关于主从复制

和MySQL的主从一样，这种结构主要是为了数据冗余和提示性能。Redis的主从同步是异步进行的，所以并不会影响主的处理性能。在数据持久化方面，可以把这个任务交给从服务器来做，这样可以减小主服务器的负担。

在主从结构中，一般把从服务器设置为只读模式，这样也是为了更好的保持数据一致性。

原理（参考《Redis设计与实现》）

Redis主从同步有两种方式，一种是全同步和部分同步，当然执行哪种同步是自动判断的无需人工干预。

Redis的复制功能分为同步（SYNC）和命令传播（COMMAND PROPAGATE）两个操作：

- 同步操作：将从服务器的数据库状态更新至主服务器当前所处的状态
- 命令传播：当主服务器数据被修改后，通过命令传播使主从数据库状态一致

旧版（Redis 2.8以前）的复制功能实现

上面提到同步有两种方式全同步和部分同步，在旧版中只有全同步没有部分同步。同步过程如下：

- 从服务器向主服务器发送SYNC命令
- 主服务器收到SYNC命令后执行BGSAVE命令，在后台生成一个RDB文件，并使用缓冲区记录从现在（当BGSAVE执行的那一刻）开始执行的所有写命令。
- 当主服务器使用BGSAVE命令执行完毕后，主服务器会将RDB文件发送给从服务器，从服务器载入这个RDB文件，将自己的数据库状态更新至主服务器执行BGSAVE那一刻的状态。
- 主服务器将缓冲区里面记录的所有写命令发送给从服务器，从服务器执行这些命令，从而使主从数据库状态一致。

同步完成后，主服务器的数据也会随着后期的写操作而变化，所以就需要通过命令传播功能把在主服务器上执行的写命令传递给从服务器来实现主从数据库状态一致。

旧版复制功能的缺陷：

旧版复制功能的缺陷：

- 初次复制：从服务器以前没有同步过任何主服务器，或者从服务器当前要同步的主服务器和上次同步的不同
- 断线后复制：处在同步过程中，但因其他原因导致网络中断，进而造成复制中断，再重新恢复网络连通后的复制。

对于初次复制整个过程没有问题，但是断线后复制则会执行全同步，这样效率比较低。因为SYNC过程很消耗资源，生成RDB会消耗CPU、内存和磁盘IO资源，发送RDB的过程会占用网络带宽，从服务器载入RDB时会阻塞服务器从而无法处理请求。

新版（Redis 2.8及以后）的复制功能实现

在新版中使用了PSYNC来代替SYNC执行同步任务。PSYNC有两种模式也就是上面提到的全同步和部分同步：

- 全同步用于初次同步，执行过程和SYNC一致
- 部分同步则重点在于同步断线期间（一定期间而非无限期间）的内容，也就是说在规定期间内执行增量部分的同步，如果超过了规定期间则执行全同步。

部分同步的三个重要概念：

- 主服务器的复制偏移量和从服务器的复制偏移量
- 服务器的复制积压缓冲区
- 服务器的运行ID

复制偏移量：主服务器每次给从服务器传递N个字节的数据时，就将自己的复制偏移量增加N，同时从服务器收到N个字节以后，也会再自己的复制偏移量上加N，从而保持主从的复制偏移量一样，这样就可以判断主从的数据库状态是否一致。

复制积压缓冲区：改缓冲区是一个定长的先进先出的队列，默认是1MB。主服务器把命令传递给从服务器的同时也会向自己的积压缓冲区队列写入这个命令。所以这个缓冲区里面保存了一部分最近的写命令，并且缓冲区会为队列的每个字节记录偏移量。当从服务器断线重新连接后，会发送PSYNC命令，同时包含自己的复制偏移量，主服务器根据这个复制偏移量来决定用何种同步方式。如果该偏移量之后的数据（偏移量+1开始的数据）仍然在复制积压缓冲区内，那么执行部分同步，否则执行全同步。

服务器的运行ID：每个Redis服务器都有一个唯一ID标识符，该ID是自动生成的一个40位十六进制字符。当初次复制时，主从服务器会保存彼此的ID，当断线后，从服务器除了向主服务器提供复制偏移量以外还需提供从服务器保存的主服务器的ID，当主服务器接受到ID后，先检查是否和自己的一样，如果一样表示从服务器请求的是一个

断线后同步，再根据偏移量来决定执行全同步还是部分同步；如果该ID不是自己的ID，则表示是一个初次同步，则执行全同步。

在新版中的命令传播，**从服务器会以默认每秒的频率向主服务器发送REPLCONF_ACK**，该任务会传递从服务器的复制偏移量，同时也是为了进行心跳检测。如果主服务器超过1秒都没有收到从服务器的REPLCONF_ACK，则主服务器会认为连接出现了中断。

在主服务器的控制台输入INFOreplication可以在lag一栏看到从服务器最后一次向主服务器发送REPLCONF_ACK命令距离限制过了多少**秒**，一般该值为0或者1，超过1秒则表示有问题，如下图：