# 3master-3slave Redis集群方案

[**Redis 3.2.1集群搭建**](https://www.cnblogs.com/yuanermen/p/5717885.html)

**一、概述**

    Redis3.0版本之后支持Cluster.

1.1、redis cluster的现状

 　　目前redis支持的cluster特性：

　　1):节点自动发现

　　2):slave->master 选举,集群容错

　　3):Hot resharding:在线分片

　　4):进群管理:cluster xxx

　　5):基于配置(nodes-port.conf)的集群管理

　　6):ASK 转向/MOVED 转向机制.

1.2、redis cluster 架构

　　1)redis-cluster架构图



　　架构细节:

　　(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,cluster 负责维护node<->slot<->value

   2) redis-cluster选举:容错



　　(1)领着选举过程是集群中所有master参与,如果半数以上master节点与master节点通信超过(cluster-node-timeout),认为当前master节点挂掉.

　　(2):什么时候整个集群不可用(cluster\_state:fail),当集群不可用时,所有对集群的操作做都不可用，收到((error) CLUSTERDOWN The cluster is down)错误

    　　a:如果集群任意master挂掉,且当前master没有slave.集群进入fail状态,也可以理解成进群的slot映射[0-16383]不完成时进入fail状态.

    　　b:如果进群超过半数以上master挂掉，无论是否有slave集群进入fail状态.

**二、redis cluster安装**

1、下载和解包

|  |  |
| --- | --- |
| 1  2  3 | cd /usr/local/  wget http://download.redis.io/releases/redis-3.2.1.tar.gz  tar -zxvf /redis-3.2.1.tar.gz |

　2、 编译安装

cd redis-3.2.1

make && make install

  3、创建redis节点

     测试我们选择2台服务器，分别为：192.168.1.237，192.168.1.238.每分服务器有3个节点。

  我先在192.168.1.237创建3个节点：

[复制代码](javascript:void(0);)

cd /usr/local/

mkdir redis\_cluster //创建集群目录

mkdir 7000 7001 7002 //分别代表三个节点 其对应端口 7000 7001 7002

//创建7000节点为例，拷贝到7000目录

cp /usr/local/redis-3.2.1/redis.conf ./redis\_cluster/7000/

//拷贝到7001目录

cp /usr/local/redis-3.2.1/redis.conf ./redis\_cluster/7001/

//拷贝到7002目录

cp /usr/local/redis-3.2.1/redis.conf ./redis\_cluster/7002/

[复制代码](javascript:void(0);)

   分别对7001，7002、7003文件夹中的3个文件修改对应的配置

[复制代码](javascript:void(0);)

daemonize yes //redis后台运行

pidfile /var/run/redis\_7000.pid //pidfile文件对应7000,7002,7003

port 7000 //端口7000,7002,7003

cluster-enabled yes //开启集群 把注释#去掉

cluster-config-file nodes\_7000.conf //集群的配置 配置文件首次启动自动生成 7000,7001,7002

cluster-node-timeout 5000 //请求超时 设置5秒够了

appendonly yes //aof日志开启 有需要就开启，它会每次写操作都记录一条日志

[复制代码](javascript:void(0);)

   在192.168.1.238创建3个节点：对应的端口改为7003,7004,7005.配置对应的改一下就可以了。

   4、两台机启动各节点(两台服务器方式一样)

[复制代码](javascript:void(0);)

cd /usr/local

redis-server redis\_cluster/7000/redis.conf

redis-server redis\_cluster/7001/redis.conf

redis-server redis\_cluster/7002/redis.conf

redis-server redis\_cluster/7003/redis.conf

redis-server redis\_cluster/7004/redis.conf

redis-server redis\_cluster/7005/redis.conf

[复制代码](javascript:void(0);)

   5、查看服务

      ps -ef | grep redis   #查看是否启动成功

     netstat -tnlp | grep redis #可以看到redis监听端口

**三、创建集群**

  前面已经准备好了搭建集群的redis节点，接下来我们要把这些节点都串连起来搭建集群。官方提供了一个工具：redis-trib.rb(/usr/local/redis-3.2.1/src/redis-trib.rb) 看后缀就知道这鸟东西不能直接执行，它是用ruby写的一个程序，所以我们还得安装ruby.

yum -y install ruby ruby-devel rubygems rpm-build

  再用 gem 这个命令来安装 redis接口    gem是ruby的一个工具包.

gem install redis //等一会儿就好了

当然，方便操作，两台Server都要安装。

  上面的步骤完事了，接下来运行一下redis-trib.rb

[复制代码](javascript:void(0);)

/usr/local/redis-3.2.1/src/redis-trib.rb

   Usage: redis-trib <command> <options> <arguments ...>

   reshard        host:port  
                  --to <arg>  
                  --yes  
                  --slots <arg>  
                  --from <arg>  
  check          host:port  
  call            host:port command arg arg .. arg  
  set-timeout    host:port milliseconds  
  add-node        new\_host:new\_port existing\_host:existing\_port  
                  --master-id <arg>  
                  --slave  
  del-node        host:port node\_id  
  fix            host:port  
  import          host:port  
                  --from <arg>  
  help            (show this help)  
  create          host1:port1 ... hostN:portN  
                  --replicas <arg>

For check, fix, reshard, del-node, set-timeout you can specify the host and port of any working node in the cluster.

[复制代码](javascript:void(0);)

     看到这，应该明白了吧， 就是靠上面这些操作 完成redis集群搭建的.

 确认所有的节点都启动，接下来使用参数create 创建 (在192.168.1.237中来创建)

/usr/local/redis-3.2.1/src/redis-trib.rb create --replicas 1 192.168.1.237:7000 192.168.1.237:7001 192.168.1.237:7003 192.168.1.238:7003 192.168.1.238:7004 192.168.1.238:7005

    解释下， --replicas  1  表示 自动为每一个master节点分配一个slave节点    上面有6个节点，程序会按照一定规则生成 3个master（主）3个slave(从)

    前面已经提醒过的 防火墙一定要开放监听的端口，否则会创建失败。

 运行中，提示Can I set the above configuration? (type 'yes' to accept): yes    //输入yes

 接下来 提示  Waiting for the cluster to join..........  安装的时候在这里就一直等等等，没反应，傻傻等半天，看这句提示上面一句，Sending Cluster Meet Message to join the Cluster.

    这下明白了，我刚开始在一台Server上去配，也是不需要等的，这里还需要跑到Server2上做一些这样的操作。

    在192.168.1.238, redis-cli -c -p 700\*  分别进入redis各节点的客户端命令窗口， 依次输入 cluster meet 192.168.1.238 7000……

    回到Server1，已经创建完毕了。

    查看一下 /usr/local/redis/src/redis-trib.rb check 192.168.1.237:7000

    到这里集群已经初步搭建好了。

**四、测试**

1）get 和 set数据

    redis-cli -c -p 7000

    进入命令窗口，直接 set  hello  howareyou

    直接根据hash匹配切换到相应的slot的节点上。

    还是要说明一下，redis集群有16383个slot组成，通过分片分布到多个节点上，读写都发生在master节点。

  2）假设测试

    果断先把192.168.1.238服务Down掉，（192.168.1.238有1个Master, 2个Slave） ,  跑回192.168.1.238, 查看一下 发生了什么事，192.168.1.237的3个节点全部都是Master，其他几个Server2的不见了

    测试一下，依然没有问题，集群依然能继续工作。

    原因：  redis集群  通过选举方式进行容错，保证一台Server挂了还能跑，这个选举是全部集群超过半数以上的Master发现其他Master挂了后，会将其他对应的Slave节点升级成Master.

    疑问： 要是挂的是192.168.1.237怎么办？    哥试了，cluster is down!!    没办法，超过半数挂了那救不了了，整个集群就无法工作了。 要是有三台Server，每台两Master，切记对应的主从节点

            不要放在一台Server,别问我为什么自己用脑子想想看，互相交叉配置主从，挂哪台也没事，你要说同时两台crash了，呵呵哒......

  3）关于一致性

    我还没有这么大胆拿redis来做数据库持久化哥网站数据，只是拿来做cache，官网说的很清楚，Redis Cluster is not able to guarantee strong consistency.

**五、安装遇到的问题**

1、

　　CC adlist.o  
　　/bin/sh: cc: command not found  
　　make[1]: \*\*\* [adlist.o] Error 127  
　　make[1]: Leaving directory `/usr/local/redis-3.2.1/src  
　　make: \*\*\* [all] Error 2

     解决办法：GCC没有安装或版本不对，安装一下

yum install gcc

   2、

　　zmalloc.h:50:31:   
　　error: jemalloc/jemalloc.h: No such file or directory  
　　zmalloc.h:55:2: error:   
  
　　#error "Newer version of jemalloc required"  
　　make[1]: \*\*\* [adlist.o] Error   
　　1  
　　make[1]: Leaving directory `/data0/src/redis-2.6.2/src  
　　make: \*\*\* [all]   
　　Error 2

    解决办法：原因是没有安装jemalloc内存分配器，可以安装jemalloc 或 直接

     输入make MALLOC=libc  && make install

记住该记住的,忘记该忘记的,改变能改变的,接受不能改变的!