秒杀场景下超卖问题解决方案

作者：xjlnjut730
原文链接：https://ld246.com/article/1536335417613
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

问题域一定要把握好。今天有人问到我“秒杀场景下超卖问题怎么解决”，我立马把问题域扩大到“杀系统要怎么建设”，方向已经偏到十万八千里了，而我也不知道如何回答了。

秒杀超卖现象：在高并发下，多个线程并发更新库存，导致库存为负的情况。

我搜集了一些资料，整理了一下，秒杀可选方案主要有以下三种：

1. 悲观锁

悲观锁方案最容易理解：在更新库存期间加锁，不允许其它线程修改。

1.1 select xxx for update

优点：确保了线程安全。

缺点：高并发场景下会导致多个请求一直等待，数据库性能下降，系统的链接数上升，负载飙升，影系统的平均响应时间，甚至会瘫痪。

1.2 文件锁

优点与1.1类似，缺点是磁盘IO开销会变大。

1.3 缓存锁

当用户A要修改某个id的数据时，把要修改的id存入缓存，若其他用户触发修改此id的数据时，读到mmcache有这个id的值时，就阻止其它用户修改。

优点与缺点与1.2，但总体效果要好于以上两种方案。如果缓存是独立的集群，还可以解决跨进程乐锁处理不了的问题。

1.4 分布式锁

与1.3类似。

2. 乐观锁

使用带版本号的更新。每个线程都可以并发修改，但在并发时，只有一个线程会修改成功，其它会返失败。

Redis的watch

3. FIFO队列

通过FIFO队列，使修改库存的操作串行化。

Redis的队列

优点：不需要在单独加锁（无论是悲观锁还是乐观锁）。

缺点：队列的长度是有限的，必须控制好，不然请求会越积越多。

4. 结论

总的来说，不能把压力放在数据库上，所以使用"select xxx for update"的方式在高并发的场景下是可行的。FIFO同步队列的方式，可以结合库存限制队列长，但是在库存较多的场景下，又不太适用。以相对来说，我会倾向于选择：乐观锁/缓存锁/分布式锁的方式。

5. 参考文献

https://blog.csdn.net/maikelsong/article/details/53322942

https://www.aliyun.com/jiaocheng/808179.html

https://blog.csdn.net/b1303110335/article/details/81365347