Redis数据库主从复制

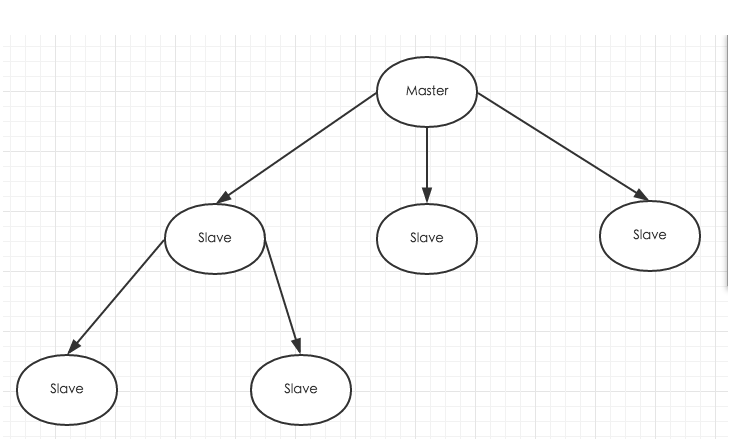
**一、主从复制的概述；**

**二、同步详解；**

**三、案例：配置主从复制；**

**一、主从复制的概述：**

**概述：**为了分担读写压力，Redis支持主从复制，Redis的主从结构可以采用一主多从或者级联结构，Redis主从复制可以根据是否是全量分为全量同步和增量同步。下图为级联结构。



**二、同步详解：**

**同步类型：全量同步、增量同步**

1.全量同步：

　　Redis全量复制一般发生在Slave初始化阶段，这时Slave需要将Master上的所有数据都复制一份。具体步骤如下：

　　1）从服务器连接主服务器，发送SYNC命令（从服务器向主服务器初次同步时，不会影响主服务器接收客户端的请求）；

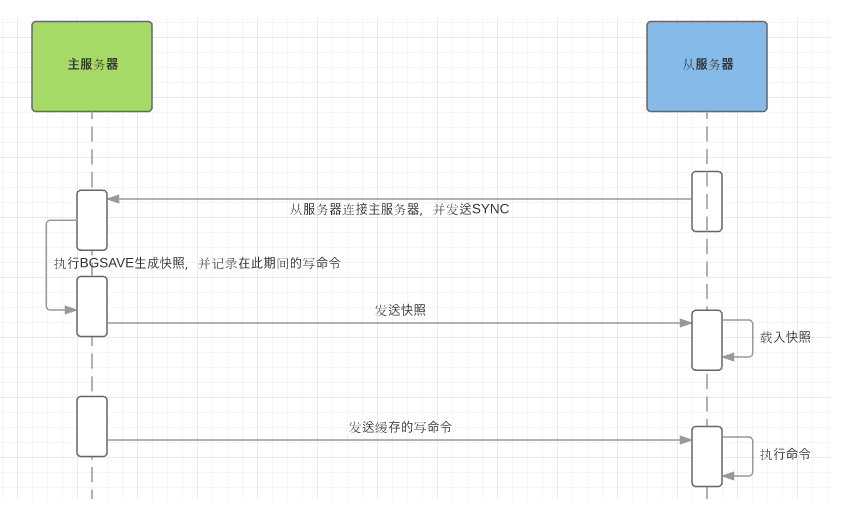
　　2）主服务器接收到SYNC命名后，开始执行BGSAVE命令生成RDB文件并使用缓冲区记录此后执行的所有写命令；

　　3）主服务器BGSAVE执行完后，向所有从服务器发送快照文件，并在发送期间继续记录被执行的写命令；

　　4）从服务器收到快照文件后丢弃所有旧数据，载入收到的快照；

　　5）主服务器快照发送完毕后开始向从服务器发送缓冲区中的写命令；

　　6）从服务器完成对快照的载入，开始接收命令请求，并执行来自主服务器缓冲区的写命令；



完成上面几个步骤后就完成了从服务器数据初始化的所有操作，从服务器此时可以接收来自用户的读请求。

2.增量同步：

　　Redis增量复制是指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。 增量复制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。

**同步策略：**

主从刚刚连接的时候，进行全量同步；

全同步结束后，进行增量同步。当然，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，要求从机进行全量同步。

**注：**

如果多个Slave断线了，需要重启的时候，因为只要Slave启动，就会发送sync请求和主机全量同步，当多个同时出现的时候，可能会导致Master IO剧增导致宕机。

建议开启master主服务器的持久化功能，避免出现master重启后，数据无法恢复；

**三、案例：配置主从复制：**

**实验环境：**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 系统 | IP地址 | 主机名 | 所需软件 | Redis角色及端口 |
| Centos 7.4 | 192.168.100.101 | master.redis.com | redis-4.0.9.tar.gz | master：6379  （redis.conf） |
| Centos 7.4 | 192.168.100.102 | slave.redis.com | redis-4.0.9.tar.gz | slave：6379  （redis.conf）  slave：6380  （redis01.conf） |

**实验步骤：**

* 安装并配置master角色的redis服务；
* 安装并配置slave角色的redis服务（双实例）；
* 配置master角色的文件；
* 配置slave角色的文件；
* 验证主从复制；
* **安装并配置master角色的redis服务；**

[root@master ~]# wget http://download.redis.io/releases/redis-4.0.9.tar.gz

[root@master ~]# tar zxvf redis-4.0.9.tar.gz

[root@master ~]# cd redis-4.0.9

[root@master redis-4.0.9]# make

[root@master redis-4.0.9]# echo $?

[root@master redis-4.0.9]# cd

[root@master ~]# mkdir -p /usr/local/redis

[root@master ~]# cp /root/redis-4.0.9/src/redis-server /usr/local/redis/ ##服务端程序

[root@master ~]# cp /root/redis-4.0.9/src/redis-cli /usr/local/redis/ ##客户端程序

[root@master ~]# cp /root/redis-4.0.9/redis.conf /usr/local/redis/ ##主配置文件

[root@master ~]# ls /usr/local/redis/

redis-cli redis.conf redis-server

[root@master ~]# sed -i '/^bind 127.0.0.1$/s/127.0.0.1/192.168.100.101/g' /usr/local/redis/redis.conf

[root@master ~]# sed -i '/protected-mode/s/yes/no/g' /usr/local/redis/redis.conf ##关闭redis的保护模式

[root@master ~]# sed -i '/daemonize/s/no/yes/g' /usr/local/redis/redis.conf ##开启redis的后台守护进程模式

[root@master ~]# sed -i '/requirepass/s/foobared/123123/g' /usr/local/redis/redis.conf ##设置redis的密码为123123

[root@master ~]# sed -i '/requirepass 123123/s/^#//g' /usr/local/redis/redis.conf ##开启redis的密码

[root@master ~]# ln -s /usr/local/redis/redis-cli /usr/local/bin/redis

[root@master ~]# cat <<END >>/etc/init.d/redis

#!/bin/sh

# chkconfig: 2345 80 90

# description: Start and Stop redis

#PATH=/usr/local/bin:/sbin:/usr/bin:/bin

REDISPORT=6379

EXEC=/usr/local/redis/redis-server

REDIS\_CLI=/usr/local/redis/redis-cli

PIDFILE=/var/run/redis\_6379.pid

CONF="/usr/local/redis/redis.conf"

AUTH="123123"

LISTEN\_IP=\$(netstat -utpln |grep redis-server |awk '{print \$4}'|awk -F':' '{print \$1}')

case "\$1" in

start)

if [ -f \$PIDFILE ]

then

echo "\$PIDFILE exists, process is already running or crashed"

else

echo "Starting Redis server..."

\$EXEC \$CONF

fi

if [ "\$?"="0" ]

then

echo "Redis is running..."

fi

;;

stop)

if [ ! -f \$PIDFILE ]

then

echo "\$PIDFILE does not exist, process is not running"

else

PID=\$(cat \$PIDFILE)

echo "Stopping ..."

\$REDIS\_CLI -h \$LISTEN\_IP -p \$REDISPORT -a \$AUTH SHUTDOWN

while [ -x \${PIDFILE} ]

do

echo "Waiting for Redis to shutdown ..."

sleep 1

done

echo "Redis stopped"

fi

;;

restart|force-reload)

\${0} stop

\${0} start

;;

\*)

echo "Usage: /etc/init.d/redis {start|stop|restart|force-reload}" >&2

exit 1

esac

END

[root@master ~]# chmod 755 /etc/init.d/redis

[root@master ~]# chkconfig --add redis

[root@master ~]# /etc/init.d/redis start

Starting Redis server...

4390:C 04 May 02:16:45.232 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo

4390:C 04 May 02:16:45.232 # Redis version=4.0.9, bits=64, commit=00000000, modified=0, pid=4390, just started

4390:C 04 May 02:16:45.232 # Configuration loaded

Redis is running...

[root@master ~]# netstat -utpln |grep redis

tcp 0 192.168.100.101:6379 0.0.0.0:\* LISTEN 4204/redis-server \*

[root@master ~]# redis -h 192.168.100.101 -a 123123 -p 6379

192.168.100.101:6379> exit

* **安装并配置slave角色的redis服务（双实例）；**

[root@slave ~]# wget http://download.redis.io/releases/redis-4.0.9.tar.gz

[root@slave ~]# tar zxvf redis-4.0.9.tar.gz

[root@slave ~]# cd redis-4.0.9

[root@slave redis-4.0.9]# make

[root@slave redis-4.0.9]# echo $?

[root@slave redis-4.0.9]# cd

[root@slave ~]# mkdir -p /usr/local/redis

[root@slave ~]# cp /root/redis-4.0.9/src/redis-server /usr/local/redis/ ##服务端程序

[root@slave ~]# cp /root/redis-4.0.9/src/redis-cli /usr/local/redis/ ##客户端程序

[root@slave ~]# cp /root/redis-4.0.9/redis.conf /usr/local/redis/ ##主配置文件

[root@slave ~]# ls /usr/local/redis/

redis-cli redis.conf redis-server

[root@slave ~]# sed -i '/^bind 127.0.0.1$/s/127.0.0.1/192.168.100.102/g' /usr/local/redis/redis.conf

[root@slave ~]# sed -i '/protected-mode/s/yes/no/g' /usr/local/redis/redis.conf ##关闭redis的保护模式

[root@slave ~]# sed -i '/daemonize/s/no/yes/g' /usr/local/redis/redis.conf ##开启redis的后台守护进程模式

[root@slave ~]# sed -i '/requirepass/s/foobared/123123/g' /usr/local/redis/redis.conf ##设置redis的密码为123123

[root@slave ~]# sed -i '/requirepass 123123/s/^#//g' /usr/local/redis/redis.conf ##开启redis的密码

[root@slave ~]# cp /usr/local/redis/redis.conf /usr/local/redis/redis01.conf

[root@slave ~]# sed -i '92s/6379/6380/g' /usr/local/redis/redis01.conf ##更改监听端口

[root@slave ~]# sed -i '158s/6379/6380/g' /usr/local/redis/redis01.conf ##更改PID文件

[root@slave ~]# sed -i '171s/^\(.\).\{9\}/logfile "\/usr\/local\/redis\/redis01.log"/g' /usr/local/redis/redis01.conf ##更改日志文件位置，9代表旧内容的字节数

[root@slave ~]# ln -s /usr/local/redis/redis-cli /usr/local/bin/redis

[root@slave ~]# cat <<END >>/etc/init.d/redis

#!/bin/sh

# chkconfig: 2345 80 90

# description: Start and Stop redis

#PATH=/usr/local/bin:/sbin:/usr/bin:/bin

REDISPORT=6379

EXEC=/usr/local/redis/redis-server

REDIS\_CLI=/usr/local/redis/redis-cli

PIDFILE=/var/run/redis\_6379.pid

CONF="/usr/local/redis/redis.conf"

AUTH="123123"

LISTEN\_IP=\$(netstat -utpln |grep redis-server |awk '{print \$4}'|awk -F':' '{print \$1}' |uniq)

case "\$1" in

start)

if [ -f \$PIDFILE ]

then

echo "\$PIDFILE exists, process is already running or crashed"

else

echo "Starting Redis server..."

\$EXEC \$CONF

fi

if [ "\$?"="0" ]

then

echo "Redis is running..."

fi

;;

stop)

if [ ! -f \$PIDFILE ]

then

echo "\$PIDFILE does not exist, process is not running"

else

PID=\$(cat \$PIDFILE)

echo "Stopping ..."

\$REDIS\_CLI -h \$LISTEN\_IP -p \$REDISPORT -a \$AUTH SHUTDOWN

while [ -x \${PIDFILE} ]

do

echo "Waiting for Redis to shutdown ..."

sleep 1

done

echo "Redis stopped"

fi

;;

restart|force-reload)

\${0} stop

\${0} start

;;

\*)

echo "Usage: /etc/init.d/redis {start|stop|restart|force-reload}" >&2

exit 1

esac

END

[root@slave ~]# chmod 755 /etc/init.d/redis

[root@slave ~]# chkconfig --add redis

[root@slave ~]# cp /etc/init.d/redis /etc/init.d/redis01

[root@slave ~]# sed -i 's/6379/6380/g' /etc/init.d/redis01

[root@slave ~]# sed -i '/CONF=/s/redis.conf/redis01.conf/g' /etc/init.d/redis01

[root@slave ~]# /etc/init.d/redis start

Starting Redis server...

4390:C 04 May 02:16:45.232 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo

4390:C 04 May 02:16:45.232 # Redis version=4.0.9, bits=64, commit=00000000, modified=0, pid=4390, just started

4390:C 04 May 02:16:45.232 # Configuration loaded

Redis is running...

[root@slave ~]# /etc/init.d/redis01 start

Starting Redis server...

4390:C 04 May 02:16:45.232 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo

4390:C 04 May 02:16:45.232 # Redis version=4.0.9, bits=64, commit=00000000, modified=0, pid=4390, just started

4390:C 04 May 02:16:45.232 # Configuration loaded

Redis is running...

[root@slave ~]# netstat -utpln |grep redis

tcp 0 0 192.168.100.102:6379 0.0.0.0:\* LISTEN 11864/redis-server

tcp 0 0 192.168.100.102:6380 0.0.0.0:\* LISTEN 11877/redis-server

[root@ slave ~]# redis -h 192.168.100.102 -a 123123 -p 6379

192.168.100.102:6379> exit

[root@ slave ~]# redis -h 192.168.100.102 -a 123123 -p 6380

192.168.100.102:6380> exit

* **配置master角色的文件；**

[root@master ~]# sed -i '450s/^\(.\).\{22\}/min-slaves-to-write 2/g' /usr/local/redis/redis.conf

##设置slave节点的数量，如果slave节点数量少于此值，那么master节点将停止客户端的一切写请求

[root@master ~]# sed -n '451s/^\(.\).\{22\}/min-slaves-max-lag 10/g' /usr/local/redis/redis.conf

##master与slave之间同步数据的超时时间，若超过此时间，master节点将停止客户端的一切写操作

[root@master ~]# /etc/init.d/redis restart

Stopping ...

Redis stopped

Starting Redis server...

1638:C 15 May 16:32:08.301 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo

1638:C 15 May 16:32:08.301 # Redis version=4.0.9, bits=64, commit=00000000, modified=0, pid=1638, just started

1638:C 15 May 16:32:08.301 # Configuration loaded

Redis is running...

* **配置slave角色的文件；**

[root@slave ~]# sed -i '281s/^\(.\).\{32\}/slaveof 192.168.100.101 6379/g' /usr/local/redis/redis.conf

##指定master的ip地址以及端口

[root@slave ~]# sed -i '288s/^\(.\).\{29\}/masterauth 123123/g' /usr/local/redis/redis.conf

##指定master的连接密码

[root@slave ~]# /etc/init.d/redis restart

/var/run/redis\_6379.pid does not exist, process is not running

Starting Redis server...

4387:C 18 May 03:24:00.027 # oO0OoO0OoO0Oo Redis is starting oO0OoO0OoO0Oo

4387:C 18 May 03:24:00.027 # Redis version=4.0.9, bits=64, commit=00000000, modified=0, pid=4387, just started

4387:C 18 May 03:24:00.027 # Configuration loaded

Redis is running...

[root@slave ~]# sed -i '281s/^\(.\).\{32\}/slaveof 192.168.100.101 6379/g' /usr/local/redis/redis01.conf

[root@slave ~]# sed -i '288s/^\(.\).\{29\}/masterauth 123123/g' /usr/local/redis/redis01.conf

[root@slave ~]# /etc/init.d/redis01 restart

Stopping ...

Redis stopped

Starting Redis server...

Redis is running...

* **验证主从复制；**

[root@master ~]# redis -h 192.168.100.101 -a 123123 -p 6379 ##在master节点上创建键值对

192.168.100.101:6379> set name xiaoming

OK

192.168.100.101:6379> keys \*

1) "name"

192.168.100.101:6379> get name

"xiaoming"

192.168.100.101:6379> info replication ##查看复制信息

# Replication

role:master

connected\_slaves:2

min\_slaves\_good\_slaves:2

slave0:ip=192.168.100.102,port=6380,state=online,offset=522,lag=1

slave1:ip=192.168.100.102,port=6379,state=online,offset=522,lag=1

master\_replid:46ec2c7e971d337d060d183d3a0c1313c5dd1683

master\_replid2:0000000000000000000000000000000000000000

master\_repl\_offset:522

second\_repl\_offset:-1

repl\_backlog\_active:1

repl\_backlog\_size:1048576

repl\_backlog\_first\_byte\_offset:1

repl\_backlog\_histlen:522

192.168.100.101:6379> exit

[root@slave ~]# redis -h 192.168.100.102 -p 6379 -a 123123 ##登录slave节点验证键值同步情况，并测试无法写入

192.168.100.102:6379> keys \*

1) "name"

192.168.100.102:6379> get name

"xiaoming"

192.168.100.102:6380> set name1 xiaohong

(error) READONLY You can't write against a read only slave.

192.168.100.102:6379> exit

[root@slave ~]# redis -h 192.168.100.102 -p 6380 -a 123123

192.168.100.102:6380> keys \*

1) "name"

192.168.100.102:6380> get name

"xiaoming"

192.168.100.102:6380> exit

[root@master ~]# reboot ##重启master节点，验证redis默认的RDB持久化

[root@master ~]# /etc/init.d/redis restart

[root@master ~]# redis -h 192.168.100.101 -a 123123 -p 6379

192.168.100.101:6379> keys \*

1) "name"

192.168.100.101:6379> exit

[root@slave ~]# redis -h 192.168.100.102 -p 6380 -a 123123

192.168.100.102:6380> keys \*

1) "name"

192.168.100.102:6380> get name

"xiaoming"

192.168.100.102:6380> exit