参考：http://www.open-open.com/lib/view/open1473405167217.html

唯品会多线程redis设计与实现|Redis中国用户组

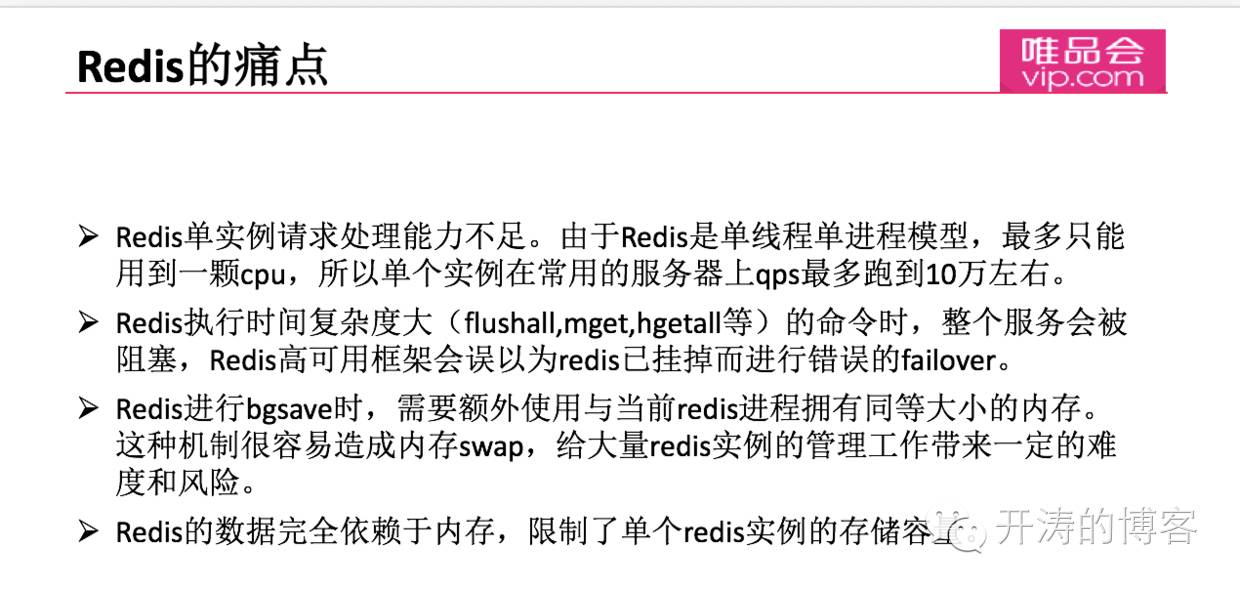
目录：

### redis cluster的C客户端(hiredis-vip)

### 集群迁移工具(redis-migrate-tool)

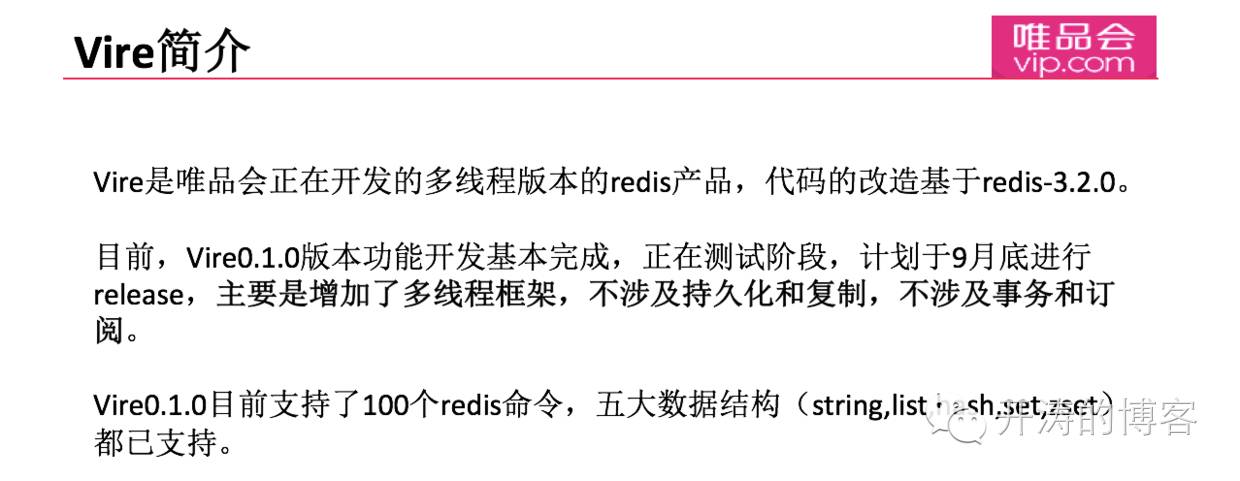
### 多线程版Twemproxy(Twemproxies)

大家好，我是deep，今天跟大家分享下我们正在开发的多线程redis。在我们的redis使用中，发现了一些痛点问题，涉及到了redis框架的设计。



所以bgsave需要机器有至少两倍的redis运行消耗内存。

我们线上有大量的redis实例在运行，规模比较庞大，有些redis集群实例规模超过100+，我们开始对redis进行了多线程版本的改造，就是我们现在正在开发的产品vire。



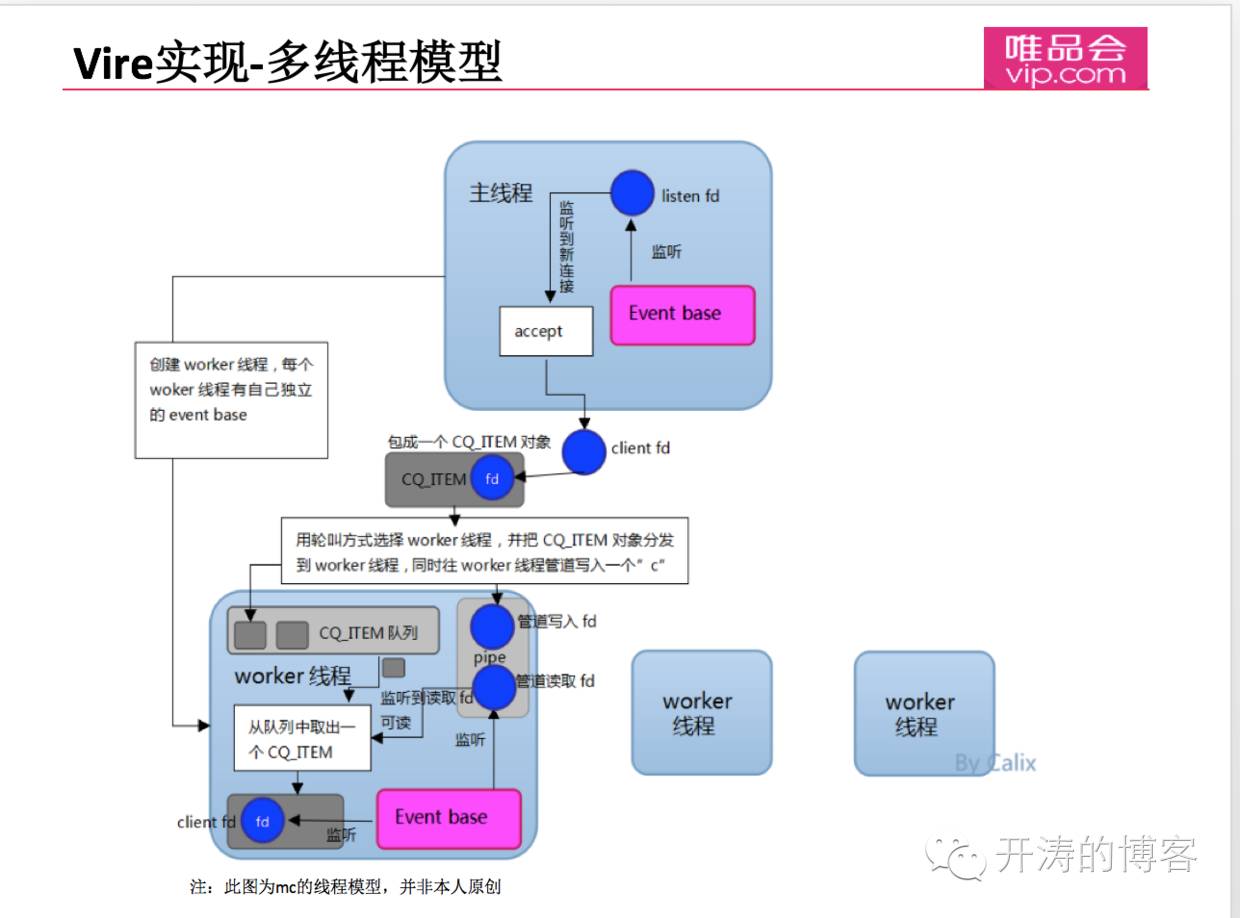
这是vire的一个现状，分几个阶段进行开发，现在是0.1.0版本。



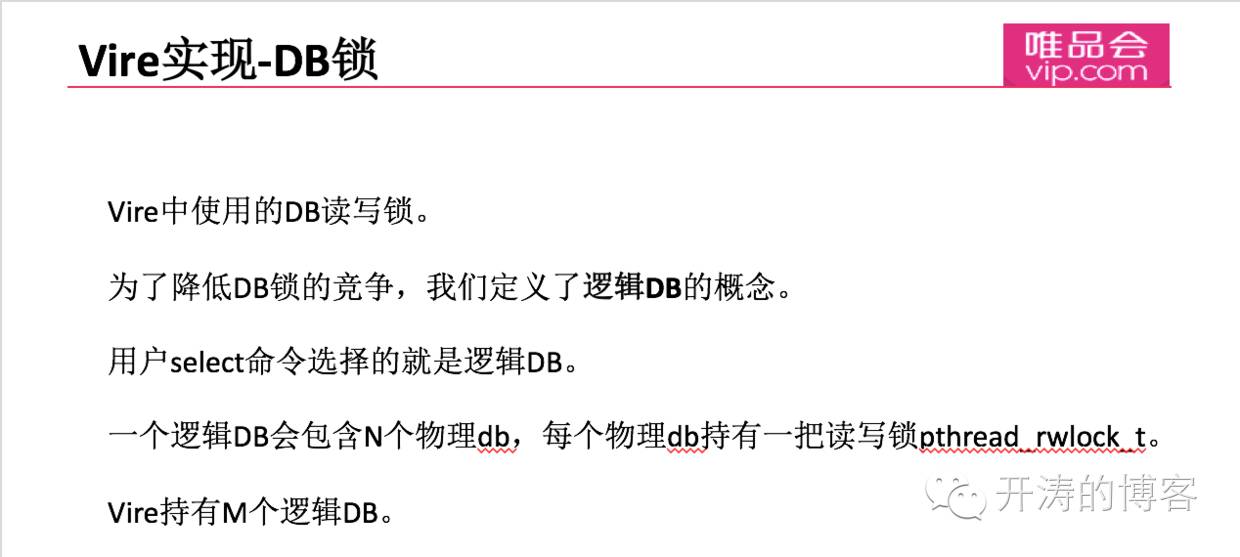
以上是vire0.1.0的一些设计思路。下面说说具体实现：



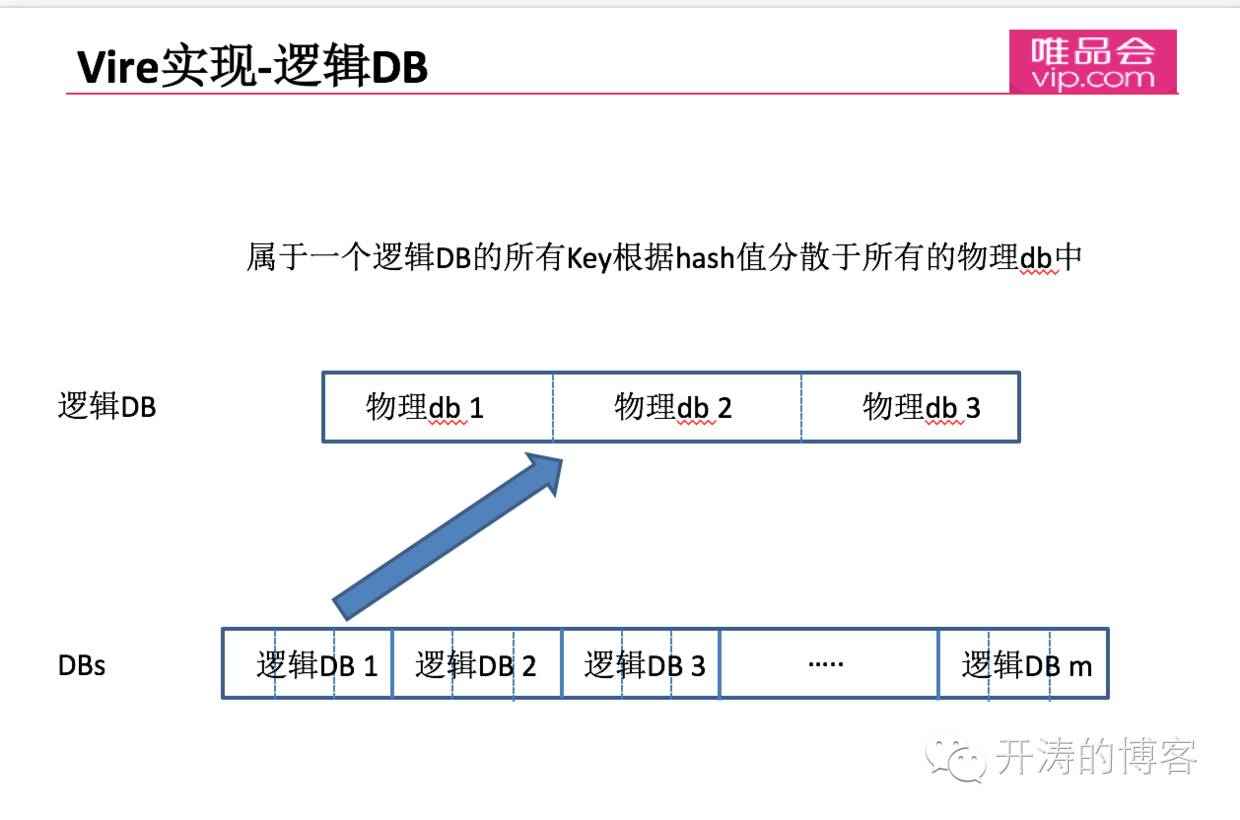
这是vire的多线程模型，借鉴于memcached，master+worker线程模型。



这个图比较直观的反映了线程模型的工作原理，多线程不可避免会用到锁，以下是vire的锁机制：



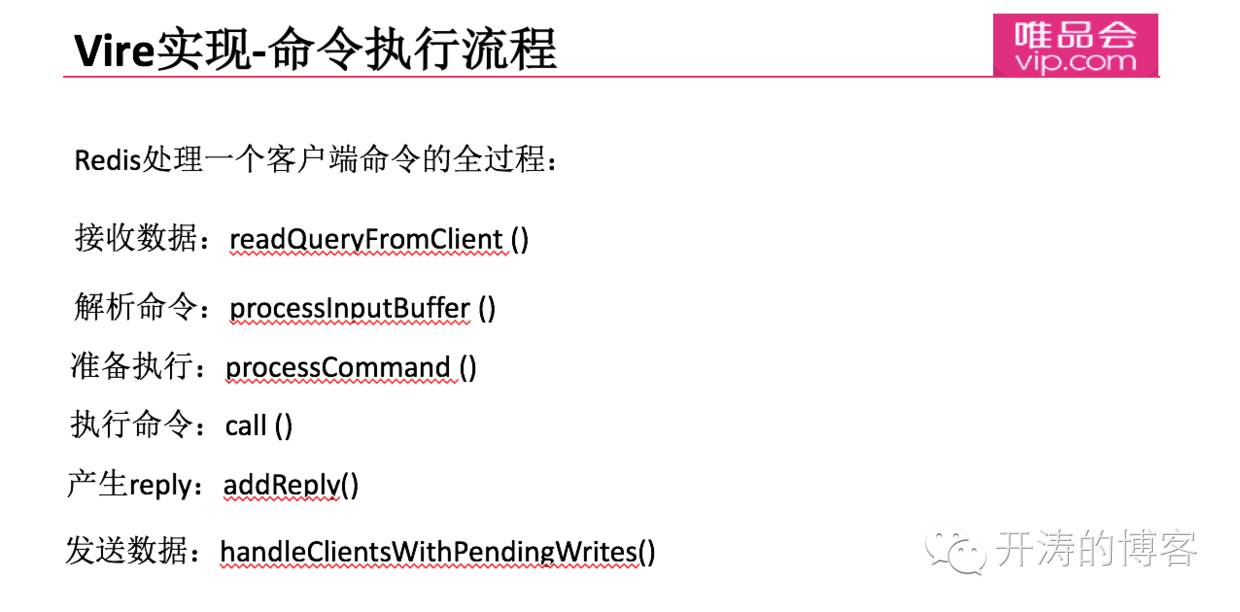
这里有个逻辑DB的概念，其实就是把多个redis db伪装成一个DB提供给用户。DB级别的锁，会不会性能很差呢？后续会有测试报告给出。



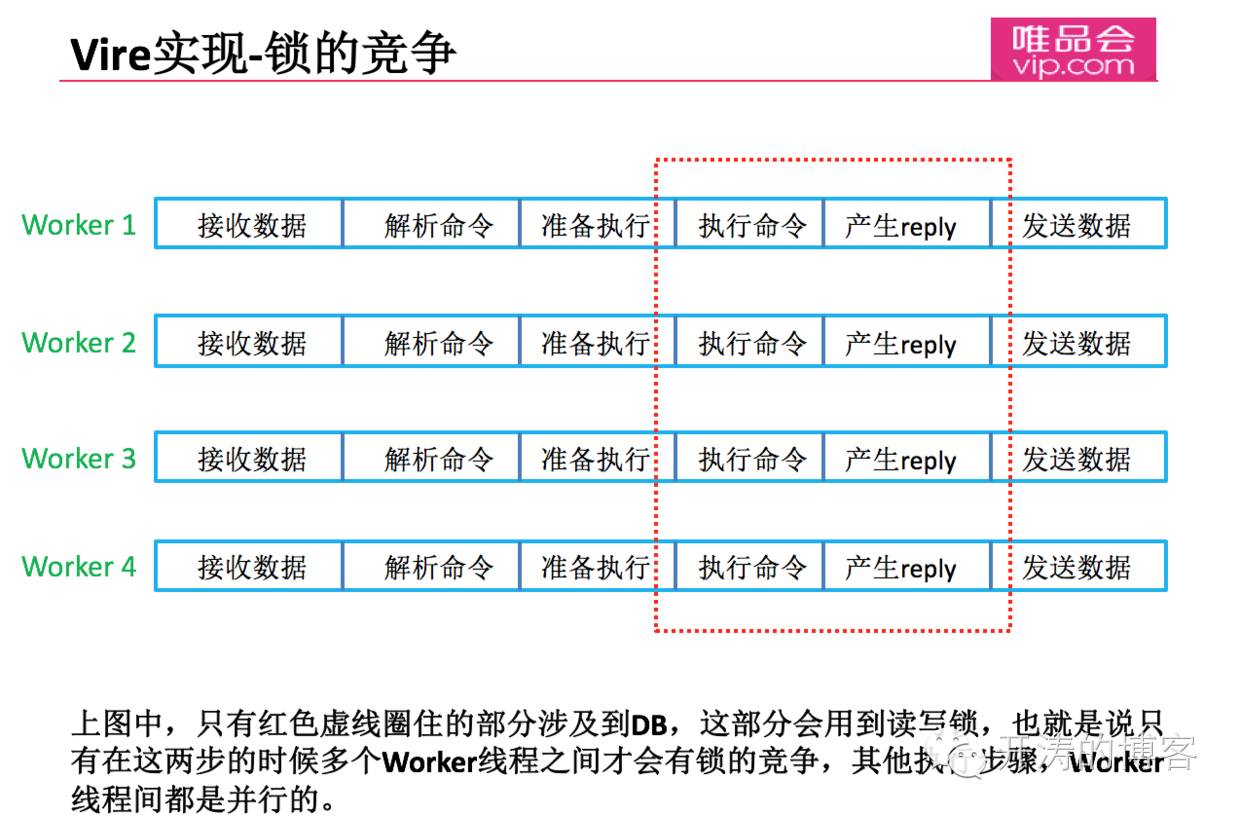
用户使用的所有key，是通过key的hash值被分散到了各个物理db上，目的就是降低DB锁的竞争提升qps，可以通过info 命令看到物理db中key的分布：



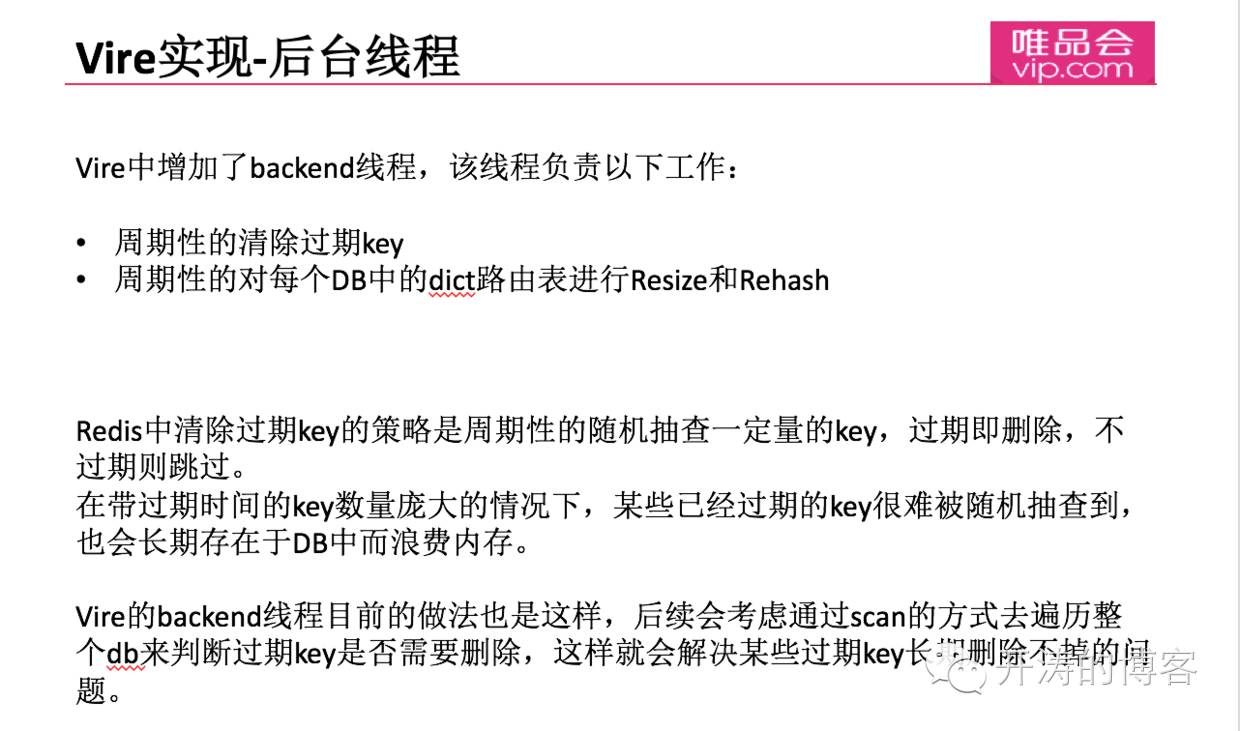
下面来看下成功执行一个redis命令要走的流程：



我们的db锁是在哪一步使用呢？



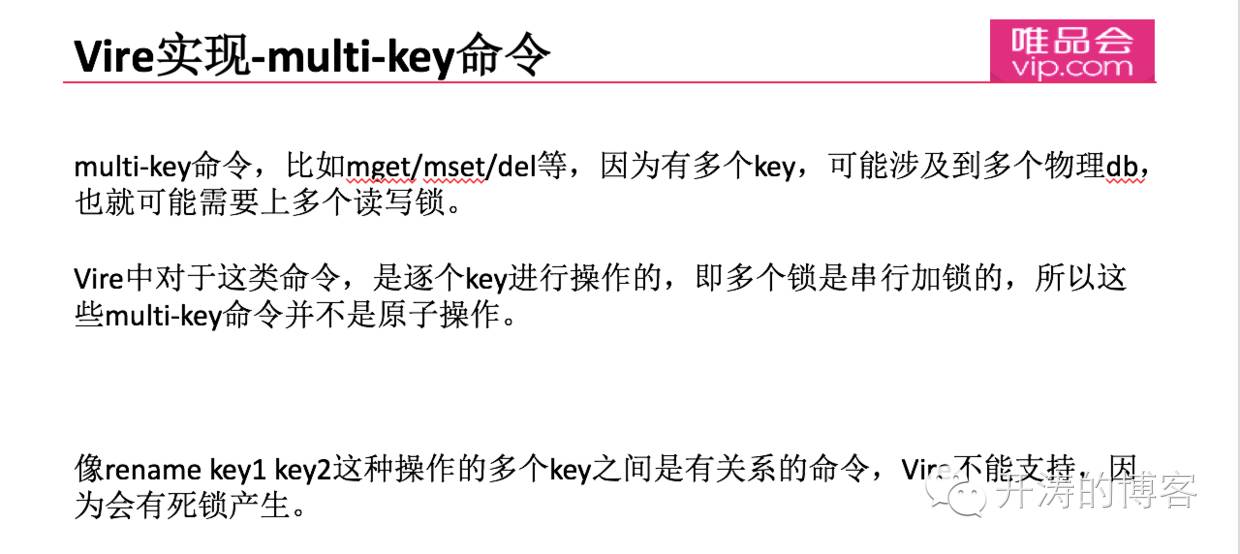
有可能用到db锁的步骤就是红框中的两步，但像ping这样命令，在整个过程中是用不到db锁的，可以看出，Worker线程在一部分时间是完全并行执行的，关于vire中的后台线程：



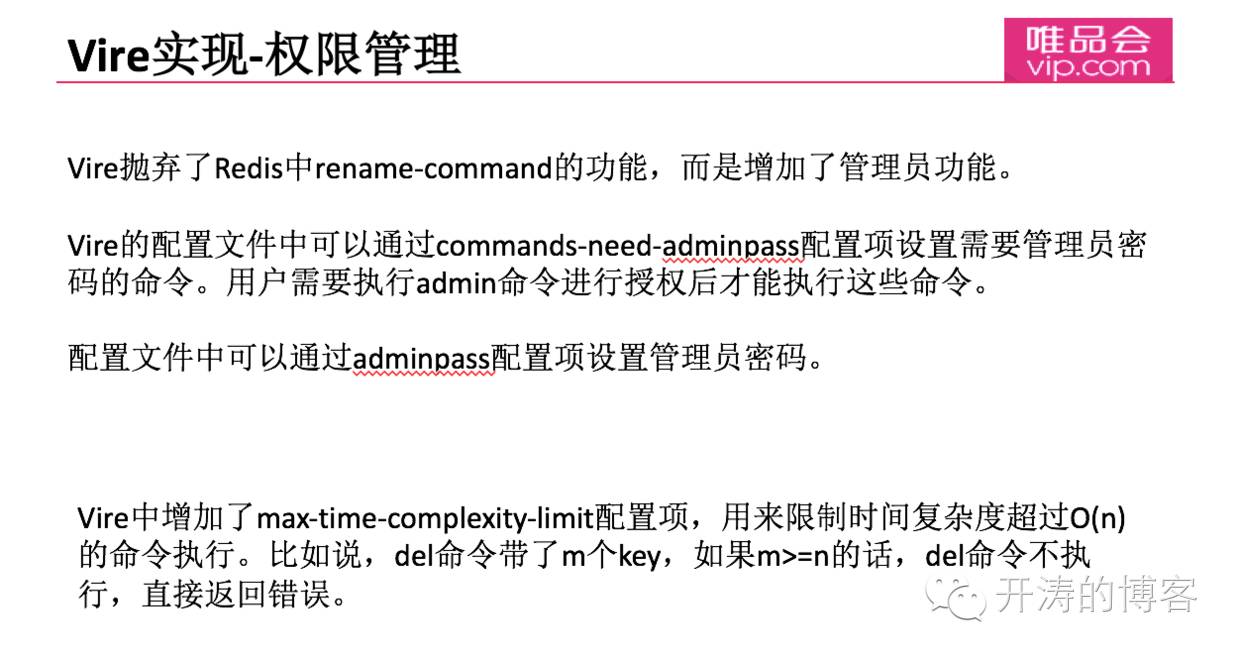
Worker线程专注于处理客户端的请求，杂活累活有backend线程来做，backend线程在vire后续版本中，会发挥更重要的作用。



这里是Vire代码内部对object的处理，这里会有些性能退化。

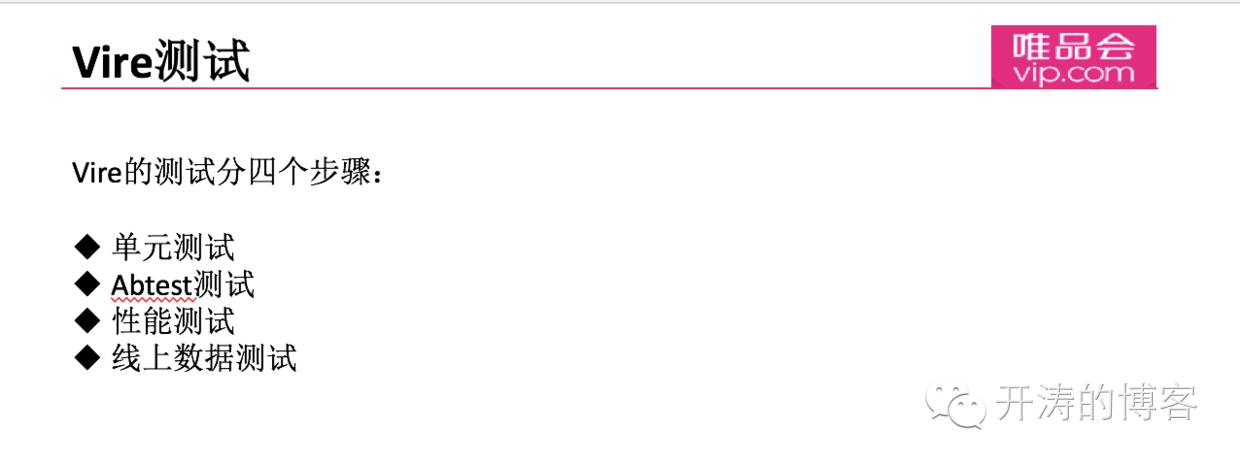


这是vire对多key命令的一些特殊处理，死锁的问题，导致个别redis命令在vire中暂时无法实现。



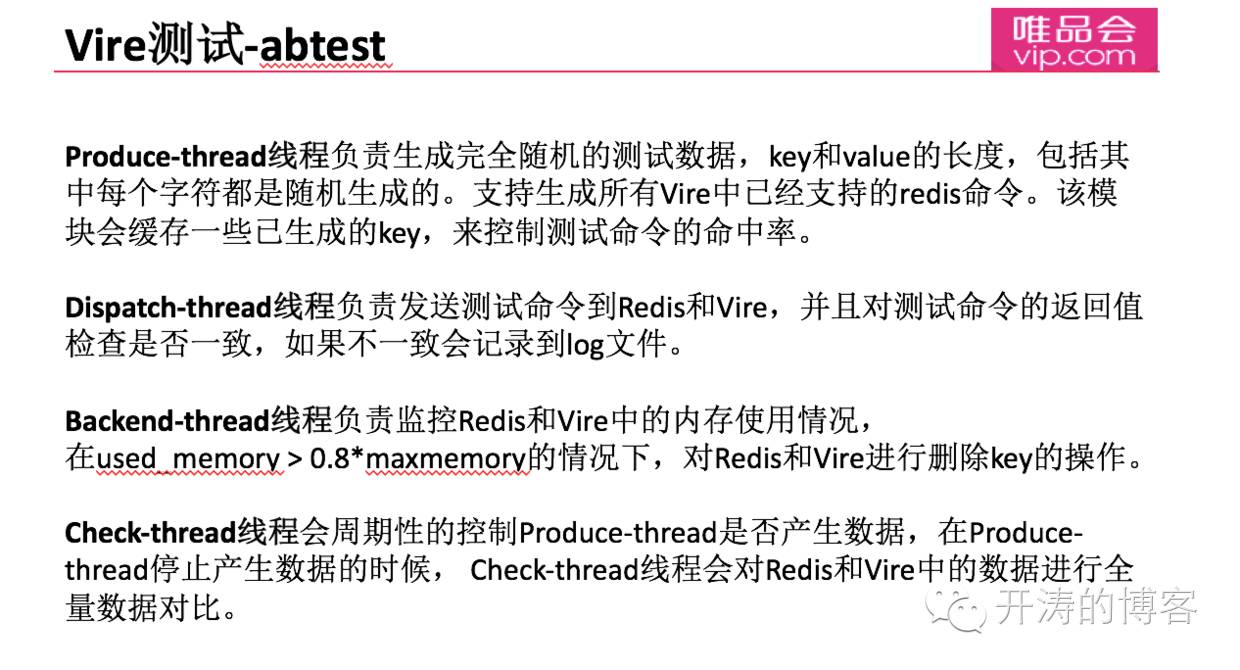
Vire中增加了一些权限管理，vire增加了管理员的角色，保证了一些危险命令不被开发执行。

下面说说vire的测试：

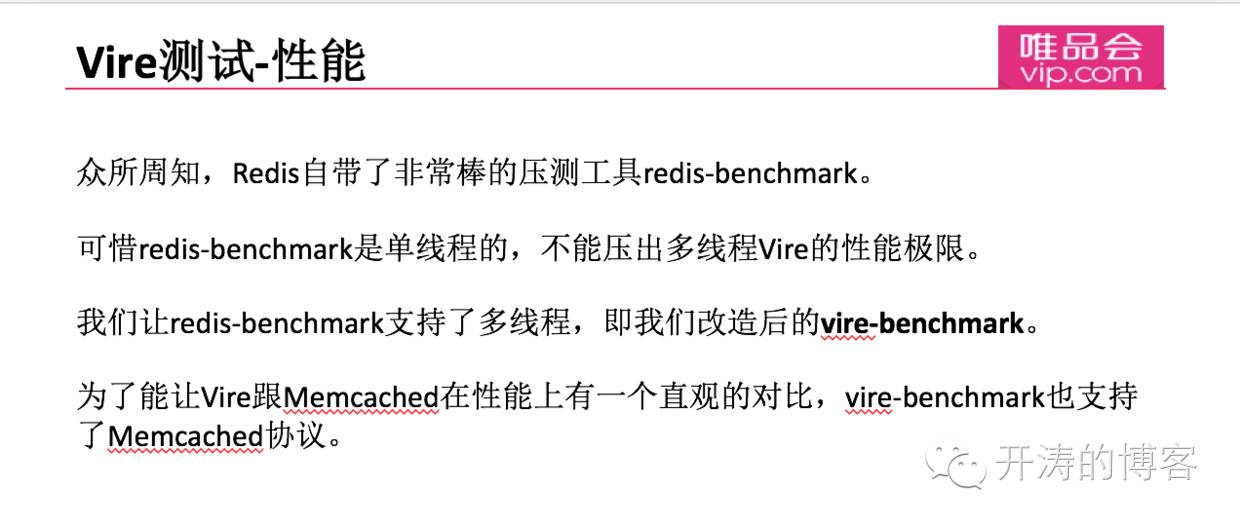


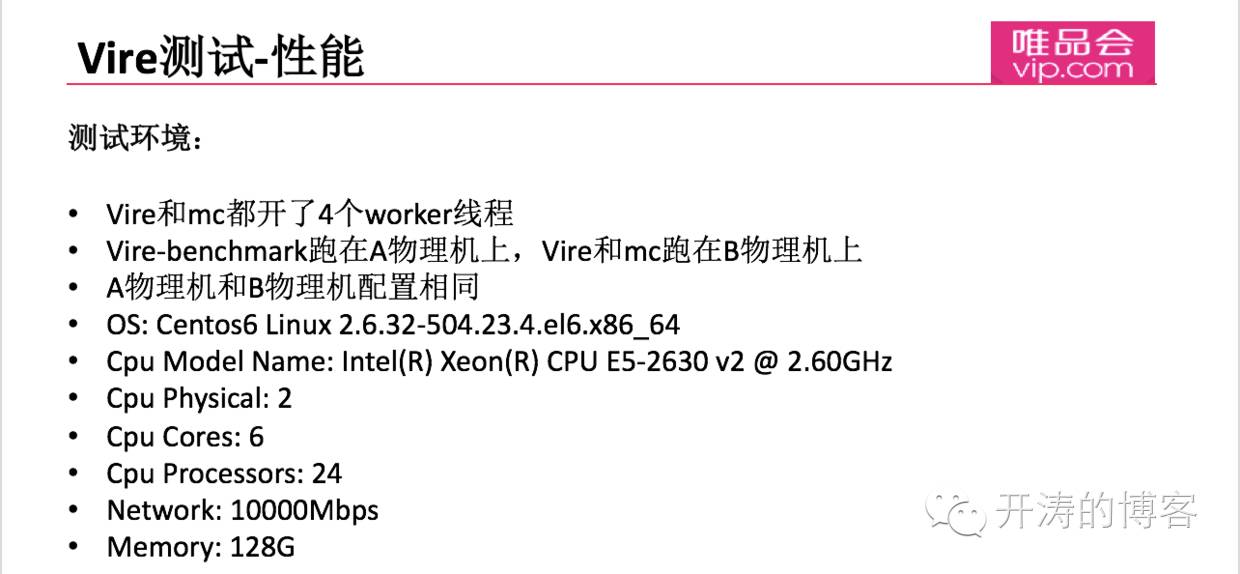
这里着重说说abtest和性能测试， 为了保证vire的命令执行起来与redis一模一样，我们开发了abtest测试框架。详细说说abtest框架中各模块的作用：

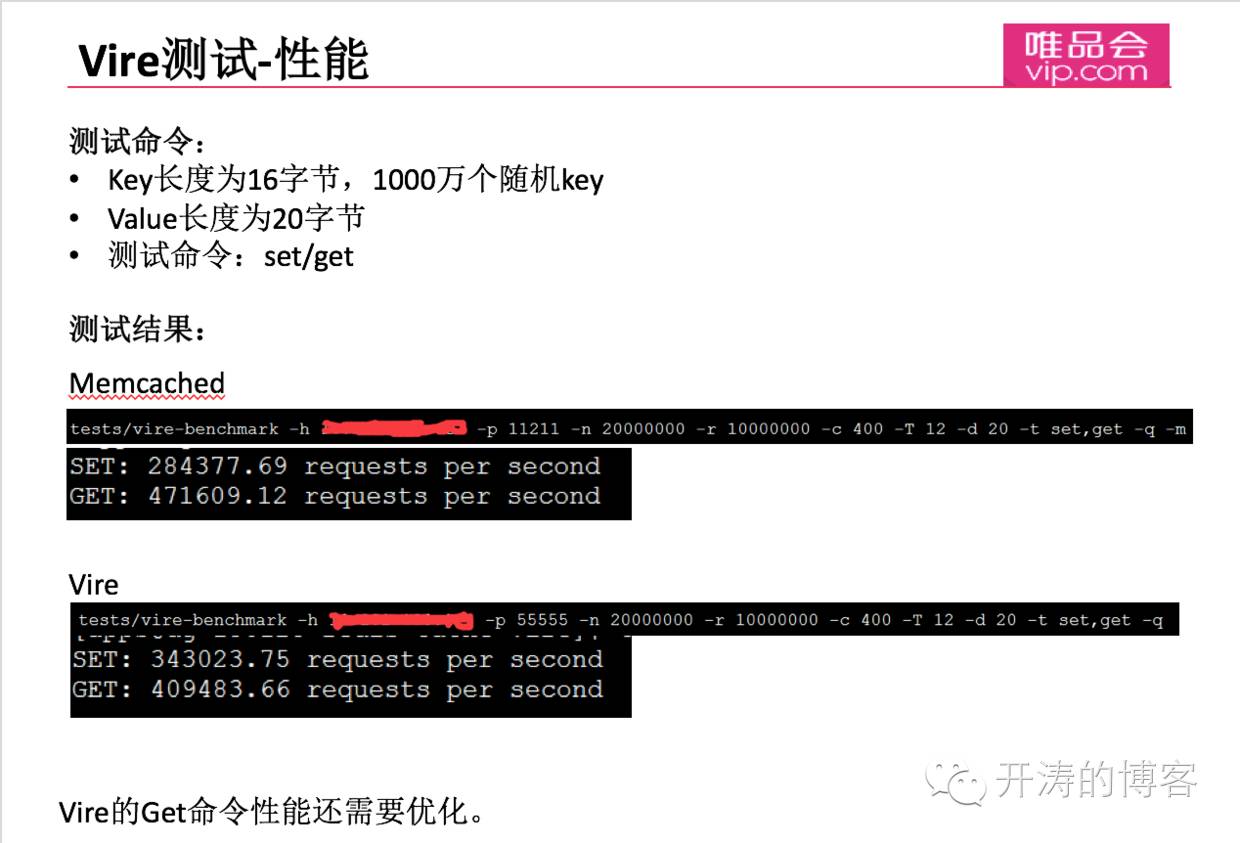




这个测试框架有效的帮我们发现了一些bug，以下是性能测试：







我们的目标就是性能接近或跟mc一样，以下是hotkey测试：



hotkey的效果还不错。

## ****Q&A****

1. Q：客户端需要换吗？

A：客户端兼容，无需更换，使用起来跟原生redis一样

1. Q：worker和db的关系是什么？

A：worker和db没有关系, client是数据worker线程的, DB是完全独立的

1. Q：后期主备会支持吗？

A：以后会支持主备，集群和脚本等高级功能

1. Q：有没有想过把锁降低至key级别?

A：没必要key级别的锁

1. Q：死锁问题为何不通过顺序锁定相关db来解决呢，我们的redis是分布式锁，通过按统一的顺序锁定，就可以避免死锁

A：锁的数量会太多，你说的这个死锁问题很好，有这样的想法，但还没有时间去验证可不可行，以后可以尝试。

1. Q：vire和redis-cluster比起来哪个性能更好？

A：redis-cluster是集群模式，vire是单实例，没办法比较性能，vire最后一个版本希望能支持到集群

1. Q：给我的理解vire的多个逻辑db的设计原理和redis-cluster里多个分片原理是一样吧？

A：非常类似, 只不过redis-cluster里的slot是海量的，16384

1. Q：现在redis-cluster的解决方案是客户端自己计算slot的位置，可以通过根据操作的读写类型，实现负载均衡，vire采取的多db+多worker的方案，他这样的优势在哪里？

A：主要是提升单个实例的qps能力

1. Q现在的设计是全部基于内存上的？服务器宕机是不是数据全都会消失

A: vire0.1.0版本数据全部在内存，只适合于做缓存, vire后续版本会做持久化和复制，甚至是集群

来自：http://mp.weixin.qq.com/s?\_\_biz=MzIwODA4NjMwNA==&mid=2652897994&idx=1&sn=92748c118550e160abb65e0f25a76031&chk