**mysq order by 不走索引问题**

[](https://www.jianshu.com/u/2818b62e3023)

[leoss\_H](https://www.jianshu.com/u/2818b62e3023) 关注

2019.01.04 15:43 字数 1123 阅读 63评论 0喜欢 0

**问题描述**

今天遇到一个问题，在一个报表导出的功能中，需要导出用户的某些记录的所有数据，有些用户有多达10万的记录，导出过程非常慢，需要10多秒。经过排查，发现是查询数据库的时候的问题，查询数据居然用了7秒左右。起初看到sql没发现什么问，sql写得中规中矩，只查询需要的列，使用索引，遵循索引规则等等，但是它就是要那么长时间。

**问题分析**

表面上看不出问题，那只能用explain大法来分析分析了

原sql（这里隐去关键信息）

select drawno,col from `tb\_loxxx` where drawno >'15010101' order by drawno desc;

expalin结果

expalin结果

从extra里出现了Using filesort，说明这里是没有走索引的，而且type为ALL，说明进行了一次全表扫描。而该表确实对drawno字段建立了索引，那么是什么原因导致这里出现Using filesort呢？在继续分析前先来了解下Using filesort。如果对mysql稍有了解的话，就会知道Using filesort代表着mysql进行了排序操作。下面是官方的描述

Using filesort：  
MySQL must do an extra pass to find out how to retrieve the rows in sorted order. The sort is done by going through all rows according to the join type and storing the sort key and pointer to the row for all rows that match the WHERE clause.  
Mysql需要额外的一次传递，以找出如何按排序顺序检索行，通过根据联接类型浏览所有行并为所有匹配where子句的行保存排序关键字和行的指针来完成排序，然后关键字被排序，并按排序顺序检索行。

这句话直接翻译过来有点难以理解，简单说就是由于索引不满足你的sql，mysql需要对数据行进行一次额外的排序操作，这个排序操作==既费空间又费时间==。当数据量较少的时候并不会对应用产生多大影响，但数据量一多，就会出现非常可怕的后果，轻则服务响应变慢，重则拖垮服务，甚至引发雪崩效应导致应用宕机。

再回来看看我的sql，查询列和搜索条件应该都没有问题，那么应该order by影响了。通过搜索发现，order by 使用不当确实会导致索引失效。

**解决方案**

**1.强制索引 FORCE INDEX（key）**

force index 的作用是让mysql强制使用某个索引，对应的有ignore index 强制忽略索引。**除非非常明确sql目的和运行效率情况下，一般不推荐使用这2个操作：一是强制索引可能导致其他索引失效，二是强制索引不一定会提高sql效率，还会导致mysql优化器没有作用，三是强制使用的索引如果被删除了，会导致程序异常。**  
强制索引使用方式如下：

select drawno,col from `tb\_loxxx` force index(tb\_loxxx\_01) where drawno >'15010101' order by drawno desc;

再次进行expalin

再次expalin结果

可以看到使用了索引，type也变成了range。再次执行sql，发现sql的效率又回到正常水平了，仅用300多毫米就完成了这次查询。

**2.联合索引**

联合索引就行将需要的列组成一个索引。这个需要根据具体sql分析。  
针对我这个sql，是因为我的order by 和搜索列不匹配，order by drawno，而搜索列除了drawno，还有其他列，根据sql条件，我建立相应索引后，sql效率也回归到正常水平。

**从效率上来说联合索引和强制索引基本是一样，从应用上来说联合索引会好一些，但是如果sql复杂或者数据库设计的混乱，可能造成一个表建立一堆索引的情况，索引太多对表性能也有很多影响，最根本的解决方案还是简化sql，去掉不必要的查询等等。具体使用哪种方式，就看大家自己需要了。**