Redis Presharding

在解释Redis Presharding之前，先来说下sharding。Sharding，译为分片，有时也用partitioning( 分区 )，这俩是同一个意思，说的是数据分散到Redis多个实例。

Presharding是Redis 3.0之前的概念。Redis 3.0官方支持了Redis集群，在那之前，要想实现集群，用的是各种民间方法，如客户端侧的分片、代理等，典型的代表是Twitter的Twemproxy。同时，Redis作者也提出了一种集群方案，即为presharding。他戏称之为乞丐版的集群。

如何分片，大家可能首先会想到如下的分片方案：Node = Hash(key) mod N。hash是简单的，但rehash是困难的，同时涉及到redis运行时key在不同实例间的迁移。

Redis Presharding不需要resharding，主要步骤如下：

1. 在项目的开始阶段，在同一服务器上部署多个redis实例；
2. 在扩容阶段，新增服务器，部署新的redis实例；
3. 将新实例设为要迁移的旧实例的slave;
4. 断开客户端；
5. 更新配置：新实例的ip;
6. 新实例执行slaveof no one，成为master;
7. 重启客户端，使客户端使用新实例；
8. 关闭旧实例。

作者提出这种方案，也是基于Redis轻量的特点。因为Redis代码量少，三方依赖少，因此内存占用相当少（1M+），故而可以在单机上部署相当多的实例。

使用presharding时，可以部署脚本来管理集群、增加监控、通过主从复制增强容错能力。

# 重点说明：

1. presharding前后，集群中的Redis实例总数是不变的。正因为如此，避免了resharding，不存在key值从一个区转移到另一个分片区。
2. 新增服务器上应部署多少个Redis实例？

答：如果之前有1台服务器，加1台服务器，那么应迁移1/2的实例，以此类推。原则是每台服务器上实例数均衡。

1. presharding前后Redis实例不变，那有什么意义？

答：集群的思想是用多台（可能性能相对差的）机器，得到高效的处理能力。随着业务的发展，单机上可能存储、CPU、内存不够用了，此时就需要增加机器，并进行实例的迁移，此时单机的实例数变少，单机的压力减小。因此，假设有N台机器，每台M个实例，随着业务的发展，N会逐渐增大，M逐渐减小，但N\*M是保持不变的。

1. 限制。该方案受限于初始设定的实例数，如果N\*M个Redis实例还满足不了需求，再增加机器就会出现resharding。作者建议初始单机部署32、64、128个Redis实例，当然最终还是应该根据自己的业务实际情况和对未来的预估，或者直接升级redis版本。