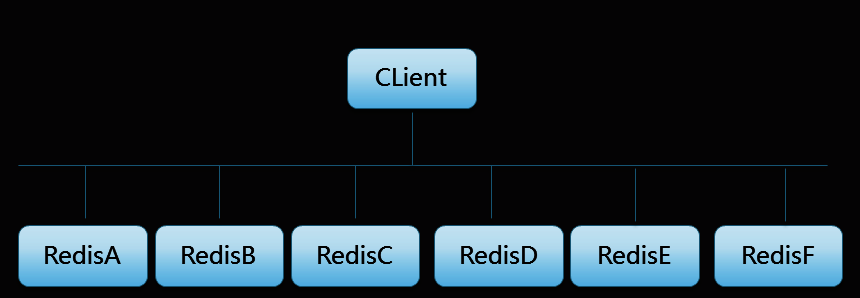
NOSQL-day02

# redis 集群

准备集群环境

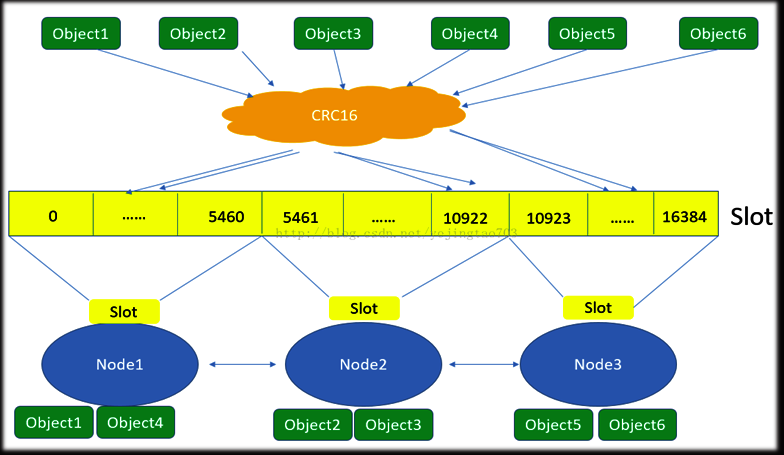
六台Redis服务器



IP地址规划

|  |  |  |
| --- | --- | --- |
| redis服务器 | IP地址 | 端口规划 |
| RedisA | 192.168.4.51 | 6351 |
| RedisB | 192.168.4.52 | 6352 |
| RedisC | 192.168.4.53 | 6353 |
| RedisD | 192.168.4.54 | 6354 |
| RedisE | 192.168.4.55 | 6355 |
| RedisF | 192.168.4.56 | 6356 |

集群工作原理



## 创建集群

### 1. 装包

为六台主机安装并运行redis服务(在6台主机上做相同工作)

#] yum -y install gcc

#] tar -xf /root/redis-4.0.8.tar.gz

#] cd /root/redis-4.0.8/

#] make && make install

#] ./utils/install\_server.sh

### 2. 调整配置文件（以51为例）

6台主机全部执行此操作

# vim /etc/redis/6351.conf

port 6351 //对口号不要使用默认的6379

bind 192.168.4.51 //只写物理接口IP地址

daemonize yes //守护进程方式运行

cluster-enabled yes //启用集群

cluster-config-file nodes-6351.conf //指定集群信息文件

cluster-node-timeout 5000 //请求超时5秒

以上修改部分

**查看服务信息**

# /etc/init.d/redis\_6351 status

# /etc/init.d/redis\_6351 restart

# ss -natpul | grep redis-server

users:(("redis-server",pid=2538,fd=6))

users:(("redis-server",pid=2538,fd=8))

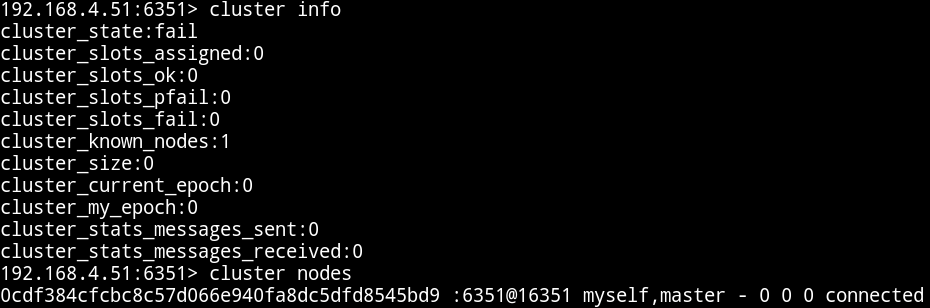
16351:集群通信端口，默认服务端口+10000

**查看集群信息(此时为初始状态)**

[root@mysql51 ~]# redis-cli -h 192.168.4.51 -p 6351

192.168.4.51:6351> cluster info //查看集群信息

192.168.4.51:6351> cluster nodes //查看集群节点信息



### 3. 配置管理主机（以51为例）

**在选中的一台redis服务器上，执行创建集群脚本**

— 部署ruby脚本运行环境（ruby连接redis）

— 创建集群

**安装提供管理功能的软件包**

[root@mysql51 ~]# yum -y install ruby rubygems.noarch

[root@mysql51 ~]# rpm -ivh ruby-devel-2.0.0.648-30.el7.x86\_64.rpm

[root@mysql51 ~]# gem install redis-3.2.1.gem //接口程序gem

**创建管理集群的命令**

[root@mysql51 ~]# cd redis-4.0.8/src/

[root@mysql51 src]# cp redis-trib.rb /root/bin //拷贝脚本到/root/bin下

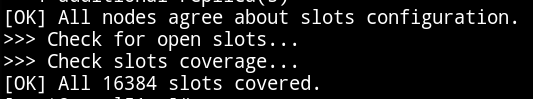
### 4. 创建集群

[root@mysql51 ~]# redis-trib.rb create --replicas 1 192.168.4.51:6351 192.168.4.52:6352 192.168.4.53:5353 192.168.4.54:6354 192.168.4.55:6355 192.168.4.56:6356

........

yes

.......

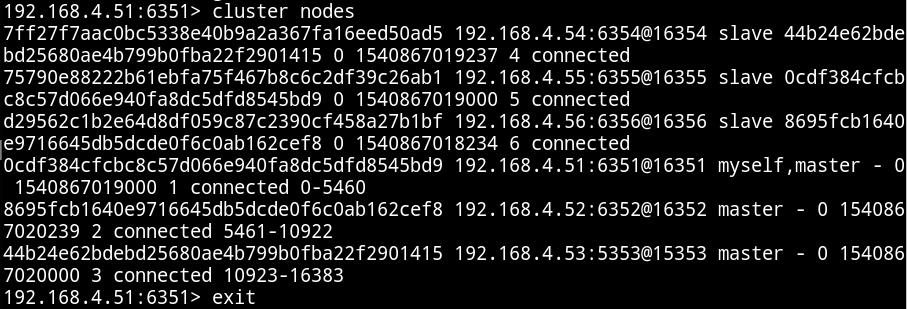


**查看集群信息**

[root@mysql51 ~]# redis-cli -h 192.168.4.51 -p 6351

192.168.4.51:6351> cluster info

192.168.4.51:6351> cluster nodes



0-5460\5461-10922\10923-16383 //hash槽范围

[root@mysql51 ~]# cat /var/lib/redis/6351/nodes.conf //集群配置文件

**检测集群：**

[root@mysql51 ~]# redis-trib.rb check 192.168.4.51:6351

**测试集群**

redis-cli -c -h ip地址 -p 端口

> set key values //存数据

**测试主从同步**

客户端连接主redis主机，进行数据写入，存储数据会存在不同的redis服务器上

[root@client50 ~]# redis-cli -c -h 192.168.4.52 -p 6352

//根据存数据时得出的hash值，确定数据存在哪台主机上

-> Redirected to slot [741] located at 192.168.4.51:6351

//741为hash值

192.168.4.52:6352> set name bob

192.168.4.52:6352> keys \*

192.168.4.52:6352> set age 21

192.168.4.51:6351> keys \*

192.168.4.51:6351> set shcool tarena

**测试高可用功能：**

宕掉三台主redis服务主机的其中一台，管理主机会自动检测并对主库作出调整

## 管理集群

### 管理命令

**redis-cli命令**

查看命令帮助

— redis-cli -h

常用选项

— -h IP地址

— -p 端口

— c 集群模式

**redis-trib.rb脚本**

语法格式

——redis-trib.rb 选项 参数

选项

— add-node 添加master 主机

— check 检测集群

— reshard 对节点主机重新分片

— add-node --slave 添加从节点主机

— del-node 删除节点主机

### master选举测试

**选举master主机**

**停止master主机的redis服务**

— master宕机后对应的slave 自动被选举为master

— 原master启动后，会自动配置为当前master的slave

**查看集群主机信息**

— redis-cli -h master\_ip -p master\_port

### 添加新节点

向集群里添加新的redis服务，配置步骤:

1. 主机57运行redis服务并启用的集群配置

装包———>初始化———>启用集群配置———>重启服务

1. 再在管理主机上把服务器添加到集群里

### 添加master主机

添加master主机———>检查主机———>重新分片

添加时不指定主机角色，默认新主机被选为Master

redis-trib.rb add-node 新主机IP：端口 源主机IP：端口

**查看当前集群管理主机**

[root@mysql51 ~]# redis-trib.rb check 192.168.4.51:6351

**添加新的master主机**

[root@mysql51 ~]# redis-trib.rb add-node 192.168.4.57:6357 \

> 192.168.4.51:6351 //添进去即为主库，但无hash槽

**重新分片（添加hash槽）**

redis-trib.rb reshard 源主机IP：端口

[root@mysql51 ~]# redis-cli -h 192.168.4.51 -p 6351

//检测集群主机，角色为master，无槽位数量

192.168.4.51:6351> cluster nodes

60be59e8dcb47c7e1fe49980d962e310d7e253c8 192.168.4.57:6357

@16357 master - 0 1540880966000 0 connected

[root@mysql51 ~]# redis-trib.rb reshard 192.168.4.51:6351

......

指定移出hash槽个数

How many slots do you want to move (from 1 to 16384)? 4096

指定接收hash槽主机ID

What is the receiving node ID?

60be59e8dcb47c7e1fe49980d962e310d7e253c8

Source node #1:all //指定移出hash槽主机ID

Do you want to proceed with the proposed reshard plan (yes/no)? yes

重新查看集群信息，确认57主机hash槽是否存在

[root@mysql51 ~]# redis-cli -h 192.168.4.51 -p 6351

192.168.4.51:6351> cluster nodes

### 添加salve主机

redis-trib.rb add-node --slave

--master-id id值 //此选项指定主节点的ID

ip地址：端口 192.168.4.51：6351

[root@mysql51 ~]# redis-trib.rb add-node --slave 192.168.4.58:6358

192.168.4.51:6351

//如果不指定主节点的ID时，会把新节点随机添加为从节点最少的主的从

[root@mysql51 ~]# redis-trib.rb check 192.168.4.51:6351

[root@mysql51 ~]# redis-cli -h 192.168.4.50 -p 6350

### 移除节点

**移除master主机**

步骤：

重新分片时方占用的hash槽

移除master值

指定移出hash槽个数

指定接受hash槽主机ID

指定移出hash槽主机ID

将被移出主机的hash槽拿出给51

How many slots do you want to move (from 1 to 16384)? 4096

What is the receiving node ID?

0cdf384cfcbc8c57d066e940fa8dc5dfd8545bd9 //接收主机ID

Source node #1:

60be59e8dcb47c7e1fe49980d962e310d7e253c8 //移出主机ID

Source node #2:done

Do you want to proceed with the proposed reshard plan (yes/no)? yes

[root@mysql51 ~]# redis-trib.rb check 192.168.4.51:6351

[root@mysql51 ~]# redis-trib.rb del-node 192.168.4.51:6351

60be59e8dcb47c7e1fe49980d962e310d7e253c8 //被移出主机ID

[root@mysql51 ~]# redis-trib.rb check 192.168.4.51:6351

[root@mysql57 ~]# redis-cli -h 192.168.4.57 -p 6357

192.168.4.57:6357> cluster reset //清除集群配置

=====>[root@mysql57 ~]# rm -rf /var/lib/redis/6357/nodes-6357.conf

**移除slave主机**

从节点主机没有槽位范围，直接删除即可

[root@mysql51 ~]# redis-trib.rb check 192.168.4.51:6351

//获取集群主机的信息

redis-trib.rb del-node 192.168.4.51:6351 从主机id

[root@mysql51 ~]# redis-trib.rb del-node 192.168.4.51:6351

adf8a7e03e3697bafe909ed483be5f6507b9c554

//移除slave主机时，保存数据关闭slave主机redis服务

彻底删除slave主机须删除集群信息文件之后重新加载

[root@mysql58 redis-4.0.8]# ls /var/lib/redis/6358/

[root@mysql58 redis-4.0.8]# rm -rf /var/lib/redis/6358/nodes-6358.conf

[root@mysql58 redis-4.0.8]# redis-cli -h 192.168.4.58 -p 6358 shutdown

[root@mysql58 redis-4.0.8]# /etc/init.d/redis\_6358 start

[root@mysql58 redis-4.0.8]# redis-cli -h 192.168.4.58 -p 6358

192.168.4.58:6358> cluster info----------->cluster\_state:fail

192.168.4.58:6358> cluster nodes

401ed0869542b4c838be670147e989c44578b2a4 :6358@16358

myself,master - 0 0 0 connected 417 741 5798

### 扩展：

此时主机58已经从集群里移出，但之前存的数据还在，只不过读取不了，若需要读取，则进行以下配置：

[root@mysql58 redis-4.0.8]# redis-cli -h 192.168.4.58 -p 6358 shutdown

[root@mysql58 redis-4.0.8]# rm -rf /var/lib/redis/6358/nodes-6358.conf

注释掉redis配置文件中的cluster选项

[root@mysql58 redis-4.0.8]# vim /etc/redis/6358.conf

#cluster-enabled yes

#cluster-config-file nodes-6358.conf

#cluster-node-timeout 5000

[root@mysql58 redis-4.0.8]# /etc/init.d/redis\_6358 start

[root@mysql58 redis-4.0.8]# ss -natpul | grep redis-server

tcp LISTEN 0 128 192.168.4.58:6358 \*:\*

users:(("redis-server",pid=7552,fd=6))

完成以上配置，进行数据读取

[root@mysql58 redis-4.0.8]# redis-cli -h 192.168.4.58 -p 6358