

95 MySQL是如何根据成本优化选择执行计划的？（中）

上次我们讲完了全表扫描的成本计算方法，相信大家应该都理解了，其实还是比较简单的，今天我们来讲一下索引的成本计算方法，因为除了全表扫描之外，还可能多个索引都可以使用，但是当然同时一般只能用一个索引，所以不同索引的使用成本都得计算一下。

这个使用索引访问数据的方式，大家应该都还记得，其实很简单，除非你直接根据主键查，那就直接走一个聚簇索引就ok了，否则普通索引，一般都是两步走，先从二级索引查询一波数据，再根据这波数据的主键去聚簇索引回表查询。

这个过程的成本计算方法稍微有点特别，首先，在二级索引里根据条件查一波数据的IO成本，一般是看你的查询条件涉及到几个范围，比如说name值在25~100, 250~350两个区间，那么就是两个范围，否则name=xx就仅仅是一个范围区间。

一般一个范围区间就粗暴的认为等同于一个数据页，所以此时可能一般根据二级索引查询的时候，这个IO成本都会预估的很小，可能就是 $1 * 1.0 = 1$ ，或者是 $n * 1.0 = n$ ，基本就是个位数这个级别。

但是到此为止，还仅仅是通过IO读取了二级索引的数据页而已，这仅仅是二级索引读取的IO成本，但是二级索引数据页到内存里以后，还得根据搜索条件去拿出来一波数据，拿这波数据的过程就是根据搜索条件在二级索引里搜索的过程。

此时就要估算从二级索引里读取符合条件的数据的成本了，这需要估算一下在二级索引里会查出多少条数据，这个过程就稍微有点复杂了，不细讲了，总之呢，他会根据一个不是怎么太准确的算法去估算一下根据查询条件可能会在二级索引里查出多少条数据来。

估算出来之后，比如估算可能会查到100条数据，此时从二级索引里查询数据的CPU成本就是 $100 * 0.2 + \text{微调值}$ ，总之就是20左右而已。

接着你拿到100条数据之后，就得回表到聚簇索引里去查询完整数据，此时先估算回表到聚簇索引的IO成本，这里比较粗暴的直接默认1条数据就得回表到聚簇索引查询一个数据页，所以100条数据就是100个数据页的IO成本，也就是 $100 * 1.0 + \text{微调值}$ ，大致是100左右。

接着因为在二级索引里搜索到的数据是100条，然后通过IO成本最多回表到聚簇索引访问100个数据页之后，就可以拿到这100条数据的完整值了，此时就可以针对这100条数据去判断，他们是否符合其他查询条件了，这里耗费的CPU成本就是 $100 * 0.2 + \text{微调值}$ ，就是20左右。

把上面的所有成本都加起来，就是 $1 + 20 + 100 + 20 = 141$ ，这就是使用一个索引进行查询的成本的计算方法，其实大家看明白这个过程了，那么每一个索引的成本计算过程就都明了了，假设你直接根据主键查询，那么也参考上述估算过程就可以了，那就不过是仅仅查询一个聚簇索引罢了。

总之，上次讲到全表扫描发现成本是4100左右，这次根据索引查找可能就141，所以，很多时候，使用索引和全表扫描，他的成本差距是非常之大的。所以一般就会针对全表扫描和各个索引的成本，都进行估算，然后比较一下，选择一个成本最低的执行计划。

End