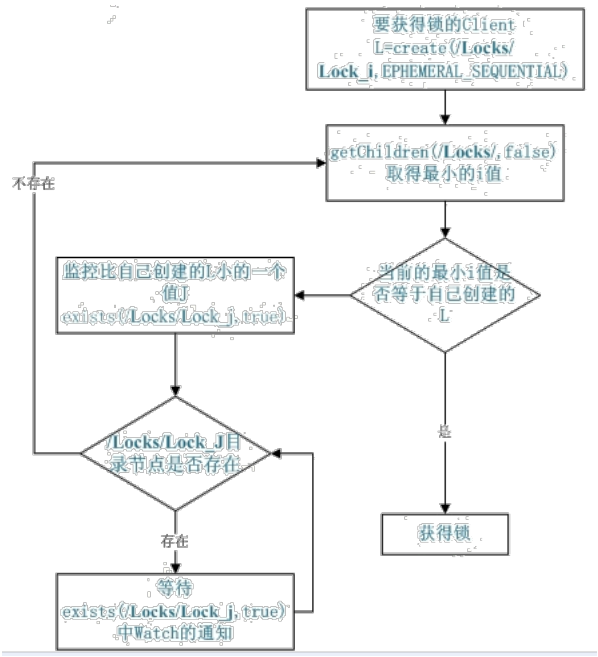
> 阿里巴巴-蚂蚁金服 3.9

1. 说说三种分布式锁？

1、Zookeeper：基于zookeeper瞬时有序节点实现的分布式锁，其主要逻辑如下（该图来自于IBM网站）。大致思想即为：每个客户端对某个功能加锁时，在zookeeper上的与该功能对应的指定节点的目录下，生成一个唯一的瞬时有序节点。判断是否获取锁的方式很简单，只需要判断有序节点中序号最小的一个。当释放锁的时候，只需将这个瞬时节点删除即可。同时，其可以避免服务宕机导致的锁无法释放，而产生的死锁问题。



2、优点

锁安全性高，zk可持久化，且能实时监听获取锁的客户端状态。一旦客户端宕机，则瞬时节点随之消失，zk因而能第一时间释放锁。这也省去了用分布式缓存实现锁的过程中需要加入超时时间判断的这一逻辑。

3、缺点

性能开销比较高。因为其需要动态产生、销毁瞬时节点来实现锁功能。所以不太适合直接提供给高并发的场景使用。

4、实现

可以直接采用zookeeper第三方库curator即可方便地实现分布式锁。

5、适用场景

对可靠性要求非常高，且并发程度不高的场景下使用。如核心数据的定时全量/增量同步等。

2、memcached：memcached带有add函数，利用add函数的特性即可实现分布式锁。add和set的区别在于：如果多线程并发set，则每个set都会成功，但最后存储的值以最后的set的线程为准。而add的话则相反，add会添加第一个到达的值，并返回true，后续的添加则都会返回false。利用该点即可很轻松地实现分布式锁。

2、优点

并发高效

3、缺点

memcached采用列入LRU置换策略，所以如果内存不够，可能导致缓存中的锁信息丢失。

memcached无法持久化，一旦重启，将导致信息丢失。

4、使用场景

高并发场景。需要 1)加上超时时间避免死锁； 2)提供足够支撑锁服务的内存空间； 3)稳定的集群化管理。

3、redis：redis分布式锁即可以结合zk分布式锁锁高度安全和memcached并发场景下效率很好的优点，其实现方式和memcached类似，采用setnx即可实现。需要注意的是，这里的redis也需要设置超时时间。以避免死锁。可以利用jedis客户端实现。

```
ICacheKey cacheKey = new ConcurrentCacheKey(key, type);
return RedisDao.setnx(cacheKey, "1");
```

备注：redis面试题推荐：https://www.jianshu.com/p/16b6db9ee410

2. redis的实现原理？

需要详读这

篇：https://h2pl.github.io/2018/07/08/Redis%E5%8E%9F%E7%90%86%E4%B8%8E%E5%AE%9E%E8%B7%B5%E6%80%BB%E7%BB%93/

3. redis数据结构，使用场景?

   https://segmentfault.com/a/1190000004012214

4. redis集群有哪几种?

   三种集群策略：https://blog.csdn.net/q649381130/article/details/79931791

5. codis原理?

   redis集群方案-codis：http://www.yunweipai.com/archives/14556.html

6. 是否熟悉金融业务？记账业务？蚂蚁金服对这部分有要求。