Mysql 查询性能优化

1:查询耗时操作包括:网络，cpu计算，生成统计信息，执行计划，锁等待等操作

2:分析查询性能低下的步骤

2.1:确认应用程序是否在检索大量超过需要的数据

2.2:确认mysql服务器是否在分析大量超过需要的数据行

3:是否向数据库请求了不需要的数据

3.1:查询不需要的记录

3.2:多表关联时返回全部列

3.3:总是取出全部列

3.4:重复查询相同的数据

4:衡量查询开销的指标

4.1:响应时间

4.2:扫描的行数和返回的行数

4.3:扫描的行数和访问类型（explain type all or ref）

5:mysql where条件使用，由好到坏依次为

5.1:在索引中使用where条件来过滤不匹配的记录

5.2:使用索引覆盖扫描（explain extra中出现using index）来返回记录

5.3:从数据表中返回数据（explain extra中出现using where）来返回记录

6:如果发现查询需要扫描大量数据但只返回少数的行，优化方法如下

6.1:使用索引覆盖扫描。把所有需要用的列都放在索引中，这样存储引擎无需回表获取对应行就可以返回结果

6.2:改变表结构，使用单独的汇总表

6.3:重写这个复杂的查询，让mysql优化器能够以更优化的方式执行这个查询

7:分解关联查询重构查询优势如下

7.1:让缓存的效率更高

7.2:查询分解，执行单个查询可以减少锁的竞争

7.3:在应用层做关联，可以更容易对数据库进行拆分，更容易做到高性能和可扩展

7.4:查询本身效率也可能会有所提升

7.5:可以减少冗余记录的查询

7.6:相当于在应用中实现了哈希关联。而不是使用mysql的嵌套循环关联

8：查询执行的生命周期

8.1：客户端发送一条查询指令给服务器

8.2：服务器先检查查询缓存，如果命中了缓存，则立刻返回存储在缓存中的结果。否则进入下一阶段。

8.3：服务器端进行SQL解析、预处理，再有优化器生成对应的执行计划。

8.4：MySQL根据优化器生成的执行计划，再电泳存储引擎的API来执行查询

8.5：将结果返回客户端

9：Show FULL PROCESSLIST （该命令返回结果中的Command列就表示当前的状态）

9.1：Sleep 线程正在等待客户端发送新的请求

9.2：Query 线程正在执行查询或者正在将结果发送给客户端

9.3：Locked 在mysql服务器层面，该线程正在等待表锁

9.4：ANalyzing and statistics 线程正在收集存储引擎的统计信息，并生成查询的执行计划

9.5：Copying to tmp table [on dist] 线程正在执行查询，并且将其结果集都复制到一个临时表中

9.6：Sorting result 线程正在对结果集进行排序

9.7：Send data 线程可能在多个状态之间传送数据，或者在生成结果集，或者在向客户端返回数据

10：查询缓存

10.1：在解析一个查询语句之前，如果查询是打开的，那么MySQL会优化检查这个查询是否命中查询缓存中的数据。