94 MySQL是如何根据成本优化选择执行计划的？（上）

之前已经给大家讲解清楚了 MySQL 在执行单表查询时候的一些执行计划，比如说const、ref、range、index、all之类的，也讲了多表关联的时候是如何执行的，本质其实就是先查一个驱动表，接着根据连接条件去被驱动表里循环查询，现在大家对MySQL执行查询的一些基本原理都有了一个了解了。

好，那么从今天开始，我们再更深入一步，因为其实大家之前或多或少也感觉到了一个问题，就是其实我们在执行单表查询也好，多表关联也好，似乎都有多种执行计划可以选择，比如有的表可以全表扫描，也可以用索引A，也可以用索引B，那么到底是用哪种执行计划呢？

所以今天开始，我们用为期两周的时间，彻底给大家讲解清楚MySQL是如何对一个查询语句的多个执行计划评估他的成本的？如何根据成本评估选择一个成本最低的执行计划，保证最佳的查询速度？

大家耐心学习，我们已经一点一点接近了MySQL查询原理的本质了，当大家透彻理解了这些内容，再去学习通过explain看真实的SQL语句的执行计划，就会完全明白是怎么回事了。当你能透彻理解了explain看SQL执行计划之后，那么任何SQL语句的调优都不在话下。

我们先了解一下MySQL里的成本是什么意思，简单来说，跑一个SQL语句，一般成本是两块，首先是那些数据如果在磁盘里，你要不要从磁盘里把数据读出来？这个从磁盘读数据到内存就是IO成本，而且MySQL里都是一页一页读的，读一页的成本的约定为1.0。

然后呢，还有一个成本，那就是说你拿到数据之后，是不是要对数据做一些运算？比如验证他是否符合搜索条件了，或者是搞一些排序分组之类的事，这些都是耗费CPU资源的，属于CPU成本，一般约定读取和检测一条数据是否符合条件的成本是0.2.

这个所谓1.0和0.2就是他自定义的一个成本值，代表的意思就是一个数据页IO成本就是1.0，一条数据检测的CPU成本就是0.2，就这个意思罢了。

然后呢，当你搞一个SQL语句给MySQL的时候，比如：

select * from t where x1=xx and x2=xx

此时你有两个索引，分别是针对x1和x2建立的，就会先看看这个SQL可以用到哪几个索引，此时发现x1和x2的索引都能用到，他们俩索引就是possible keys。

接着会针对这个SQL计算一下全表扫描的成本，这个全表扫描的话就比较坑了，因为他是需要先磁盘IO把聚簇索引里的叶子节点上的数据页一页一页都读到内存里，这有多少数据页就得耗费多少IO成本，接着对内存里的每一条数据都判断是否符合搜索条件的，这有多少条数据就要耗费多少CPU成本。

所以说，此时就得计算一下这块成本有多少，怎么算呢？简单，教大家一个命令：

show table status like "表名"

可以拿到你的表的统计信息，你在对表进行增删改的时候，MySQL会给你维护这个表的一些统计信息，比如这里可以看到rows和data_length两个信息，不过对于innodb来说，这个rows是估计值。

rows就是表里的记录数，data_length就是表的聚簇索引的字节数大小，此时用data_length除以1024就是kb为单位的大小，然后再除以16kb（默认一页的大小），就是有多少页，此时知道数据页的数量和rows记录数，就可以计算全表扫描的成本了。

IO成本就是：数据页数量 * 1.0 + 微调值，CPU成本就是：行记录数 * 0.2 + 微调值，他们俩相加，就是一个总的成本值，比如你有数据页100个，记录数有2万条，此时总成本值大致就是100 + 4000 = 4100，在这个左右。

好，今天先讲到这儿，大家先知道了一个全表扫描执行计划的成本计算方法，下次我们继续讲索引的成本计算方法。

End