如何解决redis的并发竞争key问题

当有多个子系统去set一个key，这个时候要注意什么呢？大家思考过么？基本都是推荐用redis事务机制。但是这并不是完美的，因为我们的生产环境，基本都是redis集群环境，做了数据分片操作。一个事务中有涉及到多个key操作的时候，这多个key不一定都存储在同一个redis-server上。因此，redis的事务机制，十分鸡肋。

推荐下面的做法：

1. 如果对这个key操作，不要求顺序

这种情况下，准备一个分布式锁，大家去抢锁，抢到锁就做set操作即可，比较简单。

1. 如果对这个key操作，要求顺序

假设有一个key1，系统A需要将key1设置为valueA，系统B需要将key1设置为valueB，系统C需要将key1设置为valueC。

期望按照key1的value值按照valueA -> valueB -> valueC的顺序变化。这种时候我们在数据写入数据库的时候，需要保存一个时间戳。假设时间戳如下：

|  |  |
| --- | --- |
| 系统A key1 | {valueA 3:00} |
| 系统B key1 | {valueB 3:05} |
| 系统C key1 | {valueC 3:10} |

那么，假设现在系统B先抢到锁，将key1设置为{valueB 3:05}。接下来系统A抢到锁，发现自己的valueA的时间戳早于缓存中的时间戳，那就不做set操作了。以此类推。

其他方法，比如利用队列，将set方法变成串行访问也可以。总之，灵活变通。