链滴

Codis 的设计与实现 Part 2

作者：unhappydepig
原文链接：https://ld246.com/article/1462845419433
来源网站：链滴

<h4 id="proxy-"> </h4>
在 Codis 的设计中， Proxy 被设计成无状态的，客户端连接任何一个 Proxy 都是一样的。而且个 Proxy 启动的时候，会在 Zookeeper 上注册一个临时节点， 所以客户端甚至可以根据这个特性实 HA （其实我在豌豆荚内部就写了一个基于 Jedis 的 Codis HA RoundRobinPool）
当然，这个设计带来的好处是，请求可以被负载均衡，而且在整个系统中不会出现单点。 但是问题来了，由于 Codis 是动态扩缩容的功能的， 当 Codis 在进行数据迁移的过程中，如何保证任意个 Proxy 都不会读到老的或者错误的数据？
解释这个问题之前，我想先介绍一下 Codis 的数据存储方式和关于数据迁移的一些前置知识：</>

数据被根据key，分布在 1024 个 slot 中， slot 是一个虚拟的概念，数据存储在实际的多个 codi-server （codis 修改版的 redis-server） 中，每个 codis-server 负责一部分key-value数据， 哈希法是 crc32(key) % 1024
数据迁移是由 codis-config 发起的，在 codis-config 看来，数据迁移的最小单位是 slot
对于 codis-server 来说，没有任何分布式逻辑在其中， 只是实现了几个关于数据传输的指令： sotsmgrtone, slotsmgrt…. 其主要的作用是：随机选取特定 slot 中的一个 key-value pair, 传给另外一个 codis-server, 传输成功后，把本地的这个 key-value pair 删除， 注意， 这个整个操作原子的。

所以，这就决定了 Codis 并不太适合 key 少，但是 value 特别大的应用， 而且你的 key 越少，alue 越大，最后就会退化成单个 redis 的模型 （性能还不如 raw redis），所以 Codis 更适合海量 Ky, value比较小 （<= 1 MB) 的应用。
为什么 codis-server 的数据迁移是一个个key的，而不是类似很多其他分布式系统，采用 replicaton 的方式？ 我认为，对于 redis 这种系统来说，实现 replication 不太经济， 首先，你需要 rdb dup 吧？ 在 redis 里面所有的操作严格来说都是串行的（单线程模型决定），所以dump数据是需要 for 一个新进程的， 否则如果直接 SAVE 会阻塞唯一的主线程，同时还得考虑dump过程和传输过程中生的新数据的同步的问题， 实现起来比较复杂。所以我们每次只原子的迁走一个 key，不会把主线程lock 住， redis 操作的是内存，批量的一次性写入和分多次set几乎没有区别（对于单机而言）， 而这个模型还避免了迁移过程中的数据更新同步的问题，因为由于迁移一个 key 的操作是原子的， 对这个 redis-server 来说， 在完成这次迁移指令之前，是不会响应其他请求的。 所以保证了数据的安。
一次典型的迁移流程：

codis-config 发起迁移指令如 pre_migrate slot_1 to group 2

codis-config 等待所有的 proxy 回复收到迁移指令, 如果某台 proxy 没有响应, 则标记其下线 (于proxy启动时会在zk上注册一个临时节点, 如果这个proxy挂了, 正常来说, 这个临时节点也会删除, 在odis-config发现无响应后, codis-config会等待30s, 等待其下线, 如果还没下线或者仍然没有响应, 则cdis-config 将不会释放锁, 通知管理员出问题了) 相当于一个2阶段提交

codis-config 标记slot_1的状态为 migrate, 服务该slot的server group改为group2, 同时codis-onfig向group1的redis机器不断发送 SLOTSMGRT 命令, target参数是group2的机器, 直到group1没有剩余的属于slot_1的key

迁移过程中, 如果客户端请求 slot_1 的 key 数据, proxy 会将请求转发到group2上, proxy会先在roup1上强行执行一次 MIGRATE key 将这个键值提前迁移过来. 然后再到group2上正常读取

迁移完成, 标记slot_1状态为online

关键点：

所有的操作命令，都通过 Zookeeper 中转，所有的路由表，都放置在 ZooKeeper 中，确保任一个 proxy 的视图都是一样的。

codis-config 在实际修改slot状态之前，会确保所有的 proxy 收到这个迁移请求。

在客户端读取正在迁移的slot内的数据之前， 会强制在源redis是执行一下迁移这个key的操作。

这两点保证了，proxy 在读取数据的时候，总是能在迁移的目标机上命中这个 key。
这就是 Codis 如何进行安全的数据迁移的过程。
转自:http://0xffff.me/blog/2014/11/11/codis-de-she-ji-yu-shi-xian-part-2/