



说说你对分布式锁的理解

- 问题1： 1、什么是分布式锁？

- 首先，为了确保分布式锁可用，我们至少要确保锁的实现同时满足以下三个条件：

- 互斥性。在任意时刻，只有一个客户端能持有锁。
- 不会发生死锁。即使有一个客户端在持有锁的期间崩溃而没有主动解锁，也能保证后续其他客户端能加锁。
- 解铃还须系铃人。加锁和解锁必须是同一个客户端，客户端自己不能把别人加的锁给解开了

- 从 Redis 2.6.0 版本开始，通过内置的 Lua 解释器，可以使用 EVAL 命令对 Lua 脚本进行求值。

- Redis 使用单个 Lua 解释器去运行所有脚本，并且，Redis 也保证脚本会以原子性(atomic)的方式执行：当某个脚本正在运行的时候，不会有其他脚本或 Redis 命令被执行。这和使用 MULTI / EXEC 包围的事务很类似。在其他别的客户端看来，脚本的效果(effect)要么是不可见的(not visible)，要么就是已完成的(already completed)。

- 问题2： 怎么实现分布式锁

- 实现分布式锁的方案大概有两种

- 采用lua脚本操作分布式锁
- 采用setnx、setex命令连用的方式实现分布式锁

- 问题3：

- 解锁需要注意什么

解铃还须系铃人。加锁和解锁必须是同一个客户端，客户端自己不能把别人加的锁给解了