## Redis集群方案

Redis集群主要了解的有三种实现机制，分别是：客户端分片，代理分片，Redis Cluster

1. **客户端分片**

这种做法是将分片工作放在业务程序端，程序根据预先设计的路由规则，直接对多个Redis实例进行分布式访问。这样的好处在于不依赖第三方分布式中间件，实现方法和代码都自己掌控，可随时调整。但这实际上是一个静态分片技术，Redis实例的增减都得手工调整程序，较为少用。

优势在于中间少了一个分发环节，性能比代理分片高。但缺点是升级麻烦，对研发人员的依赖性高，出现故障时，定位和解决较为麻烦。

1. **代理分片**

将分片工作交给专门的代理程序来做。代理程序接收到来自业务程序的数据请求，根据路由规则，将这些请求分发给正确的Redis实例并返回给业务程序。

这种机制下，一般会选择第三方代理程序。这样的好处在于业务程序不用关系，业务层不用关系后端Redis实例，运维起来较为方便，虽然会带来一部分性能损耗，但是对于Redis这种内存读写型应用，相对影响不大

1. **Redis Cluster**

这个机制是在redis3.0以后的版本才支持的，在这种机制下，没有中心节点，每个节点保存数据和整个集群状态，每个节点都和其他所有节点连接。

Redis Cluster将所有key映射到16381个slot中，集群中每个Redis实例负责一部分，业务程序通过集成的Redis客户端进行操作。客户端可以向任一实例发出请求，如果所需数据不在该实例中，则该实例引导客户端自动重定向去对应实例读写数据。

Redis Cluster的成员管理（节点名称、IP、端口、状态、角色）等，都通过节点之间两两通讯，定期交换并更新。

目前主要考虑使用的是第二或者第三种

目前第三种方式了解的较多，如果采用第三种计划使用的是第三方库rediscluster，考虑使用第三种主要在于对节点的管理较为简单（有专门的工具可以用来增删节点，分配slots），且较为易用，有官方提供的文档，故障排查也能相对较快

第二种方式，可控性较高,但实现较为复杂（目前还没有考虑过如何使用）