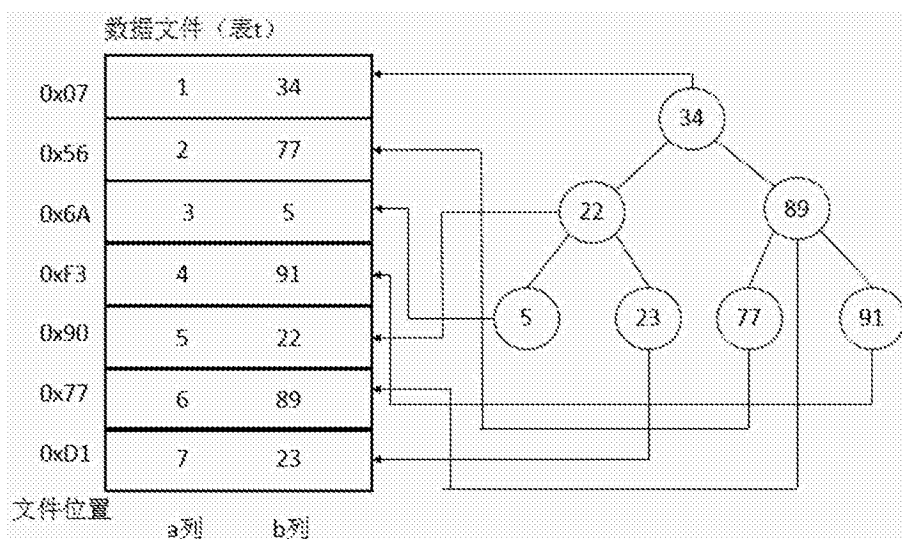


图1

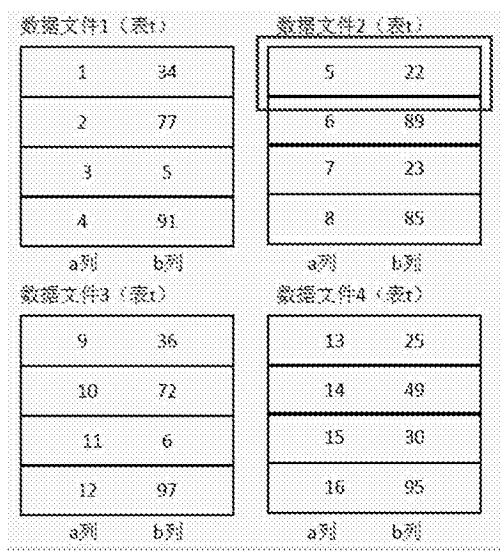


图2

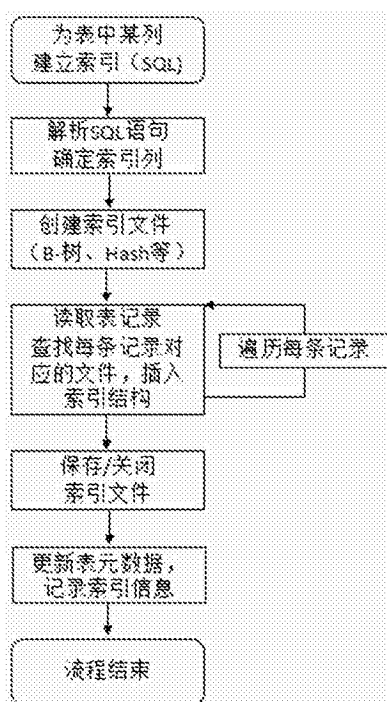


图3

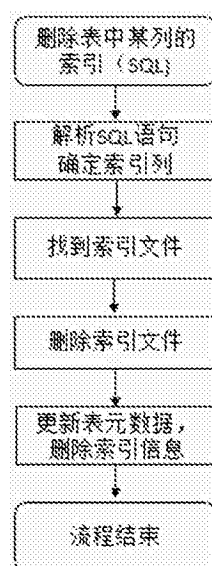


图4

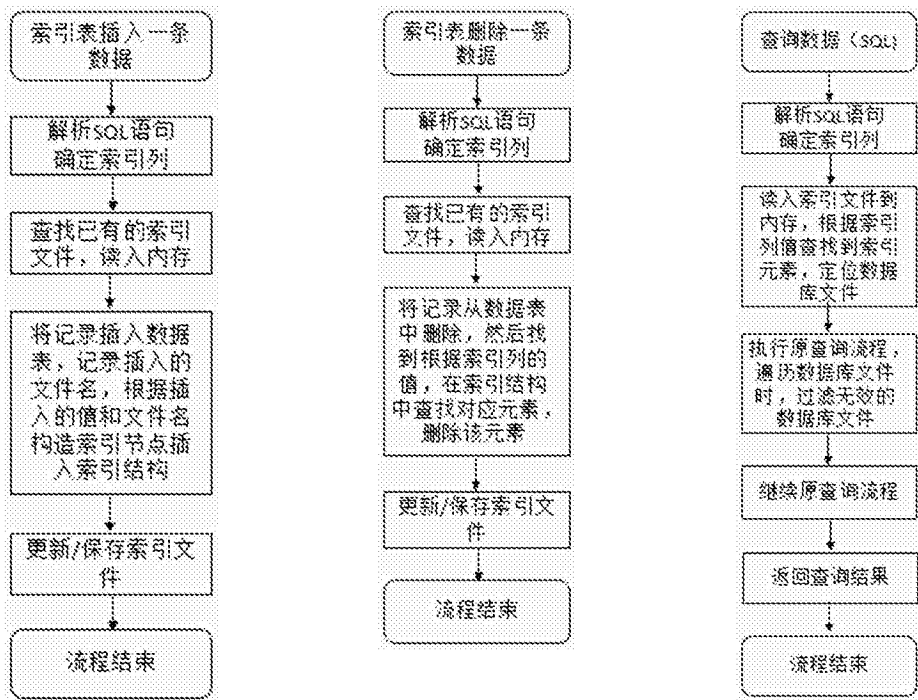


图6

图5

图7

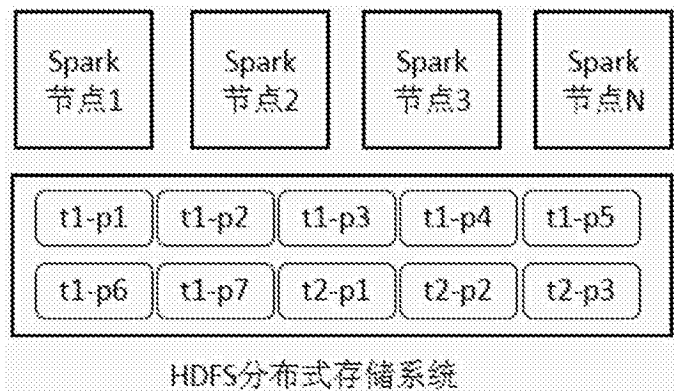


图8

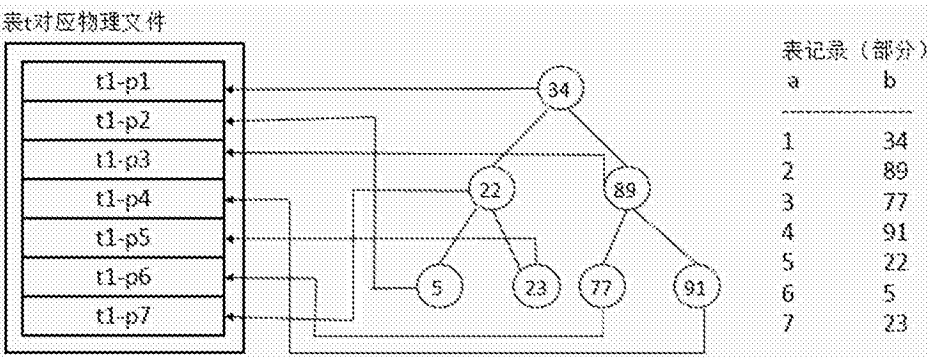


图9