Redis 主从复制实现原理

Redis主从配置比较简单，基本上就是在从节点配置文件加上：slaveof 192.168.33.130 6379

主要实现原理：

通过master server持久化的rdb文件实现的。Master server 先dump出内存快照文件，然后将rdb文件传给slave server，slave server根据rdb文件重建内存表。

redis复制过程如下：  
1、slave server启动连接到master server之后，salve server主动发送SYNC命令给master server  
2、master server接受SYNC命令之后，判断，是否有正在进行内存快照的子进程，如果有，则等待其结束，否则，fork一个子进程，子进程把内存数据保存为文件，并发送给slave server  
3、master server子进程进程做数据快照时，父进程可以继续接收client端请求写数据，此时，父进程把新写入的数据放到待发送缓存队列中  
4、slave server 接收内存快照文件之后，清空内存数据，根据接收的快照文件，重建内存表数据结构  
5、master server把快照文件发送完毕之后，发送缓存队列中保存的子进程快照期间改变的数据给slave server，slave server做相同处理，保存数据一致性  
6、master server 后续接收的数据，都会通过步骤1建立的连接，把数据发送到slave server  
需要注意：slave server如果因为网络或其他原因断与master server的连接，当slave server重新连接时，需要重新获取master server的内存快照文件，slave server的数据会自动全部清空，然后再重新建立内存表，这样会让slave server 启动恢复服务比较慢，同时也给master server带来较大压力，可以看出redis的复制没有增量复制的概念，这是redis主从复制的一个主要弊端，在实际环境中，尽量规避中途增加从库  
redis2.8之前不支持增量，到2.8之后就支持增量了！

**redis cluster（集群）**  
redis集群配置参考我的博文：[http://blog.csdn.net/nuli888/article/details/52134117](http://blog.csdn.net/nuli888/article/details/52134117" \t "_blank)  
3.0之后的功能，至少需要3(Master)+3(Slave)才能建立集群，是无中心的分布式存储架构，可以在多个节点之间进行数据共享，解决了Redis高可用、可扩展等问题。

[**Redis**](http://lib.csdn.net/base/redis)集群提供了以下两个好处  
1、将数据自动切分(split)到多个节点  
2、当集群中的某一个节点故障时，**[redis](http://lib.csdn.net/base/redis" \o "Redis知识库" \t "_blank)**还可以继续处理客户端的请求。  
一个 redis 集群包含 16384 个哈希槽（hash slot），**[数据库](http://lib.csdn.net/base/mysql" \o "MySQL知识库" \t "_blank)**中的每个数据都属于这16384个哈希槽中的一个。集群使用公式 CRC16(key) % 16384 来计算键 key 属于哪个槽。集群中的每一个节点负责处理一部分哈希槽。  
集群中的主从复制  
集群中的每个节点都有1个至N个复制品，其中一个为主节点，其余的为从节点，如果主节点下线了，集群就会把这个主节点的一个从节点设置为新的主节点，继续工作。这样集群就不会因为一个主节点的下线而无法正常工作

**注意：**

1、如果某一个主节点和他所有的从节点都下线的话，redis集群就会停止工作了。redis集群不保证数据的强一致性，在特定的情况下，redis集群会丢失已经被执行过的写命令  
2、使用异步复制（asynchronous replication）是redis 集群可能会丢失写命令的其中一个原因，有时候由于网络原因，如果网络断开时间太长，redis集群就会启用新的主节点，之前发给主节点的数据就会丢失。