# Redis Cluster集群搭建

# 前言：地址说明

对当前所有用到的软件版本及地址进行说明：

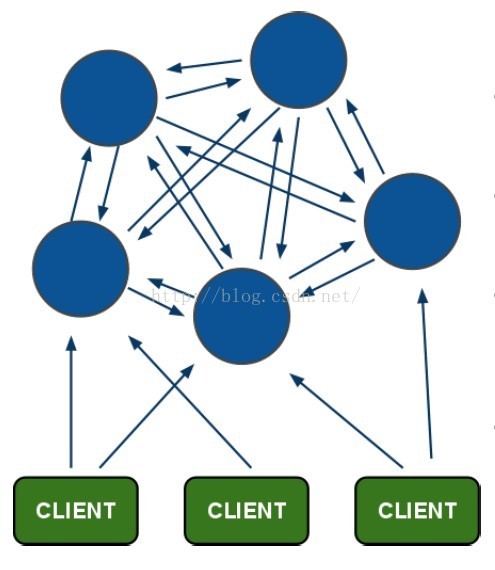
局域网Svn地址 https://192.168.1.60/svn/resources/linux/RedisCluster

阿里云code Svn地址 http://code.taobao.org/svn/RedisCluster/

Git hub 地址 https://github.com/guoshiqiufeng/RedisCluster

一、redis-cluster设计

Redis集群搭建的方式有多种，例如使用zookeeper等，但从redis 3.0之后版本支持redis-cluster集群，Redis-Cluster采用无中心结构，每个节点保存数据和整个集群状态,每个节点都和其他所有 节点连接。其redis-cluster架构图如下：



其结构特点：

     1、所有的redis节点彼此互联(PING-PONG机制),内部使用二进制协议优化传输速度和带宽。  
     2、节点的fail是通过集群中超过半数的节点检测失效时才生效。  
     3、客户端与redis节点直连,不需要中间proxy层.客户端不需要连接集群所有节点,连接集群中任何一个可用节点即可。  
     4、redis-cluster把所有的物理节点映射到[0-16383]slot上（不一定是平均分配）,cluster 负责维护node<->slot<->value。

     5、Redis集群预分好16384个桶，当需要在 Redis 集群中放置一个 key-value 时，根据 CRC16(key) mod 16384的值，决定将一个key放到哪个桶中。

## 1、redis cluster节点分配

现在我们是三个主节点分别是：A, B, C 三个节点，它们可以是一台机器上的三个端口，也可以是三台不同的服务器。那么，采用哈希槽 (hash slot)的方式来分配16384个slot 的话，它们三个节点分别承担的slot 区间是：  
  
  
      节点A覆盖0－5460;  
      节点B覆盖5461－10922;  
      节点C覆盖10923－16383.

     获取数据：

      如果存入一个值，按照redis cluster哈希槽的算法： CRC16('key')%16384 = 6782。 那么就会把这个key 的存储分配到 B 上了。同样，当我连接(A,B,C)任何一个节点想获取'key'这个key时，也会这样的算法，然后内部跳转到B节点上获取数据

     新增一个主节点：

      新增一个节点D，redis cluster的这种做法是从各个节点的前面各拿取一部分slot到D上，我会在接下来的实践中实验。大致就会变成这样：

    节点A覆盖1365-5460  
    节点B覆盖6827-10922  
    节点C覆盖12288-16383  
    节点D覆盖0-1364,5461-6826,10923-12287  
  
     同样删除一个节点也是类似，移动完成后就可以删除这个节点了。

## 2、Redis Cluster主从模式

          redis cluster 为了保证数据的高可用性，加入了主从模式，一个主节点对应一个或多个从节点，主节点提供数据存取，从节点则是从主节点拉取数据备份，当这个主节点挂掉后，就会有这个从节点选取一个来充当主节点，从而保证集群不会挂掉。  
  
      上面那个例子里, 集群有ABC三个主节点, 如果这3个节点都没有加入从节点，如果B挂掉了，我们就无法访问整个集群了。A和C的slot也无法访问。  
  
     所以我们在集群建立的时候，一定要为每个主节点都添加了从节点, 比如像这样, 集群包含主节点A、B、C, 以及从节点A1、B1、C1, 那么即使B挂掉系统也可以继续正确工作。  
  
     B1节点替代了B节点，所以Redis集群将会选择B1节点作为新的主节点，集群将会继续正确地提供服务。 当B重新开启后，它就会变成B1的从节点。  
  
    不过需要注意，如果节点B和B1同时挂了，Redis集群就无法继续正确地提供服务了。

# 二、redis集群的搭建

    集群中至少应该有奇数个节点，所以至少有三个节点，每个节点至少有一个备份节点，所以下面使用6节点（主节点、备份节点由redis-cluster集群确定）。

     下面使用redis-3.2.0安装，[下载地址](http://redis.io/download)   或者

wget http://download.redis.io/releases/redis-3.2.0.tar.gz

## 1、安装redis节点指定端口

         解压redis压缩包，编译安装

[root@localhost redis-3.2.0]# tar xzf redis-3.2.0.tar.gz

[root@localhost redis-3.2.0]# cd redis-3.2.0

[root@localhost redis-3.2.0]# make

[root@localhost redis01]# make install PREFIX=/usr/andy/redis-cluster

        在redis-cluster下 修改bin文件夹为redis01,复制redis.conf配置文件

     配置redis的配置文件redis.conf

bind 127.0.0.1 #ip

         daemonize yes #后台启动

        port 7001 #修改端口号，从7001到7006

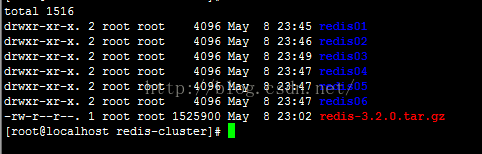
        cluster-enabled yes #开启cluster，去掉注释

        cluster-config-file nodes.conf

        cluster-node-timeout 3000

        appendonly yes

        复制六份，修改对应的端口号



## 2、安装redis-trib所需的 ruby脚本

          复制redis解压文件src下的redis-trib.rb文件到redis-cluster目录

[root@localhost redis-cluster]# cp /usr/andy/redis/redis-3.2.0/src/redis-trib.rb ./

         安装ruby环境：

[root@localhost redis-cluster]# yum install ruby

[root@localhost redis-cluster]# yum install rubygems

        安装redis-trib.rb运行依赖的ruby的包redis-3.2.2.gem，[下载](https://rubygems.global.ssl.fastly.net/gems/redis-3.2.2.gem)

[root@localhost redis-cluster]# gem install redis-3.2.2.gem

## 3、启动所有的redis节点

       可以写一个命令脚本start-all.sh

cd redis01

./redis-server redis.conf

cd ..

cd redis02

./redis-server redis.conf

cd ..

cd redis03

./redis-server redis.conf

cd ..

cd redis04

./redis-server redis.conf

cd ..

cd redis05

./redis-server redis.conf

cd ..

cd redis06

./redis-server redis.conf

cd ..

      设置权限启动

[root@localhost redis-cluster]# chmod 777 start-all.sh

[root@localhost redis-cluster]# ./start-all.sh

       查看redis进程启动状态

[root@localhost redis-cluster]# ps -ef | grep redis

root       4547      1  0 23:12 ?        00:00:00 ./redis-server 127.0.0.1:7001 [cluster]

root       4551      1  0 23:12 ?        00:00:00 ./redis-server 127.0.0.1:7002 [cluster]

root       4555      1  0 23:12 ?        00:00:00 ./redis-server 127.0.0.1:7003 [cluster]

root       4559      1  0 23:12 ?        00:00:00 ./redis-server 127.0.0.1:7004 [cluster]

root       4563      1  0 23:12 ?        00:00:00 ./redis-server 127.0.0.1:7005 [cluster]

root       4567      1  0 23:12 ?        00:00:00 ./redis-server 127.0.0.1:7006 [cluster]

root       4840   4421  0 23:26 pts/1    00:00:00 grep --color=auto redis

  可以看到redis的6个节点已经启动成功

    杀死全部的几点：

[root@localhost redis-cluster]# pkill -9 redis

## 4、使用redis-trib.rb创建集群

./redis-trib.rb create --replicas 1 127.0.0.1:7001 127.0.0.1:7002 127.0.0.1:7003 127.0.0.1:7004 127.0.0.1:7005 127.0.0.1:7006

       使用create命令 --replicas 1 参数表示为每个主节点创建一个从节点，其他参数是实例的地址集合。

[root@localhost redis-cluster]# ./redis-trib.rb create --replicas 1 127.0.0.1:7001 127.0.0.1:7002 127.0.0.1:7003 127.0.0.1:7004 127.0.0.1:7005 127.0.0.1:7006

>>> Creating cluster

>>> Performing hash slots allocation on 6 nodes...

Using 3 masters:

127.0.0.1:7001

127.0.0.1:7002

127.0.0.1:7003

Adding replica 127.0.0.1:7004 to 127.0.0.1:7001

Adding replica 127.0.0.1:7005 to 127.0.0.1:7002

Adding replica 127.0.0.1:7006 to 127.0.0.1:7003

M: dfd510594da614469a93a0a70767ec9145aefb1a 127.0.0.1:7001

   slots:0-5460 (5461 slots) master

M: e02eac35110bbf44c61ff90175e04d55cca097ff 127.0.0.1:7002

   slots:5461-10922 (5462 slots) master

M: 4385809e6f4952ecb122dbfedbee29109d6bb234 127.0.0.1:7003

   slots:10923-16383 (5461 slots) master

S: ec02c9ef3acee069e8849f143a492db18d4bb06c 127.0.0.1:7004

   replicates dfd510594da614469a93a0a70767ec9145aefb1a

S: 83e5a8bb94fb5aaa892cd2f6216604e03e4a6c75 127.0.0.1:7005

   replicates e02eac35110bbf44c61ff90175e04d55cca097ff

S: 10c097c429ca24f8720986c6b66f0688bfb901ee 127.0.0.1:7006

   replicates 4385809e6f4952ecb122dbfedbee29109d6bb234

Can I set the above configuration? (type 'yes' to accept): yes

>>> Nodes configuration updated

>>> Assign a different config epoch to each node

>>> Sending CLUSTER MEET messages to join the cluster

Waiting for the cluster to join......

>>> Performing Cluster Check (using node 127.0.0.1:7001)

M: dfd510594da614469a93a0a70767ec9145aefb1a 127.0.0.1:7001

   slots:0-5460 (5461 slots) master

M: e02eac35110bbf44c61ff90175e04d55cca097ff 127.0.0.1:7002

   slots:5461-10922 (5462 slots) master

M: 4385809e6f4952ecb122dbfedbee29109d6bb234 127.0.0.1:7003

   slots:10923-16383 (5461 slots) master

M: ec02c9ef3acee069e8849f143a492db18d4bb06c 127.0.0.1:7004

   slots: (0 slots) master

   replicates dfd510594da614469a93a0a70767ec9145aefb1a

M: 83e5a8bb94fb5aaa892cd2f6216604e03e4a6c75 127.0.0.1:7005

   slots: (0 slots) master

   replicates e02eac35110bbf44c61ff90175e04d55cca097ff

M: 10c097c429ca24f8720986c6b66f0688bfb901ee 127.0.0.1:7006

   slots: (0 slots) master

   replicates 4385809e6f4952ecb122dbfedbee29109d6bb234

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

       上面显示创建成功，有3个主节点，3个从节点，每个节点都是成功连接状态。

       3个主节点[M]以及分配的哈希卡槽如下：

  M: dfd510594da614469a93a0a70767ec9145aefb1a 127.0.0.1:7001

       slots:0-5460 (5461 slots) master

  M: e02eac35110bbf44c61ff90175e04d55cca097ff 127.0.0.1:7002

       slots:5461-10922 (5462 slots) master

  M: 4385809e6f4952ecb122dbfedbee29109d6bb234 127.0.0.1:7003

       slots:10923-16383 (5461 slots) master

          3个从节点[S]以及附属的主节点如下：

          S: ec02c9ef3acee069e8849f143a492db18d4bb06c 127.0.0.1:7004  
             replicates dfd510594da614469a93a0a70767ec9145aefb1a  
          S: 83e5a8bb94fb5aaa892cd2f6216604e03e4a6c75 127.0.0.1:7005  
             replicates e02eac35110bbf44c61ff90175e04d55cca097ff  
          S: 10c097c429ca24f8720986c6b66f0688bfb901ee 127.0.0.1:7006  
             replicates 4385809e6f4952ecb122dbfedbee29109d6bb234

        以上集群安装成功了，如果安装未成功报如下错误  
       >>> Creating cluster  
       [ERR] Sorry, can't connect to node  ....

          需要安装最新的ruby源码，[下载](https://cache.ruby-lang.org/pub/ruby/2.3/ruby-2.3.1.tar.gz)

[root@localhost redis-cluster]# tar -zxvf ruby-2.3.1.tar.gz

[root@localhost redis-cluster]# cd

[root@localhost redis-cluster]# ./configure --prefix=/usr/local/ruby-2.3.1

[root@localhost redis-cluster]# make && make install

[root@localhost redis-cluster]#gem install redis



# 三、redis集群的测试

## 1、测试存取值

         客户端连接集群redis-cli需要带上 -c ，redis-cli -c -p 端口号

[root@localhost redis01]# ./redis-cli -c -p 7001

127.0.0.1:7001> set name andy

-> Redirected to slot [5798] located at 127.0.0.1:7002

OK

127.0.0.1:7002> get name

"andy"

127.0.0.1:7002>

        根据redis-cluster的key值分配，name应该分配到节点7002[5461-10922]上，上面显示redis cluster自动从7001跳转到了7002节点。

       我们可以测试一下7006从节点获取name值

[root@localhost redis06]# ./redis-cli -c -p 7006

127.0.0.1:7006> get name

-> Redirected to slot [5798] located at 127.0.0.1:7002

"andy"

127.0.0.1:7002>

        7006位7003的从节点，从上面也是自动跳转至7002获取值，这也是redis cluster的特点，它是去中心化，每个节点都是对等的，连接哪个节点都可以获取和设置数据。

# 四、集群节点选举

         现在模拟将7002节点挂掉，按照redis-cluster原理会选举会将 7002的从节点7005选举为主节点。

[root@localhost redis-cluster]# ps -ef | grep redis

root       7950      1  0 12:50 ?        00:00:28 ./redis-server 127.0.0.1:7001 [cluster]

root       7952      1  0 12:50 ?        00:00:29 ./redis-server 127.0.0.1:7002 [cluster]

root       7956      1  0 12:50 ?        00:00:29 ./redis-server 127.0.0.1:7003 [cluster]

root       7960      1  0 12:50 ?        00:00:29 ./redis-server 127.0.0.1:7004 [cluster]

root       7964      1  0 12:50 ?        00:00:29 ./redis-server 127.0.0.1:7005 [cluster]

root       7966      1  0 12:50 ?        00:00:29 ./redis-server 127.0.0.1:7006 [cluster]

root      11346  10581  0 14:57 pts/2    00:00:00 grep --color=auto redis

[root@localhost redis-cluster]# kill 7952

       在查看集群中的7002节点

[root@localhost redis-cluster]#

[root@localhost redis-cluster]# ./redis-trib.rb check 127.0.0.1:7002

[ERR] Sorry, can't connect to node 127.0.0.1:7002

[root@localhost redis-cluster]# ./redis-trib.rb check 127.0.0.1:7005

>>> Performing Cluster Check (using node 127.0.0.1:7005)

M: a5db243087d8bd423b9285fa8513eddee9bb59a6 127.0.0.1:7005

   slots:5461-10922 (5462 slots) master

   0 additional replica(s)

S: 50ce1ea59106b4c2c6bc502593a6a7a7dabf5041 127.0.0.1:7004

   slots: (0 slots) slave

   replicates dd19221c404fb2fc4da37229de56bab755c76f2b

M: f9886c71e98a53270f7fda961e1c5f730382d48f 127.0.0.1:7003

   slots:10923-16383 (5461 slots) master

   1 additional replica(s)

M: dd19221c404fb2fc4da37229de56bab755c76f2b 127.0.0.1:7001

   slots:0-5460 (5461 slots) master

   1 additional replica(s)

S: 8bb3ede48319b46d0015440a91ab277da9353c8b 127.0.0.1:7006

   slots: (0 slots) slave

   replicates f9886c71e98a53270f7fda961e1c5f730382d48f

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

[root@localhost redis-cluster]#

可以看到集群连接不了7002节点，而7005有原来的S转换为M节点，代替了原来的7002节点。我们可以获取name值：

[root@localhost redis01]# ./redis-cli -c -p 7001

127.0.0.1:7001> get name

-> Redirected to slot [5798] located at 127.0.0.1:7005

"andy"

127.0.0.1:7005>

127.0.0.1:7005>

从7001节点连入，自动跳转到7005节点，并且获取name值。

    现在我们将7002节点恢复，看是否会自动加入集群中以及充当的M还是S节点。

[root@localhost redis-cluster]# cd redis02

[root@localhost redis02]# ./redis-server redis.conf

[root@localhost redis02]#

     在check一下7002节点

[root@localhost redis-cluster]# ./redis-trib.rb check 127.0.0.1:7002

>>> Performing Cluster Check (using node 127.0.0.1:7002)

S: 1f07d76585bfab35f91ec711ac53ab4bc00f2d3a 127.0.0.1:7002

   slots: (0 slots) slave

   replicates a5db243087d8bd423b9285fa8513eddee9bb59a6

M: f9886c71e98a53270f7fda961e1c5f730382d48f 127.0.0.1:7003

   slots:10923-16383 (5461 slots) master

   1 additional replica(s)

M: a5db243087d8bd423b9285fa8513eddee9bb59a6 127.0.0.1:7005

   slots:5461-10922 (5462 slots) master

   1 additional replica(s)

S: 50ce1ea59106b4c2c6bc502593a6a7a7dabf5041 127.0.0.1:7004

   slots: (0 slots) slave

   replicates dd19221c404fb2fc4da37229de56bab755c76f2b

S: 8bb3ede48319b46d0015440a91ab277da9353c8b 127.0.0.1:7006

   slots: (0 slots) slave

   replicates f9886c71e98a53270f7fda961e1c5f730382d48f

M: dd19221c404fb2fc4da37229de56bab755c76f2b 127.0.0.1:7001

   slots:0-5460 (5461 slots) master

   1 additional replica(s)

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

[root@localhost redis-cluster]#

    可以看到7002节点变成了a5db243087d8bd423b9285fa8513eddee9bb59a6 7005的从节点。

# 五、集群节点添加

   节点新增包括新增主节点、从节点两种情况。以下分别做一下测试：

## 1、新增主节点

       新增一个节点7007作为主节点修改配置文件

[root@localhost redis-cluster]# cp -r  redis01 redis07

[root@localhost redis-cluster]# cd redis07/

[root@localhost redis07]# sed -i "s/7001/7007/g" ./redis.conf

      启动7007redis服务

[root@localhost redis07]# ./redis-server redis.conf

[root@localhost redis07]# netstat -anp | grep 7007

tcp        0      0 127.0.0.1:17007         0.0.0.0:\*               LISTEN      13441/./redis-serve

tcp        0      0 127.0.0.1:7007          0.0.0.0:\*               LISTEN      13441/./redis-serve

[root@localhost redis07]#

上面可以看到，7007已经启动，现在加入集群中。添加使用redis-trib.rb的add-node命令

./redis-trib.rb add-node 127.0.0.1:7007 127.0.0.1:7002

add-node是加入集群节点，127.0.0.1:7007为要加入的节点，127.0.0.1:7002 表示加入的集群的一个节点，用来辨识是哪个集群，理论上那个集群的节点都可以。

    执行以下add-node

[root@localhost redis-cluster]# ./redis-trib.rb add-node 127.0.0.1:7007 127.0.0.1:7002

>>> Adding node 127.0.0.1:7007 to cluster 127.0.0.1:7002

>>> Performing Cluster Check (using node 127.0.0.1:7002)

S: 1f07d76585bfab35f91ec711ac53ab4bc00f2d3a 127.0.0.1:7002

 slots: (0 slots) slave

   replicates a5db243087d8bd423b9285fa8513eddee9bb59a6

M: f9886c71e98a53270f7fda961e1c5f730382d48f 127.0.0.1:7003

   slots:10923-16383 (5461 slots) master

   1 additional replica(s)

M: a5db243087d8bd423b9285fa8513eddee9bb59a6 127.0.0.1:7005

   slots:5461-10922 (5462 slots) master

   1 additional replica(s)

S: 50ce1ea59106b4c2c6bc502593a6a7a7dabf5041 127.0.0.1:7004

   slots: (0 slots) slave

   replicates dd19221c404fb2fc4da37229de56bab755c76f2b

S: 8bb3ede48319b46d0015440a91ab277da9353c8b 127.0.0.1:7006

   slots: (0 slots) slave

   replicates f9886c71e98a53270f7fda961e1c5f730382d48f

M: dd19221c404fb2fc4da37229de56bab755c76f2b 127.0.0.1:7001

   slots:0-5460 (5461 slots) master

   1 additional replica(s)

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

>>> Send CLUSTER MEET to node 127.0.0.1:7007 to make it join the cluster.

[OK] New node added correctly.

[root@localhost redis-cluster]#

     可以看到7007加入这个Cluster，并成为一个新的节点。

     可以check以下7007节点状态

[root@localhost redis-cluster]# ./redis-trib.rb check 127.0.0.1:7007

>>> Performing Cluster Check (using node 127.0.0.1:7007)

M: ee3efb90e5ac0725f15238a64fc60a18a71205d7 127.0.0.1:7007

   slots: (0 slots) master

   0 additional replica(s)

S: 8bb3ede48319b46d0015440a91ab277da9353c8b 127.0.0.1:7006

   slots: (0 slots) slave

   replicates f9886c71e98a53270f7fda961e1c5f730382d48f

M: dd19221c404fb2fc4da37229de56bab755c76f2b 127.0.0.1:7001

   slots:0-5460 (5461 slots) master

   1 additional replica(s)

M: f9886c71e98a53270f7fda961e1c5f730382d48f 127.0.0.1:7003

   slots:10923-16383 (5461 slots) master

   1 additional replica(s)

S: 1f07d76585bfab35f91ec711ac53ab4bc00f2d3a 127.0.0.1:7002

   slots: (0 slots) slave

   replicates a5db243087d8bd423b9285fa8513eddee9bb59a6

M: a5db243087d8bd423b9285fa8513eddee9bb59a6 127.0.0.1:7005

   slots:5461-10922 (5462 slots) master

   1 additional replica(s)

S: 50ce1ea59106b4c2c6bc502593a6a7a7dabf5041 127.0.0.1:7004

   slots: (0 slots) slave

   replicates dd19221c404fb2fc4da37229de56bab755c76f2b

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

[root@localhost redis-cluster]#

  M: ee3efb90e5ac0725f15238a64fc60a18a71205d7 127.0.0.1:7007  
   slots: (0 slots) master  
   0 additional replica(s)

  上面信息可以看到有4个M节点，3个S节点，7007成为了M主节点，它没有附属的从节点，而且Cluster并未给7007分配哈希卡槽（0 slots）。

   可以从客户端连接集群查看一下，集群节点的连接情况

[root@localhost redis-cluster]# cd redis07/

[root@localhost redis07]# ./redis-cli -c -p 7007

127.0.0.1:7007> cluster nodes

8bb3ede48319b46d0015440a91ab277da9353c8b 127.0.0.1:7006 slave f9886c71e98a53270f7fda961e1c5f730382d48f 0 1462955393326 3 connected

dd19221c404fb2fc4da37229de56bab755c76f2b 127.0.0.1:7001 master - 0 1462955388247 1 connected 0-5460

ee3efb90e5ac0725f15238a64fc60a18a71205d7 127.0.0.1:7007 myself,master - 0 0 0 connected

f9886c71e98a53270f7fda961e1c5f730382d48f 127.0.0.1:7003 master - 0 1462955390270 3 connected 10923-16383

1f07d76585bfab35f91ec711ac53ab4bc00f2d3a 127.0.0.1:7002 slave a5db243087d8bd423b9285fa8513eddee9bb59a6 0 1462955394334 7 connected

a5db243087d8bd423b9285fa8513eddee9bb59a6 127.0.0.1:7005 master - 0 1462955392309 7 connected 5461-10922

50ce1ea59106b4c2c6bc502593a6a7a7dabf5041 127.0.0.1:7004 slave dd19221c404fb2fc4da37229de56bab755c76f2b 0 1462955389663 1 connected

127.0.0.1:7007>

     redis-cluster在新增节点时并未分配卡槽，需要我们手动对集群进行重新分片迁移数据，需要重新分片命令 reshard

              redis-trib.rb reshard 127.0.0.1:7005

     这个命令是用来迁移slot节点的，后面的127.0.0.1:7005是表示是哪个集群，端口填［7000-7007］都可以，执行结果如下：

[root@localhost redis-cluster]# ./redis-trib.rb reshard 127.0.0.1:7005

>>> Performing Cluster Check (using node 127.0.0.1:7005)

M: a5db243087d8bd423b9285fa8513eddee9bb59a6 127.0.0.1:7005

   slots:5461-10922 (5462 slots) master

   1 additional replica(s)

S: 50ce1ea59106b4c2c6bc502593a6a7a7dabf5041 127.0.0.1:7004

   slots: (0 slots) slave

   replicates dd19221c404fb2fc4da37229de56bab755c76f2b

M: f9886c71e98a53270f7fda961e1c5f730382d48f 127.0.0.1:7003

   slots:10923-16383 (5461 slots) master

   1 additional replica(s)

S: 1f07d76585bfab35f91ec711ac53ab4bc00f2d3a 127.0.0.1:7002

   slots: (0 slots) slave

   replicates a5db243087d8bd423b9285fa8513eddee9bb59a6

M: ee3efb90e5ac0725f15238a64fc60a18a71205d7 127.0.0.1:7007

   slots: (0 slots) master

   0 additional replica(s)

M: dd19221c404fb2fc4da37229de56bab755c76f2b 127.0.0.1:7001

   slots:0-5460 (5461 slots) master

   1 additional replica(s)

S: 8bb3ede48319b46d0015440a91ab277da9353c8b 127.0.0.1:7006

   slots: (0 slots) slave

   replicates f9886c71e98a53270f7fda961e1c5f730382d48f

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

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

      它提示我们需要迁移多少slot到7007上，我们平分16384个哈希槽给4个节点：16384/4 = 4096，我们需要移动4096个槽点到7007上。

[OK] All 16384 slots covered.

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

What is the receiving node ID?

     需要输入7007的节点id，ee3efb90e5ac0725f15238a64fc60a18a71205d7

Please enter all the source node IDs.

  Type 'all' to use all the nodes as source nodes for the hash slots.

  Type 'done' once you entered all the source nodes IDs.

Source node #1:

      redis-trib 会向你询问重新分片的源节点（source node），即，要从特点的哪个节点中取出 4096 个哈希槽，还是从全部节点提取4096个哈希槽， 并将这些槽移动到7007节点上面。  
  
     如果我们不打算从特定的节点上取出指定数量的哈希槽，那么可以向redis-trib输入 all，这样的话， 集群中的所有主节点都会成为源节点，redis-trib从各个源节点中各取出一部分哈希槽，凑够4096个，然后移动到7007节点上：

Source node #1:all

    然后开始从别的主节点迁移哈希槽，并且确认。

    Moving slot 1343 from dd19221c404fb2fc4da37229de56bab755c76f2b

    Moving slot 1344 from dd19221c404fb2fc4da37229de56bab755c76f2b

    Moving slot 1345 from dd19221c404fb2fc4da37229de56bab755c76f2b

    Moving slot 1346 from dd19221c404fb2fc4da37229de56bab755c76f2b

    Moving slot 1347 from dd19221c404fb2fc4da37229de56bab755c76f2b

    Moving slot 1348 from dd19221c404fb2fc4da37229de56bab755c76f2b

    Moving slot 1349 from dd19221c404fb2fc4da37229de56bab755c76f2b

    Moving slot 1350 from dd19221c404fb2fc4da37229de56bab755c76f2b

    Moving slot 1351 from dd19221c404fb2fc4da37229de56bab755c76f2b

    Moving slot 1352 from dd19221c404fb2fc4da37229de56bab755c76f2b

    Moving slot 1353 from dd19221c404fb2fc4da37229de56bab755c76f2b

    Moving slot 1354 from dd19221c404fb2fc4da37229de56bab755c76f2b

    Moving slot 1355 from dd19221c404fb2fc4da37229de56bab755c76f2b

    Moving slot 1356 from dd19221c404fb2fc4da37229de56bab755c76f2b

    Moving slot 1357 from dd19221c404fb2fc4da37229de56bab755c76f2b

    Moving slot 1358 from dd19221c404fb2fc4da37229de56bab755c76f2b

    Moving slot 1359 from dd19221c404fb2fc4da37229de56bab755c76f2b

    Moving slot 1360 from dd19221c404fb2fc4da37229de56bab755c76f2b

    Moving slot 1361 from dd19221c404fb2fc4da37229de56bab755c76f2b

    Moving slot 1362 from dd19221c404fb2fc4da37229de56bab755c76f2b

    Moving slot 1363 from dd19221c404fb2fc4da37229de56bab755c76f2b

    Moving slot 1364 from dd19221c404fb2fc4da37229de56bab755c76f2b

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

     确认之后，redis-trib就开始执行分片操作，将哈希槽一个一个从源主节点移动到7007目标主节点。

     重新分片结束后我们可以check以下节点的分配情况。

[root@localhost redis-cluster]# ./redis-trib.rb check 127.0.0.1:7001

>>> Performing Cluster Check (using node 127.0.0.1:7001)

M: dd19221c404fb2fc4da37229de56bab755c76f2b 127.0.0.1:7001

   slots:1365-5460 (4096 slots) master

   1 additional replica(s)

M: ee3efb90e5ac0725f15238a64fc60a18a71205d7 127.0.0.1:7007

   slots:0-1364,5461-6826,10923-12287 (4096 slots) master

   0 additional replica(s)

M: a5db243087d8bd423b9285fa8513eddee9bb59a6 127.0.0.1:7005

   slots:6827-10922 (4096 slots) master

   1 additional replica(s)

S: 8bb3ede48319b46d0015440a91ab277da9353c8b 127.0.0.1:7006

   slots: (0 slots) slave

   replicates f9886c71e98a53270f7fda961e1c5f730382d48f

M: f9886c71e98a53270f7fda961e1c5f730382d48f 127.0.0.1:7003

   slots:12288-16383 (4096 slots) master

   1 additional replica(s)

S: 1f07d76585bfab35f91ec711ac53ab4bc00f2d3a 127.0.0.1:7002

   slots: (0 slots) slave

   replicates a5db243087d8bd423b9285fa8513eddee9bb59a6

S: 50ce1ea59106b4c2c6bc502593a6a7a7dabf5041 127.0.0.1:7004

   slots: (0 slots) slave

   replicates dd19221c404fb2fc4da37229de56bab755c76f2b

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

[root@localhost redis-cluster]#

   slots:0-1364,5461-6826,10923-12287 (4096 slots) master

   可以看到7007节点分片的哈希槽片不是连续的，间隔的移动。

[root@localhost redis-cluster]# cd redis07/

[root@localhost redis07]# ./redis-cli -c 7007

Could not connect to Redis at 127.0.0.1:6379: Connection refused

[root@localhost redis07]# ./redis-cli -c -p 7007

127.0.0.1:7007> keys \*

1) "name"

2) "age"

127.0.0.1:7007>

127.0.0.1:7007>

    可以看到将7001的age[741]和name[5798]移动到7007节点上，

    主节点7007添加成功。

## 2、新增从节点

       新增一个节点7008节点，使用add-node --slave命令。

[root@localhost redis-cluster]# cp -r redis01/ redis08

[root@localhost redis-cluster]# cd redis08/

[root@localhost redis08]# sed -i "s/7001/7008/g" ./redis.conf

[root@localhost redis08]# ./redis-server redis.conf

   redis-trib增加从节点的命令为：

./redis-trib.rb add-node --slave --master-id $[nodeid] 127.0.0.1:7008 127.0.0.1:7000

    nodeid为要加到master主节点的node id，127.0.0.1:7008为新增的从节点，127.0.0.1:7000为集群的一个节点（集群的任意节点都行），用来辨识是哪个集群；如果没有给定那个主节点--master-id的话，redis-trib将会将新增的从节点随机到从节点较少的主节点上。

    现在我们添加一下7008，看是否会自动加到没有从节点的7007主节点上。

[root@localhost redis-cluster]# ./redis-trib.rb add-node --slave 127.0.0.1:7008 127.0.0.1:7001>>> Adding node 127.0.0.1:7008 to cluster 127.0.0.1:7001

>>> Performing Cluster Check (using node 127.0.0.1:7001)

M: dd19221c404fb2fc4da37229de56bab755c76f2b 127.0.0.1:7001

   slots:1365-5460 (4096 slots) master

   1 additional replica(s)

M: ee3efb90e5ac0725f15238a64fc60a18a71205d7 127.0.0.1:7007

   slots:0-1364,5461-6826,10923-12287 (4096 slots) master

   0 additional replica(s)

M: a5db243087d8bd423b9285fa8513eddee9bb59a6 127.0.0.1:7005

   slots:6827-10922 (4096 slots) master

   1 additional replica(s)

S: 8bb3ede48319b46d0015440a91ab277da9353c8b 127.0.0.1:7006

   slots: (0 slots) slave

   replicates f9886c71e98a53270f7fda961e1c5f730382d48f

M: f9886c71e98a53270f7fda961e1c5f730382d48f 127.0.0.1:7003

   slots:12288-16383 (4096 slots) master

   1 additional replica(s)

S: 1f07d76585bfab35f91ec711ac53ab4bc00f2d3a 127.0.0.1:7002

   slots: (0 slots) slave

   replicates a5db243087d8bd423b9285fa8513eddee9bb59a6

S: 50ce1ea59106b4c2c6bc502593a6a7a7dabf5041 127.0.0.1:7004

   slots: (0 slots) slave

   replicates dd19221c404fb2fc4da37229de56bab755c76f2b

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

Automatically selected master 127.0.0.1:7007

>>> Send CLUSTER MEET to node 127.0.0.1:7008 to make it join the cluster.

Waiting for the cluster to join.

>>> Configure node as replica of 127.0.0.1:7007.

[OK] New node added correctly.

[root@localhost redis-cluster]#

     可以看到自动选择了127.0.0.1:7007为master主节点，并且添加成功。

     可以check一下7008：

[root@localhost redis-cluster]# ./redis-trib.rb check 127.0.0.1:7008

>>> Performing Cluster Check (using node 127.0.0.1:7008)

S: 2ab1b061c36f30ae35604e9a171ae3afdc3c87e5 127.0.0.1:7008

   slots: (0 slots) slave

   replicates ee3efb90e5ac0725f15238a64fc60a18a71205d7

M: a5db243087d8bd423b9285fa8513eddee9bb59a6 127.0.0.1:7005

   slots:6827-10922 (4096 slots) master

   1 additional replica(s)

M: dd19221c404fb2fc4da37229de56bab755c76f2b 127.0.0.1:7001

   slots:1365-5460 (4096 slots) master

   1 additional replica(s)

S: 8bb3ede48319b46d0015440a91ab277da9353c8b 127.0.0.1:7006

   slots: (0 slots) slave

   replicates f9886c71e98a53270f7fda961e1c5f730382d48f

M: ee3efb90e5ac0725f15238a64fc60a18a71205d7 127.0.0.1:7007

   slots:0-1364,5461-6826,10923-12287 (4096 slots) master

   1 additional replica(s)

S: 50ce1ea59106b4c2c6bc502593a6a7a7dabf5041 127.0.0.1:7004

   slots: (0 slots) slave

   replicates dd19221c404fb2fc4da37229de56bab755c76f2b

M: f9886c71e98a53270f7fda961e1c5f730382d48f 127.0.0.1:7003

   slots:12288-16383 (4096 slots) master

   1 additional replica(s)

S: 1f07d76585bfab35f91ec711ac53ab4bc00f2d3a 127.0.0.1:7002

   slots: (0 slots) slave

   replicates a5db243087d8bd423b9285fa8513eddee9bb59a6

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

[root@localhost redis-cluster]#

       可以看到7008作为了7007的从节点。

       再测试一下指定主节点添加从节点，给7007增加7009从节点。

[root@localhost redis-cluster]# cp -r redis01/ redis09

[root@localhost redis-cluster]# cd redis09

[root@localhost redis09]# sed -i "s/7001/7009/g" ./redis.conf

[root@localhost redis09]# ./redis-server redis.conf

       添加7007主节点上

[root@localhost redis-cluster]# ./redis-trib.rb add-node --slave --master-id ee3efb90e5ac0725f15238a64fc60a18a71205d7 127.0.0.1:7009 127.0.0.1:7001

>>> Adding node 127.0.0.1:7009 to cluster 127.0.0.1:7001

>>> Performing Cluster Check (using node 127.0.0.1:7001)

M: dd19221c404fb2fc4da37229de56bab755c76f2b 127.0.0.1:7001

   slots:1365-5460 (4096 slots) master

   1 additional replica(s)

S: 2ab1b061c36f30ae35604e9a171ae3afdc3c87e5 127.0.0.1:7008

   slots: (0 slots) slave

   replicates ee3efb90e5ac0725f15238a64fc60a18a71205d7

M: ee3efb90e5ac0725f15238a64fc60a18a71205d7 127.0.0.1:7007

   slots:0-1364,5461-6826,10923-12287 (4096 slots) master

   1 additional replica(s)

M: a5db243087d8bd423b9285fa8513eddee9bb59a6 127.0.0.1:7005

   slots:6827-10922 (4096 slots) master

   1 additional replica(s)

S: 8bb3ede48319b46d0015440a91ab277da9353c8b 127.0.0.1:7006

   slots: (0 slots) slave

   replicates f9886c71e98a53270f7fda961e1c5f730382d48f

M: f9886c71e98a53270f7fda961e1c5f730382d48f 127.0.0.1:7003

   slots:12288-16383 (4096 slots) master

   1 additional replica(s)

S: 1f07d76585bfab35f91ec711ac53ab4bc00f2d3a 127.0.0.1:7002

   slots: (0 slots) slave

   replicates a5db243087d8bd423b9285fa8513eddee9bb59a6

S: 50ce1ea59106b4c2c6bc502593a6a7a7dabf5041 127.0.0.1:7004

   slots: (0 slots) slave

   replicates dd19221c404fb2fc4da37229de56bab755c76f2b

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

>>> Send CLUSTER MEET to node 127.0.0.1:7009 to make it join the cluster.

Waiting for the cluster to join.

>>> Configure node as replica of 127.0.0.1:7007.

[OK] New node added correctly.

[root@localhost redis-cluster]#

     显示从节点7009节点添加到7007主节点，可以看一下7007的从节点，如下：

[root@localhost redis-cluster]# cd ./redis07

[root@localhost redis07]# ./redis-cli -c -p 7007 cluster nodes | grep ee3efb90e5ac0725f15238a64fc60a18a71205d7

1f51443ede952b98724fea2a12f61fe710ab6cb1 127.0.0.1:7009 slave ee3efb90e5ac0725f15238a64fc60a18a71205d7 0 1462962710266 8 connected

ee3efb90e5ac0725f15238a64fc60a18a71205d7 127.0.0.1:7007 myself,master - 0 0 8 connected 0-1364 5461-6826 10923-12287

2ab1b061c36f30ae35604e9a171ae3afdc3c87e5 127.0.0.1:7008 slave ee3efb90e5ac0725f15238a64fc60a18a71205d7 0 1462962711607 8 connected

[root@localhost redis07]#

     maser 7007有2个slave 7008,7009。

     我们测试一下7007节点挂掉，看7008和7009那个成为主节点。

[root@localhost redis-cluster]# ps -ef | grep redis

root       7950      1  0 12:50 ?        00:02:05 ./redis-server 127.0.0.1:7001 [cluster]

root       7956      1  0 12:50 ?        00:02:11 ./redis-server 127.0.0.1:7003 [cluster]

root       7960      1  0 12:50 ?        00:01:47 ./redis-server 127.0.0.1:7004 [cluster]

root       7964      1  0 12:50 ?        00:02:07 ./redis-server 127.0.0.1:7005 [cluster]

root       7966      1  0 12:50 ?        00:01:46 ./redis-server 127.0.0.1:7006 [cluster]

root      12070      1  0 15:14 ?        00:01:08 ./redis-server 127.0.0.1:7002 [cluster]

root      13441      1  0 16:09 ?        00:01:25 ./redis-server 127.0.0.1:7007 [cluster]

root      15939      1  0 17:41 ?        00:00:20 ./redis-server 127.0.0.1:7008 [cluster]

root      16623      1  0 18:07 ?        00:00:10 ./redis-server 127.0.0.1:7009 [cluster]

root      17295  10581  0 18:37 pts/2    00:00:00 grep --color=auto redis

[root@localhost redis-cluster]# kill -9 13441

[root@localhost redis-cluster]# cd ./redis08

[root@localhost redis08]# ./redis-cli -c -p 7008

127.0.0.1:7008> get name

-> Redirected to slot [5798] located at 127.0.0.1:7009

"andy"

127.0.0.1:7009> cluster nodes

ee3efb90e5ac0725f15238a64fc60a18a71205d7 127.0.0.1:7007 master,fail - 1462963082317 1462963080194 8 disconnected

50ce1ea59106b4c2c6bc502593a6a7a7dabf5041 127.0.0.1:7004 slave dd19221c404fb2fc4da37229de56bab755c76f2b 0 1462963170968 1 connected

f9886c71e98a53270f7fda961e1c5f730382d48f 127.0.0.1:7003 master - 0 1462963168525 3 connected 12288-16383

dd19221c404fb2fc4da37229de56bab755c76f2b 127.0.0.1:7001 master - 0 1462963164466 1 connected 1365-5460

2ab1b061c36f30ae35604e9a171ae3afdc3c87e5 127.0.0.1:7008 slave 1f51443ede952b98724fea2a12f61fe710ab6cb1 0 1462963167508 9 connected

1f51443ede952b98724fea2a12f61fe710ab6cb1 127.0.0.1:7009 myself,master - 0 0 9 connected 0-1364 5461-6826 10923-12287

1f07d76585bfab35f91ec711ac53ab4bc00f2d3a 127.0.0.1:7002 slave a5db243087d8bd423b9285fa8513eddee9bb59a6 0 1462963170564 7 connected

8bb3ede48319b46d0015440a91ab277da9353c8b 127.0.0.1:7006 slave f9886c71e98a53270f7fda961e1c5f730382d48f 0 1462963167915 3 connected

a5db243087d8bd423b9285fa8513eddee9bb59a6 127.0.0.1:7005 master - 0 1462963169538 7 connected 6827-10922

127.0.0.1:7009>

      可以看到7009代替7007成了主节点。

       重启7007之后，会自动变成7009的从节点。

[root@localhost redis-cluster]# cd redis07

[root@localhost redis07]# ./redis-server redis.conf

[root@localhost redis07]# cd ../

[root@localhost redis-cluster]# ./redis-trib.rb check 127.0.0.1:7007

>>> Performing Cluster Check (using node 127.0.0.1:7007)

S: ee3efb90e5ac0725f15238a64fc60a18a71205d7 127.0.0.1:7007

   slots: (0 slots) slave

   replicates 1f51443ede952b98724fea2a12f61fe710ab6cb1

S: 50ce1ea59106b4c2c6bc502593a6a7a7dabf5041 127.0.0.1:7004

   slots: (0 slots) slave

   replicates dd19221c404fb2fc4da37229de56bab755c76f2b

M: 1f51443ede952b98724fea2a12f61fe710ab6cb1 127.0.0.1:7009

   slots:0-1364,5461-6826,10923-12287 (4096 slots) master

   2 additional replica(s)

S: 8bb3ede48319b46d0015440a91ab277da9353c8b 127.0.0.1:7006

   slots: (0 slots) slave

   replicates f9886c71e98a53270f7fda961e1c5f730382d48f

M: dd19221c404fb2fc4da37229de56bab755c76f2b 127.0.0.1:7001

   slots:1365-5460 (4096 slots) master

   1 additional replica(s)

M: a5db243087d8bd423b9285fa8513eddee9bb59a6 127.0.0.1:7005

   slots:6827-10922 (4096 slots) master

   1 additional replica(s)

S: 1f07d76585bfab35f91ec711ac53ab4bc00f2d3a 127.0.0.1:7002

   slots: (0 slots) slave

   replicates a5db243087d8bd423b9285fa8513eddee9bb59a6

M: f9886c71e98a53270f7fda961e1c5f730382d48f 127.0.0.1:7003

   slots:12288-16383 (4096 slots) master

   1 additional replica(s)

S: 2ab1b061c36f30ae35604e9a171ae3afdc3c87e5 127.0.0.1:7008

   slots: (0 slots) slave

   replicates 1f51443ede952b98724fea2a12f61fe710ab6cb1

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

[root@localhost redis-cluster]#

    验证了之前的测试。

# 六、节点的移除

   和节点添加一样，移除节点也有移除主节点，从节点。

## 1、移除主节点

       移除节点使用redis-trib的del-node命令，

redis-trib del-node 127.0.0.1:7002  ${node-id}

    127.0.0.1:7002位集群节点，node-id为要删除的主节点。 和添加节点不同，移除节点node-id是必需的，测试删除7001主节点：

[root@localhost redis-cluster]# ./redis-trib.rb del-node 127.0.0.1:7001 <span style="font-size: 14px;">dd19221c404fb2fc4da37229de56bab755c76f2b</span>

>>> Removing node <span style="font-size: 14px;">dd19221c404fb2fc4da37229de56bab755c76f2b</span> from cluster 127.0.0.1:7002

[ERR] Node 127.0.0.1:7001 is not empty! Reshard data away and try again.

[root@localhost redis-cluster]#

  redis cluster提示7001已经有数据了，不能够被删除，需要将他的数据转移出去，也就是和新增主节点一样需重新分片。

[root@localhost redis-cluster]# ./redis-trib.rb reshard 127.0.0.1:7002

  执行以后会提示我们移除的大小，因为7001占用了4096个槽点

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

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

  输入4096

  提示移动的node id，填写7009的node id。

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

What is the receiving node ID?

   需要移动到全部主节点上还是单个主节点

Please enter all the source node IDs.

  Type 'all' to use all the nodes as source nodes for the hash slots.

  Type 'done' once you entered all the source nodes IDs.

Source node #1:

   将4096个槽点移动到7009上，填写7001的node id ：dd19221c404fb2fc4da37229de56bab755c76f2b

Source node #1:dd19221c404fb2fc4da37229de56bab755c76f2b

Source node #2:done

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

   确认之后会一个一个将7001的卡槽移到到7009上。

[root@localhost redis-cluster]# ./redis-trib.rb check 127.0.0.1:7009

>>> Performing Cluster Check (using node 127.0.0.1:7009)

M: 1f51443ede952b98724fea2a12f61fe710ab6cb1 127.0.0.1:7009

   slots:0-6826,10923-12287 (8192 slots) master

   3 additional replica(s)

S: ee3efb90e5ac0725f15238a64fc60a18a71205d7 127.0.0.1:7007

   slots: (0 slots) slave

   replicates 1f51443ede952b98724fea2a12f61fe710ab6cb1

S: 50ce1ea59106b4c2c6bc502593a6a7a7dabf5041 127.0.0.1:7004

   slots: (0 slots) slave

   replicates 1f51443ede952b98724fea2a12f61fe710ab6cb1

M: f9886c71e98a53270f7fda961e1c5f730382d48f 127.0.0.1:7003

   slots:12288-16383 (4096 slots) master

   1 additional replica(s)

M: dd19221c404fb2fc4da37229de56bab755c76f2b 127.0.0.1:7001

   slots: (0 slots) master

   0 additional replica(s)

S: 2ab1b061c36f30ae35604e9a171ae3afdc3c87e5 127.0.0.1:7008

   slots: (0 slots) slave

   replicates 1f51443ede952b98724fea2a12f61fe710ab6cb1

S: 1f07d76585bfab35f91ec711ac53ab4bc00f2d3a 127.0.0.1:7002

   slots: (0 slots) slave

   replicates a5db243087d8bd423b9285fa8513eddee9bb59a6

S: 8bb3ede48319b46d0015440a91ab277da9353c8b 127.0.0.1:7006

   slots: (0 slots) slave

   replicates f9886c71e98a53270f7fda961e1c5f730382d48f

M: a5db243087d8bd423b9285fa8513eddee9bb59a6 127.0.0.1:7005

   slots:6827-10922 (4096 slots) master

   1 additional replica(s)

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

[root@localhost redis-cluster]#

     可以看到7001有0个卡槽，而7009有8192个卡槽。

     在执行移除操作

[root@localhost redis-cluster]# ./redis-trib.rb del-node 127.0.0.1:7002 dd19221c404fb2fc4da37229de56bab755c76f2b

>>> Removing node dd19221c404fb2fc4da37229de56bab755c76f2b from cluster 127.0.0.1:7002

>>> Sending CLUSTER FORGET messages to the cluster...

>>> SHUTDOWN the node.

[root@localhost redis-cluster]#

    已经删除了7001节点。

[root@localhost redis-cluster]# ./redis-trib.rb check 127.0.0.1:7001

[ERR] Sorry, can't connect to node 127.0.0.1:7001

[root@localhost redis-cluster]# ./redis-trib.rb check 127.0.0.1:7009

>>> Performing Cluster Check (using node 127.0.0.1:7009)

M: 1f51443ede952b98724fea2a12f61fe710ab6cb1 127.0.0.1:7009

   slots:0-6826,10923-12287 (8192 slots) master

   3 additional replica(s)

S: ee3efb90e5ac0725f15238a64fc60a18a71205d7 127.0.0.1:7007

   slots: (0 slots) slave

   replicates 1f51443ede952b98724fea2a12f61fe710ab6cb1

S: 50ce1ea59106b4c2c6bc502593a6a7a7dabf5041 127.0.0.1:7004

   slots: (0 slots) slave

   replicates 1f51443ede952b98724fea2a12f61fe710ab6cb1

M: f9886c71e98a53270f7fda961e1c5f730382d48f 127.0.0.1:7003

   slots:12288-16383 (4096 slots) master

   1 additional replica(s)

S: 2ab1b061c36f30ae35604e9a171ae3afdc3c87e5 127.0.0.1:7008

   slots: (0 slots) slave

   replicates 1f51443ede952b98724fea2a12f61fe710ab6cb1

S: 1f07d76585bfab35f91ec711ac53ab4bc00f2d3a 127.0.0.1:7002

   slots: (0 slots) slave

   replicates a5db243087d8bd423b9285fa8513eddee9bb59a6

S: 8bb3ede48319b46d0015440a91ab277da9353c8b 127.0.0.1:7006

   slots: (0 slots) slave

   replicates f9886c71e98a53270f7fda961e1c5f730382d48f

M: a5db243087d8bd423b9285fa8513eddee9bb59a6 127.0.0.1:7005

   slots:6827-10922 (4096 slots) master

   1 additional replica(s)

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

[root@localhost redis-cluster]#

  可以看到7001已经连接不了；而7001的从节点7004自动分配到了7009主节点中，7009现在3个从节点。

## 2、移除从节点

     比如删除7009的7008节点：

[root@localhost redis-cluster]# ./redis-trib.rb del-node 127.0.0.1:7009 2ab1b061c36f30ae35604e9a171ae3afdc3c87e5

>>> Removing node 2ab1b061c36f30ae35604e9a171ae3afdc3c87e5 from cluster 127.0.0.1:7009

>>> Sending CLUSTER FORGET messages to the cluster...

>>> SHUTDOWN the node.

[root@localhost redis-cluster]# ./redis-trib.rb check 127.0.0.1:7008

[ERR] Sorry, can't connect to node 127.0.0.1:7008

[root@localhost redis-cluster]#

     删除从节点比较方便，现在redis-cluster中有3个主节点，4个从节点，如下：

[root@localhost redis-cluster]# ./redis-trib.rb check 127.0.0.1:7009

>>> Performing Cluster Check (using node 127.0.0.1:7009)

M: 1f51443ede952b98724fea2a12f61fe710ab6cb1 127.0.0.1:7009

   slots:0-6826,10923-12287 (8192 slots) master

   2 additional replica(s)

S: ee3efb90e5ac0725f15238a64fc60a18a71205d7 127.0.0.1:7007

   slots: (0 slots) slave

   replicates 1f51443ede952b98724fea2a12f61fe710ab6cb1

S: 50ce1ea59106b4c2c6bc502593a6a7a7dabf5041 127.0.0.1:7004

   slots: (0 slots) slave

   replicates 1f51443ede952b98724fea2a12f61fe710ab6cb1

M: f9886c71e98a53270f7fda961e1c5f730382d48f 127.0.0.1:7003

   slots:12288-16383 (4096 slots) master

   1 additional replica(s)

S: 1f07d76585bfab35f91ec711ac53ab4bc00f2d3a 127.0.0.1:7002

   slots: (0 slots) slave

   replicates a5db243087d8bd423b9285fa8513eddee9bb59a6

S: 8bb3ede48319b46d0015440a91ab277da9353c8b 127.0.0.1:7006

   slots: (0 slots) slave

   replicates f9886c71e98a53270f7fda961e1c5f730382d48f

M: a5db243087d8bd423b9285fa8513eddee9bb59a6 127.0.0.1:7005

   slots:6827-10922 (4096 slots) master

   1 additional replica(s)

[OK] All nodes agree about slots configuration.

>>> Check for open slots...

>>> Check slots coverage...

[OK] All 16384 slots covered.

[root@localhost redis-cluster]#

# 七、其他

redis集群重新部署需要删除各个节点下的文件，删除命令如下:

rm -rf appendonly.aof | rm -rf dump.rdb | rm -rf nodes.conf | rm -rf redis.log

如果集群要设置密码,集群启动后要在各个节点下设置:

config set masterauth xxx

config set requirepass xxx

然后在各个节点下重置配置:

config rewrite