Hive Index

2017年01月20日 15:30:56

**简介**

本文简单介绍一下hive索引的创建以及使用

**类型介绍**

hive当前的索引的类型主要有两种:   
- CompactIndexHandler(压缩索引)   
- Bitmap(位图索引)

**使用场景**

相比于传统数据库，Hive只提供有限的索引功能，通过在某些字段上建立索引来加速某些操作。通常当逻辑分区太多太细，partition无法满足时，可以考虑建立索引。用户可以使用EXPLAIN语法来分析HiveQL语句是否可以使用索引来提升用户查询的性能。像RDBMS中的索引一样，需要评估索引创建的是否合理，毕竟，索引需要更多的磁盘空间，并且创建维护索引也会有一定的代价。 用户必须要权衡从索引得到的好处和代价.

**CompactIndexHandler**

通过将列中相同的值得字段进行压缩从而减小存储和加快访问时间。需要注意的是Hive创建压缩索引时会将索引数据也存储在Hive表中。对于表tb\_index (id int, name string) 而言，建立索引后的索引表中默认的三列一次为索引列（id）、hdfs文件地址(\_bucketname)、偏移量(offset)。特别注意，offset列类型为array。

**Bitmap**

位图索引作为一种常见的索引，如果索引列只有固定的几个值，那么就可以采用位图索引来加速查询。利用位图索引可以方便的进行AND/OR/XOR等各类计算，Hive0.8版本开始引入位图索引，位图索引在大数据处理方面的应用广泛，比如可以利用bitmap来计算用户留存率（索引做与运算，效率远好于join的方式）。如果Bitmap索引很稀疏，那么就需要对索引压缩以节省存储空间和加快IO。Hive的Bitmap Handler采用的是EWAH（lemire/javaewah: A compressed alternative t…）压缩方式。

**案例**

有张表名为table的表，由三列组成，分别是姓名、性别和婚姻状况，其中性别只有男和女两项，婚姻状况由已婚、未婚、离婚这三项，该表共有100w个记录。现在有这样的查询：

* select \* from table where Gender=‘男’ and Marital=“未婚”;

|  |  |  |
| --- | --- | --- |
| 姓名(Name) | 性别(Gender) | 婚姻状况(Marital) |
| 张三 | 男 | 已婚 |
| 李四 | 女 | 已婚 |
| 王五 | 男 | 未婚 |
| 赵六 | 女 | 离婚 |
| 孙七 | 女 | 未婚 |
| … | … | … |

1）不使用索引

　　不使用索引时，数据库只能一行行扫描所有记录，然后判断该记录是否满足查询条件。

2）B树索引

　　对于性别，可取值的范围只有’男’,’女’，并且男和女可能各站该表的50%的数据，这时添加B树索引还是需要取出一半的数据， 因此完全没有必要。相反，如果某个字段的取值范围很广，几乎没有重复，比如身份证号，此时使用B树索引较为合适。事实上，当取出的行数据占用表中大部分的数据时，即使添加了B树索引，数据库如oracle、MySQL也不会使用B树索引，很有可能还是一行行全部扫描。

**位图索引出马**

如果用户查询的列的基数非常的小， 即只有的几个固定值，如性别、婚姻状况、行政区等等。要为这些基数值比较小的列建索引，就需要建立位图索引。

对于性别这个列，位图索引形成两个向量，男向量为10100…，向量的每一位表示该行是否是男，如果是则位1，否为0，同理，女向量位01011。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| RowId | 1 | 2 | 3 | 4 | 5 | … |
| 男 | 1 | 0 | 1 | 0 | 0 |  |
| 女 | 0 | 1 | 0 | 1 | 1 |  |

对于婚姻状况这一列，位图索引生成三个向量，已婚为11000…，未婚为00100…，离婚为00010…。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| RowId | 1 | 2 | 3 | 4 | 5 | … |
| 已婚 | 1 | 1 | 0 | 0 | 0 |  |
| 未婚 | 0 | 0 | 1 | 0 | 1 |  |

当我们使用查询语句

* select \* from table where Gender=‘男’ and Marital=“未婚”

的时候 首先取出男向量10100…，然后取出未婚向量00100…，将两个向量做and操作，这时生成新向量00100…，可以发现第三位为1，表示该表的第三行数据就是我们需要查询的结果。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| RowId | 1 | 2 | 3 | 4 | 5 | … |
| 男 | 1 | 0 | 1 | 0 | 0 |  |
| and |  |  |  |  |  |  |
| 未婚 | 0 | 0 | 1 | 0 | 1 |  |
| 结果 | 0 | 0 | 1 | 0 | 0 |  |

**位图索引的适用条件**

　　上面讲了，位图索引适合只有几个固定值的列，如性别、婚姻状况、行政区等等，而身份证号这种类型不适合用位图索引。

　　此外，位图索引适合静态数据，而不适合索引频繁更新的列。举个例子，有这样一个字段busy，记录各个机器的繁忙与否，当机器忙碌时，busy为1，当机器不忙碌时，busy为0。

　　这个时候有人会说使用位图索引，因为busy只有两个值。好，我们使用位图索引索引busy字段！假设用户A使用update更新某个机器的busy值，比如update table set table.busy=1 where rowid=100;，但还没有commit，而用户B也使用update更新另一个机器的busy值，update table set table.busy=1 where rowid=12; 这个时候用户B怎么也更新不了，需要等待用户A commit。

　　原因：用户A更新了某个机器的busy值为1，会导致所有busy为1的机器的位图向量发生改变，因此数据库会将busy＝1的所有行锁定，只有commit之后才解锁。



**使用索引**

**创建索引**

如上我们介绍的,hive目前支持的索引有compactindexhandler和bitmap.我们在创建的时候可以指定使用索引的类型.

* create index lxw1234\_index on table lxw1234(key) as
* 'org.apache.hadoop.hive.ql.index.compact.CompactIndexHandler'
* with deferred rebuild in table lxw1234\_index\_table;

with deferred rebuild: 这样创建的索引表中是没有数据的,要想使用索引，必须将索引进行rebuild: alter index lxw1234\_index on lxw1234 rebuild;

in table lxw1234\_index\_table: 指定创建索引后得到的表.