

大数据库系统

3.7 Redis集群配置，主从复制

3.6 Redis集群配置，主从复制

◆ 主要内容

3.6.1 Redis集群简介

3.6.2 Redis主从复制功能

3.6.1 Redis集群简介及配置

◆ Redis集群

在实际应用中，使用单台Redis服务器是很危险的

- 服务器故障停机
- 容量瓶颈
- QPS（每秒查询率）瓶颈等

影响系统及网站的正常服务，从而造成不必要的经济损失

3.6.1 Redis集群简介及配置

◆ Redis集群（Master-Slave集群）的作用

1、主从备份，防止主机宕机造成数据丢失

- 主机宕机，轻松切换从服务器继续工作

2、读写分离，分担master的压力

- 减轻负担

3、任务分离，如从服务器分担备份工作与计算工作

- 有些任务计算比较耗资源，比如大量浮点运算等可以交给更合适的高配置从服务器

3.6.1 Redis集群简介及配置

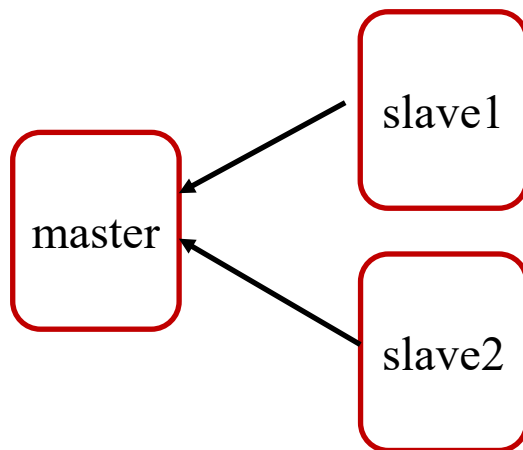
◆ Redis集群

- 一个master可以有多个slave
- 一个slave只能有一个master
- 主服务器可以有从服务器，从服务器也可以有自己的从服务器，多个服务器之间可以构成一个网状结构，它们之间具有传递关系

3.6.1 Redis集群简介及配置

◆ Redis集群结构

第一种：星型结构



这是常见的一主多从的形式

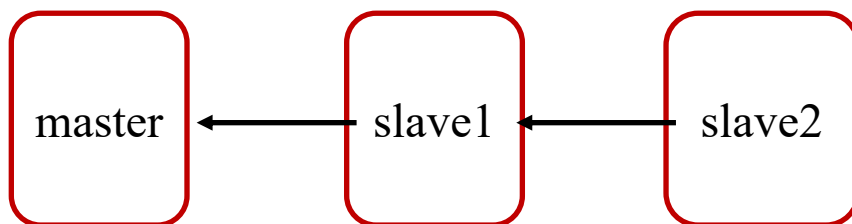
假定master宕机，slave1变为master，slave2的master变为slave1

中心化较为严重，如果有多个slave，那主机宕机后，修改的slave数量也多

3.6.1 Redis集群简介及配置

◆ Redis集群结构

第二种：线型结构



从服务器也可以有自己的从服务器，一台从服务器只能有一台主服务器

假定master宕机，只要将slave1改成master就可以了

线型结构在master宕机后，比星型结构切换主从关系更简单

3.6.2 Redis主从复制功能

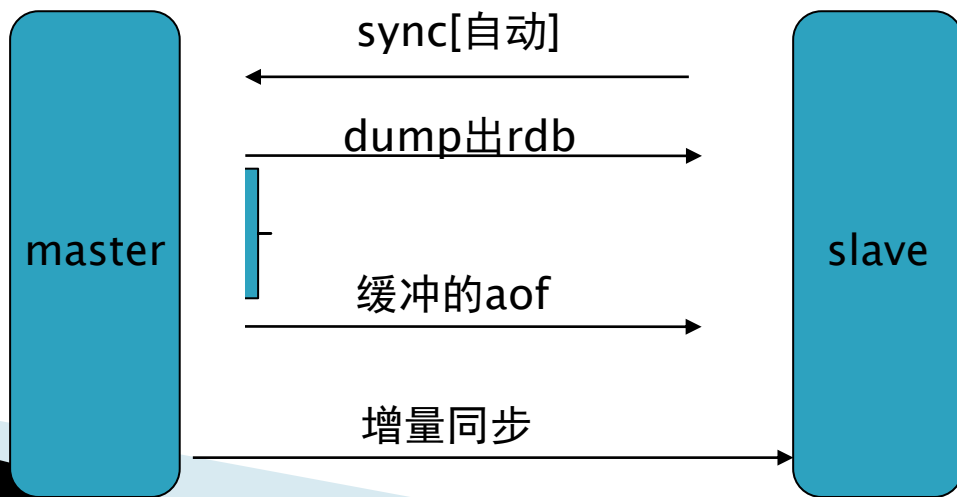
◆ Redis主从复制

- 在Redis中，通过执行REPLICAOF (SLAVEOF) 命令，或者通过配置文件设置REPLICAOF (SLAVEOF) 选项，就可以让一台服务器去复制另一台服务器
- 被复制的服务器叫主服务器 (Master)，而对主服务器进行复制的服务器叫从服务器 (Slave)
- 当主服务器中的数据更新后，根据配置和策略自动同步到从服务器上。其中，Master以写为主，Slave以读为主 (可以设置)

3.6.2 Redis主从复制功能

◆ Redis主从复制的过程

- 1、slave服务器启动后，连接master，发送一个sync命令开始同步
- 2、全量同步
- 3、增量同步



3.6.2 Redis主从复制功能

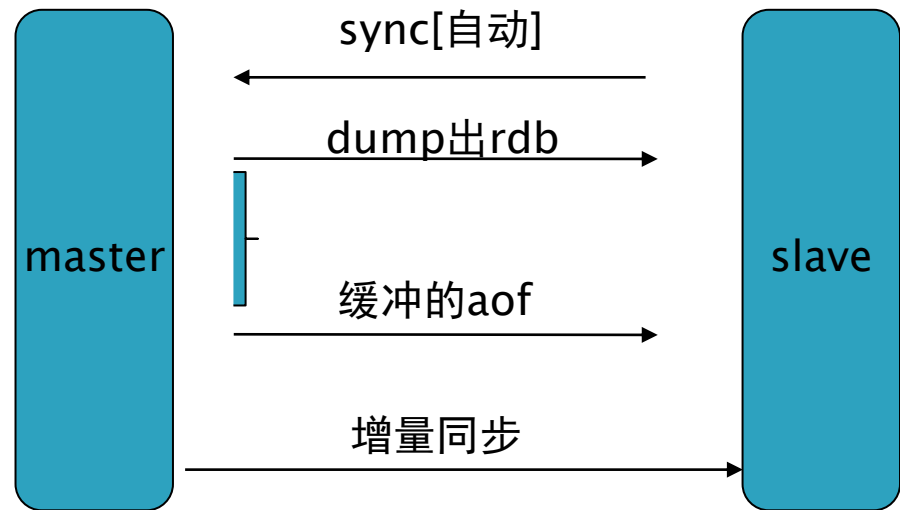
◆ Redis主从复制的过程

全量同步

- 一般发生在Slave初始化阶段，这时Slave需要将Master上的所有数据都复制一份

步骤如下：

- 1、主服务器接收到SYNC命令后，执行BGSAVE命令生成RDB文件，并向所有从服务器发送快照文件
- 2、从服务器接收RDB文件并载入到自己的内存
- 3、步骤1、2执行的过程中如果Master接收了新的命令，则这些新命令将会缓冲在AOF文件中，当Slave的RDB加载完成后，Master将缓冲后的AOF发给Slave，继续完成同步

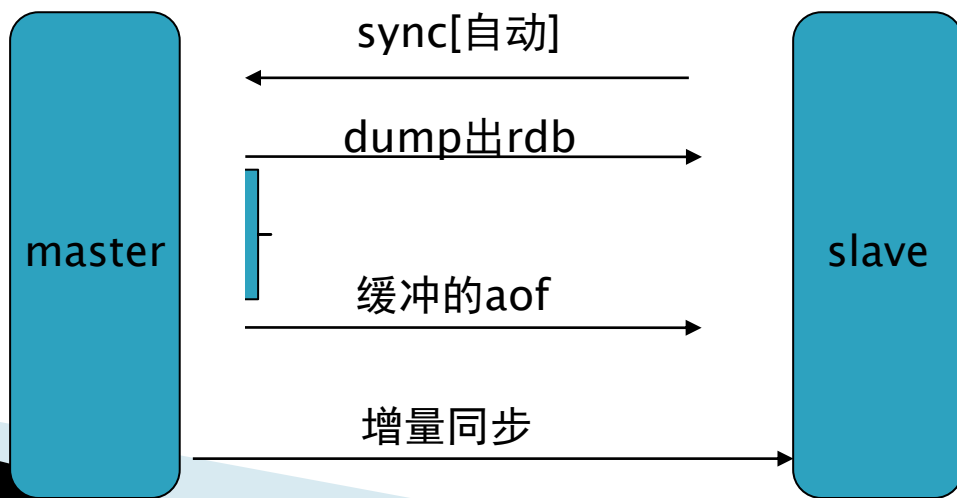


3.6.2 Redis主从复制功能

◆ Redis主从复制的过程

增量同步

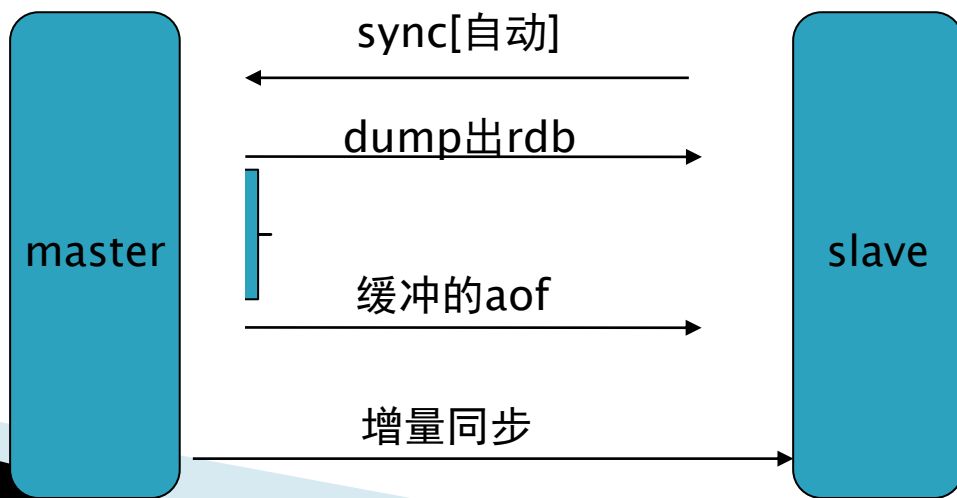
- Redis增量同步是指Slave初始化后开始正常工作时，主服务器发生的写操作同步到从服务器的过程
- 增量同步的过程主要是主服务器每执行一个写命令就会向从服务器发送相同的写命令，从服务器接收并执行收到的写命令



3.6.2 Redis主从复制功能

◆ Redis主从复制策略

- 主从刚刚连接的时候，进行全量同步；全量同步结束后，进行增量同步
- 如果有需要，slave在任何时候都可以发起全量同步
- Redis 策略是：无论如何，首先会尝试进行增量同步，如不成功，再要求进行全量同步



3.6.2 Redis主从复制功能

◆ Redis主从配置

Redis主从配置有两种方式

1、命令方式

在服务器中使用命令行直接设定主从关系

2、配置方式

在服务器开启前，修改Redis.conf文件的方式配置主从关系及参数

3.6.2 Redis主从复制功能

◆ `replicaof <masterip> <masterport>` (高版本)

◆ `slaveof <masterip> <masterport>` (低版本)

```
#       and resynchronize with them.  
#  
replicaof localhost 6379  
  
# If the master is password protected
```

<masterip>是master的IP地址，<masterport>是master的端口

- 该选项默认不开启，在进行主从复制配置时，在slave中需要开启，在master中不需要开启
- 假如master的IP 地址和端口分别为192.168.1.68:6379，那么slave的配置就应该是
`replicaof 192.168.1.68 6379`或`slaveof 192.168.1.68 6379`

3.6.2 Redis主从复制功能

◆ replica-read-only （高版本）

◆ slave-read-only （低版本）

```
# administrative / dangerous commands.  
replica-read-only yes  
  
# Replication SYNC strategy: disk or socket.
```

slave是否面向用户为只读

yes表示只读，no表示允许写入

- 如果该选项为no，则需要承担从服务器有可能与主服务器不一致的风险
- 如果该选项为yes时，用户往从服务器中写入数据，例如：set指令会报错

3.6.2 Redis主从复制功能

◆ requirepass

requirepass password

配置服务器的密码，客户端连接服务器时需要密码，password表示需要配置的密码内容

```
127.0.0.1:6379> requirepass password
127.0.0.1:6379> requirepass password
```

◆ AUTH命令

AUTH password

在成功设置密码之后，我们每次连接Redis服务器都需要使用AUTH命令来解锁密码，解锁成功之后才能使用Redis的其他命令

3.6.2 Redis主从复制功能

◆ masterauth <master-password>

Slave连接进Master的密码，在slave连接的时候进行认证，如果主服务设置了requirepass，则需要在从服务conf文件加上这条用于通过主服务密码验证

```
masterauth passwd
```

如果master设置了密码<master-password>，那么slave要连接到master上，就需要有master的密码

3.6.2 Redis主从复制功能

◆ replica-serve-stale-data （高版本）

◆ slave-serve-stale-data （低版本）

```
# COMMAND, POST, HOST: and LATENCY.  
#  
replica-serve-stale-data yes  
  
# You can configure a replica instance to
```

当slave与master主从复制过程中

- 值为yes时，表示服务器会继续处理来自客户端的命令请求
- 值为no时，表示服务器对除INFO和SLAVEOF命令之外的任何命令请求都会返回一个错误。

3.6.2 Redis主从复制功能

◆ repl-diskless-sync

是否使用socket方式复制RDB数据

Redis提供了两种全量同步时复制RDB的方式

1、disk方式（参数为no）

- master会创建一个新的进程，先把RDB文件保存到硬盘中，再把硬盘中的RDB文件传递给slave。在保存RDB文件的过程中，多个slave能共享这个RDB文件

2、socket方式（参数为yes）

- master会创建一个新的进程，直接把RDB文件以socket方式发送给slave，socket方式就是一个一个slave顺序复制

在硬盘读/写速度缓慢、网速快的情况下，推荐使用socket方式

3.6.2 Redis主从复制功能

◆ Redis主从配置

1、配置方式进行主从配置：

共3个服务器：

A：Master服务器：127.0.0.1:6379，对应的配置文件为Redis6379.conf，开启持久化

B：Master的Slave：127.0.0.1:6380，对应的配置文件为Redis6380.conf

C：Master的Slave：127.0.0.1:6381，对应的配置文件为Redis6381.conf

星形布局

开始配置

1、首先使用`pkill -9 Redis`关闭Redis，将6379的Redis.conf复制三份分别命名为Redis6379、Redis6380.conf和Redis6381.conf

```
hadoop@fisher-VirtualBox:/usr/local/redis$ cp redis.conf redis6379.conf
hadoop@fisher-VirtualBox:/usr/local/redis$ cp redis.conf redis6380.conf
hadoop@fisher-VirtualBox:/usr/local/redis$ cp redis.conf redis6381.conf
```

2、配置Redis6380.conf

```
[root@localhost redis]# vim redis6380.conf
```

修改如下几个配置项:

port 6380 #设置端口

daemonize yes #开启后台进程运行Redis

pidfile /var/run/Redis_6380.pid #设置进程ID文件

logfile "6380.log" #设置日志文件名

#save 900 1

#save 300 10

#save 60 10000

dbfilename dump6380.rdb #设置RDB文件名

Appendonly no

appendfilename "appendonly6380.aof"

replicaof localhost 6379 #将127.0.0.1:6380设置为127.0.0.1:6379的slave

replica-read-only yes #从服务器只读

3、配置Redis6381.conf

```
[root@localhost redis]# vim redis6381.conf
```

修改如下几个配置项:

port 6381 #设置端口

daemonize yes #开启后台进程运行Redis

pidfile /var/run/Redis_6381.pid #设置进程ID文件

logfile "6381.log" #设置日志文件名

#save 900 1

#save 300 10

#save 60 10000

#dbfilename dump6381.rdb #设置RDB文件名

appendonly no

replicaof localhost 6379 #127.0.0.1:6381同样是127.0.0.1:6379的slave

replica-read-only yes #从服务器只读

4、配置Redis6379.conf

修改如下几个配置项:

port 6379 #设置端口

daemonize yes #开启后台进程运行Redis

pidfile /var/run/Redis_6379.pid #设置进程ID文件

logfile "6379.log" #设置日志文件名

save 900 1 # 127.0.0.1:6379为主服务器, 开启备份

save 300 10

save 60 10000

dbfilename dump6379.rdb #设置RDB文件名

dir ./rdbaof

appendonly yes

appendfilename "appendonly6379.aof"

#replicaof localhost 6379 # 127.0.0.1:6379为主服务器, 注释掉

◆ 注意：

问：如果从服务器6380上开rdb，aof，而主服务器6379上不开rdb和aof，这种情况下主服务6379关机重启了以后，会同步从服务器6380上的rdb和aof上的数据吗？

答：不会的，由于主服务器6379上没开启aof和rdb，主服务器开机后会是个空白的数据库，更可怕的是这个空白的数据库会被同步到从服务器上，使得6380上已有的aof文件也被同步为空。实际应用中，主服务器是必须要开持久化，这样才可以达成所有开了持久化的从服务器的aof和rdb文件一致

- 5、配置完3台服务器了以后，打开这三台Redis服务器：分别打开三个终端，模拟3台服务器，分别打开Redis服务器

```
hadoop@fisher-VirtualBox:/usr/local/redis$ ./src/redis-server ./redis6379.conf  
hadoop@fisher-VirtualBox:/usr/local/redis$ ./src/redis-server ./redis6380.conf  
hadoop@fisher-VirtualBox:/usr/local/redis$ ./src/redis-server ./redis6381.conf
```

- 6、开启一个Redis客户端连接至6379服务器（master），添加一个title键值

```
hadoop@fisher-VirtualBox:/usr/local/redis$ ./src/redis-cli  
127.0.0.1:6379> set title sunshine  
OK
```

终端1

- 7、再开启一个Redis客户端连接至6380服务器

```
hadoop@fisher-VirtualBox:/usr/local/redis$ ./src/redis-cli -p 6380  
127.0.0.1:6380> keys *  
1) "title"  
127.0.0.1:6380> get title  
"sunshine"
```

终端2

- 8、在开启再开启一个Redis客户端连接至6381服务器

```
127.0.0.1:6381> get title  
"sunshine"
```

终端3

- 9、尝试在6381服务器设置键值title，出错了，为什么？

```
127.0.0.1:6381> set title snow  
(error) READONLY You can't write against a read only replica.  
127.0.0.1:6381>
```

设置了从服务器只读

3.6.2 Redis主从复制功能

◆ Redis主从配置

2、命令方式

SLAVEOF命令或REPLICAOF

不仅可以通过修改Redis.conf文件来设定服务器的主从关系，也可以直接使用命令来配置

SLAVEOF ip或REPLICAOF ip

配置主从复制，指定当前服务器作为指定ip服务器的从服务器

- 例：指定127.0.0.1:6380为127.0.0.1:6379的从服务器

```
127.0.0.1:6380>SLAVEOF 127.0.0.1 6379
```

或

```
127.0.0.1:6380> REPLICAOF 127.0.0.1 6379
```

3.6.2 Redis主从复制功能

◆ SLAVEOF no one命令或REPLICAOF no one

取消当前服务器与其他服务器之间的所有主从关系

例：取消127.0.0.1:6380与与其他服务器之间的所有主从关系

127.0.0.1:6380>SLAVEOF no one

或

127.0.0.1:6380>REPLICAOF no one

- 命令执行之后，从服务器与主服务器就会断开连接，从服务器就会变为主服务器
- 但之前的数据并不会丢失，只是在断开连接以后，数据不会再继续同步

3.6.2 Redis主从复制功能

◆ Redis主从配置

命令方式与配置方式的优缺点。

- 命令方式：
 - ✓ 优点在于不需要重启服务器，就能实现主从复制
 - ✓ 缺点是不便于管理
- 配置方式：
 - ✓ 优点在于可以进行统一配置，便于管理；
 - ✓ 缺点就是配置之后需要重启服务器，相关的配置才能生效

3.6.2 Redis主从复制功能

◆ 配置主服务器密码

为了增加安全性，可以在主从复制中配置主服务器密码，但注意，如果主服务器配置了密码，则需要修改从服务器的配置文件，否则从服务器无法连入主服务器

配置主服务器密码

- 1、关闭Redis服务器进程，进入127.0.0.1:6379服务器的Redis6379.conf中，找到“requirepass”，该配置项默认注释状态，注意查找

```
• warning: if you're running Redis on a publicly accessible network, you should
• look password option and add it to your config. This means that you should
• use a very strong password otherwise it will be very easy to break
•
requirepass passwd
•
• Command 1 completed
```

保存后退出，重启127.0.0.1:6379服务器

3.6.2 Redis主从复制功能

2、开启一个客户端连入6379服务器，尝试获取title键值

```
hadoop@fisher-VirtualBox:/usr/local/redis$ ./src/redis-server redis6379.conf
hadoop@fisher-VirtualBox:/usr/local/redis$ ./src/redis-cli
127.0.0.1:6379> get title
(error) NOAUTH Authentication required.
```

报错，提示不允许

3、此时可以输入命令“auth 密码”进行密码验证

```
127.0.0.1:6379> auth passwd
OK
```

4、再次尝试获取title键值

```
127.0.0.1:6379> get title
"sunshine"
127.0.0.1:6379>
```

成功获取title键值的value

3.6.2 Redis主从复制功能

由于主服务器6379设置了密码，因此从服务器需要修改配置

- 1、设置主服务器的键值title

```
127.0.0.1:6379> set title hothot  
OK
```

- 2、开启从服务器6380与6381

```
hadoop@fisher-VirtualBox:/usr/local/redis$ ./src/redis-server redis6380.conf  
hadoop@fisher-VirtualBox:/usr/local/redis$ ./src/redis-server redis6381.conf
```

- 3、开启两个客户端分别连接6380与6381，尝试获取键值title的内容

```
hadoop@fisher-VirtualBox:/usr/local/redis$ ./src/redis-cli -p 6381  
127.0.0.1:6381> get title  
"sunshine"
```

```
hadoop@fisher-VirtualBox:/usr/local/redis$ ./src/redis-cli -p 6380  
127.0.0.1:6380> get title  
"sunshine"
```

均返回旧值，发现与主服务器的内容不一样了

修改6380的配置文件Redis6380.conf，寻找masterauth配置项，再其之后加上主服务器密码，保存退出

```
masterauth passwd  
# When a replica loses its connection
```

同样修改6381的配置文件Redis6381.conf的masterauth配置项进行修改，保存退出

4、重启127.0.0.1:6379、127.0.0.1:6380、127.0.0.1:6381服务器

5、开启两个客户端分别连入127.0.0.1:6380、127.0.0.1:6381服务器，尝试查看title键值的内容

```
hadoop@fisher-VirtualBox:/usr/local/redis$ ./src/redis-cli -p 6380  
127.0.0.1:6380> get title  
"hothot"
```

```
hadoop@fisher-VirtualBox:/usr/local/redis$ ./src/redis-cli -p 6381  
127.0.0.1:6381> get title  
"hothot"
```

成功与主服务器同步

3.6.2 Redis主从复制功能

◆ Redis主从复制的缺陷

- 所有的写操作都是先在Master上操作，然后同步更新到Slave上
- 从Master同步到Slave机器有一定的延迟

缺陷1：当系统繁忙的时候，延迟问题会更加严重，Slave机器数量的增加也会使这个问题更加严重

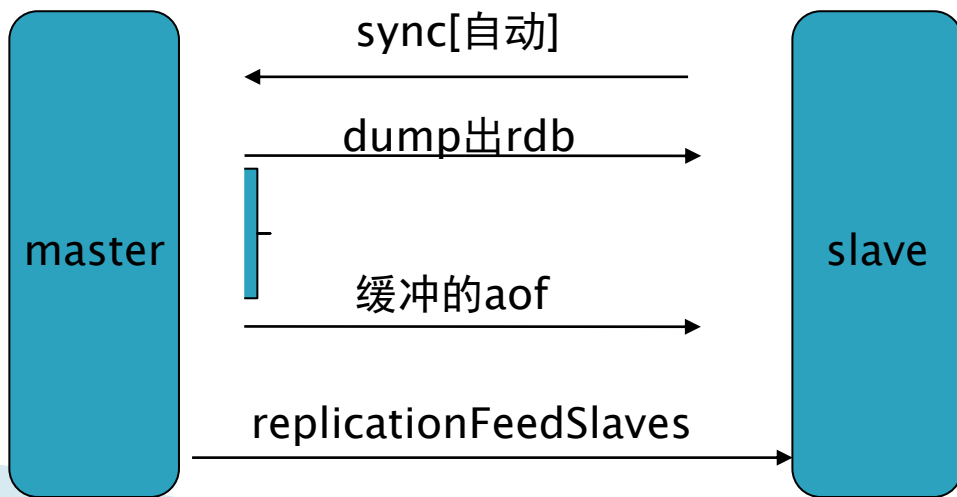
3.6.2 Redis主从复制功能

◆ Redis主从复制的缺陷

- 每次slave断开后 (无论是主动断开,还是网络故障), 再连接master都要master全部dump出来rdb, 再aof, 即同步的过程都要重新执行1遍

缺陷2: 如果多台slave同时启动master可能IO剧增

切记: 多台slave不要同时启动,
每隔几分钟启动一个



总结

◆ Redis集群

集群的作用

集群两种结构

星型特点

线型特点

◆ Redis主从复制

什么是Redis主从复制

Redis主从复制的过程

- 全量同步
- 增量同步

主从复制策略

总结

◆ Redis主从复制构造Redis集群

配置方式

- Slaveof
- slave-read-only
- requirepass
- masterauth
- slave-serve-stale-data
- repl-diskless-sync
- 通过配置项构造Redis集群

命令方式:

- AUTH
- SLAVEOF /SLAVEOF no one

◆ Redis主从复制的缺陷