Java MySQL系列课程06

------ MySQL优化之应用优化

本文档包含以下内容：

1. 连接池
2. 增加Cache缓存层

#### 前言：

除了优化MySQL Server端的SQL和索引，很多时候，在实际生产环境中，由于数据库服务器本身的性能局限，就必须要对前台的应用来进行一些优化，使得前台访问数据库的压力能够减到最小。本章我们来看看在应用上，我们一般都会做哪些优化。

#### 一 连接池：

应用上一般访问数据库，都是先和MySQL Server创建连接，然后发送SQL语句，Server处理完成后，再把结果通过网络返回给应用，然后关闭和MySQL Server的连接，因此短时间大量的数据库访问，消耗的TCP三次握手和四次挥手所花费的时间就很大了，稍微大一点的项目，我们都会在应用访问数据库的那一层上，添加连接池模块，相当于应用和MySQL Server事先创建一组连接，当应用需要请求MySQL Server时，不需要再进行TCP连接和释放连接了，一般连接池都会维护以下资源：

1. 连接池里面保持固定数量的活跃TCP连接，供应用使用。
2. 如果应用瞬间访问MySQL的量比较大，那么连接池会实时创建更多的连接给应用使用。
3. 当连接池里面的TCP连接一段时间内没有被用到，连接池会释放多余的连接资源，保留它设置的最大空闲连接量就可以了。

**连接池可以自己实现，也可以用第三方写好的库**。

#### 二 增加Cache缓存层：

比较有名的，提高数据库访问效率的缓存层**当然是redis和memcache了**，一般用缓存把经常访问的数据缓存起来，相当于就是提高了数据库的访问效率。

但是，添加Cache缓存层有一个非常大的问题，就是需要考虑**“缓存数据的一致性”问题。**

**当多个事务并发读取的时候，那么先从缓存命中，如果不能命中，则去数据库中读取，再写缓存，并发读没有问题！**

**当多个事务进行读写的时候，事务1先写数据库，还没等更新缓存，事务2把缓存的旧值读走了，数据不一致；或者事务1写之前先把缓存删掉，再写数据库，最后写缓存，但是当事务1写数据库还没写完呢，事务2把旧值又读走了，又造成缓存数据不一致的问题！**

因为大家做项目的过程中都使用了缓存层，所以**“缓存数据一致性”问题被面试官大量提及**，这个在课堂上会细讲，但是如果你要和面试官讨论Cache，那么这个数据一致性问题你就得准备好，很有可能会问到！