# 索引测试

100万行数据性能测试

实验者：宾友才

简单测试索引性能

* 分别在有索引和没有索引的情况下执行查询
* 分别在右索引和没有索引的情况下新增数据

# 准备环境

* 在数据库中创建测试表test

CREATE TABLE test(

Id int,

num int,

Pass varchar(50)

);

* 定义一个存储过程

mysql> delimiter $$

mysql> create procedure ceshi()

    -> begin

    -> declare pid int;

    -> set pid = 1000000;

    -> while pid>0 do

    -> insert into test values(pid,floor(pid+rand()\*pid),md5(pid));

    -> set pid = pid-1;

    -> end while;

-> end $$

mysql> delimiter ;

实验1:分别在右索引和没有索引的情况下执行查询

mysql> call ceshi();

这里生成100万条记录花费28.51秒

清空当前查询查询缓存

Reset query cacha;

mysql> select num,pass from test where id>50000 and id<100000;

查询结果49999 rows in set (0.30 sec)

创建表test2加入索引

mysql> create table test2( id int, num int, pass varchar(50), index id\_index (id) );

mysql> insert into test2 select \*from test;

Query OK, 1000000 rows affected (6.61 sec)

Records: 1000000  Duplicates: 0  Warnings: 0

然后查询test2表

mysql> select num,pass from test2 where id>50000 and id<100000;

结果为：

49999 rows in set (0.08 sec)

对比之下：两表的时间差=0.30-0.08=0.22秒

结论：有索引的情况下执行查询的效率比较高；

实验2:分别在有索引和没有索引的情况下新增数据。

这里我们新建一个test3表不加入索引。

mysql> create table test3(

    -> id int,

    -> num int,

    -> pass varchar(50)

    -> );

Query OK, 0 rows affected (0.01 sec)

刷新一下查询缓存

mysql> reset query cache;

Query OK, 0 rows affected, 1 warning (0.00 sec)

把test表中的数据导入到test3中

mysql> insert into test3 select \*from test;

Query OK, 1000000 rows affected (3.44 sec)

Records: 1000000  Duplicates: 0  Warnings: 0

根据第一个实验，我们创建了test2同时加入了索引

mysql> create table test2( id int, num int, pass varchar(50), index id\_index (id) );

mysql> insert into test2 select \*from test;

Query OK, 1000000 rows affected (6.61 sec)

Records: 1000000  Duplicates: 0  Warnings: 0

对比之下：6.61-3.44=2.67秒

结论：在没有索引的情况下新增数据的效率比较高

# 综上两个实验结合实际得出结论：

随着数据量的增大，数据结构的复杂，这个差异也会变得更大，所以说呢，其实我们在导入数据的时候，我们更关心的是什么？尽快的把数据导入到数据库当中，然后我们回头往数据库的表建立索引，先能把这些数据用上，这是最重要的。因此一个没有索引的表导入数据要比有索引的表导入数据的速度要快的多，为什么？因为如果一个表建立了索引，再导入数据，每增加一行数据都需要重写索引，这个重写索引的操作就增加了额外的操作时间，而对于一个没有索引的表只需要写入数据即可。当然了，后期在这个表当中在建立索引还是会消耗时间，可能两者的相差时间会大于前面那两种，但是呢，在实际应用上面，第三种方案会给我们带来好处。