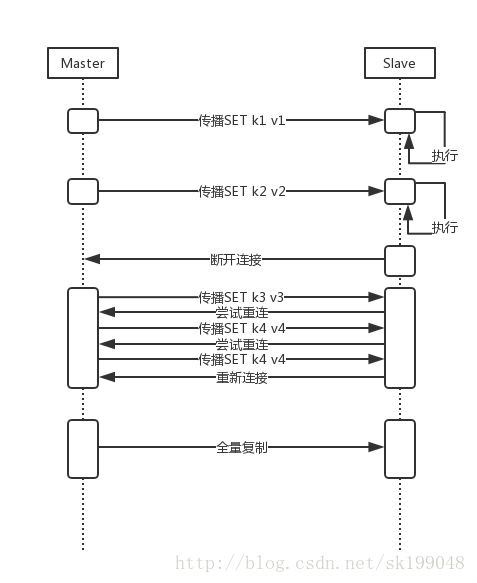
<http://blog.csdn.net/sk199048/article/details/77922589>

之前写过一篇博客（<http://blog.csdn.net/sk199048/article/details/50725369>）来介绍了Redis主从同步的过程，里面主要介绍从服务器使用SYNC命令复制数据的过程。Reids复制数据主要有2种场景：   
1. 从服务器从来第一次和当前主服务器连接，即初次复制   
2. 从服务器断线后重新和之前连接的主服务器恢复连接，即断线后重复制   
  对于初次复制来说使用SYNC命令进行全量复制是合适高效的，但是对于**断线后重复制**来说效率就不太能接受了。举例来说：   
   
  如图所示，Master在断开连接期间只传播了3个写入命令，但是重新连接之后却要全量复制，显然这是低效并且不太必要的。

### PSYNC概念

  为了应对这种情况，Redis在2.8版本提供了PSYNC命令来带代替SYNC命令，为Redis主从复制提供了部分复制的能力。PSYNC命令格式是：

PSYNC <runid> <offset>

runid:主服务器ID

offset:从服务器最后接收命令的偏移量

* 1
* 2
* 3

  PSYNC执行过程中比较重要的概念有3个：runid、offset（复制偏移量）以及复制积压缓冲区。

#### runid

  每个Redis服务器都会有一个表明自己身份的ID。在PSYNC中发送的这个ID是指之前连接的Master的ID，如果没保存这个ID，PSYNC的命令会使用”PSYNC ? -1” 这种形式发送给Master，表示需要全量复制。

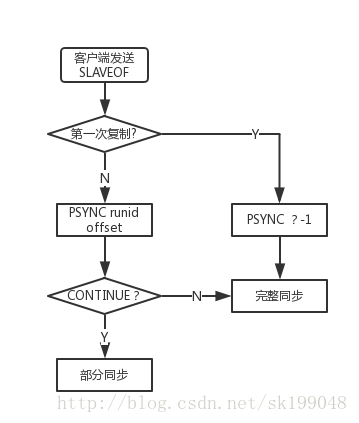
#### offset（复制偏移量）

  在主从复制的Master和Slave双方都会各自维持一个offset。Master成功发送N个字节的命令后会将Master的offset加上N，Slave在接收到N个字节命令后同样会将Slave的offset增加N。Master和Slave如果状态是一致的那么它的的offset也应该是一致的。

#### 复制积压缓冲区

  复制积压缓冲区是由Master维护的一个固定长度的FIFO队列，它的作用是缓存已经传播出去的命令。当Master进行命令传播时，不仅将命令发送给所有Slave，还会将命令写入到复制积压缓冲区里面。

### PSYNC执行过程

  理解了上面三个基本概念，PSYNC的执行过程就好理解了。   
   
  1 客户端向服务器发送SLAVEOF命令，让当前服务器成为Slave；   
  2 当前服务器根据自己是否保存Master runid来判断是否是第一次复制，如果是第一次同步则跳转到3，否则跳转到4；   
  3 向Master发送PSYNC ? -1 命令来进行完整同步；   
  4 向Master发送PSYNC runid offset；   
  5 Master接收到PSYNC 命令后首先判断runid是否和本机的id一致，如果一致则会再次判断offset偏移量和本机的偏移量相差有没有超过复制积压缓冲区大小，如果没有那么就给Slave发送CONTINUE，此时Slave只需要等待Master传回失去连接期间丢失的命令；   
  6 如果runid和本机id不一致或者双方offset差距超过了复制积压缓冲区大小，那么就会返回FULLRESYNC runid offset，Slave将runid保存起来，并进行完整同步。

### 后续

  上面内容大多数是《Redis设计与实现》这本书中的内容，接下来会仔细看下这部分实现的源码，将实现细节理出来。

### 引用

《Redis设计与实现》黄健宏