CentOS7 下搭建Redis3.2.6集群环境比较简单

1、下载[Redis](http://lib.csdn.net/base/redis" \o "Redis知识库" \t "http://blog.csdn.net/weiguolong0306/article/details/_blank)-3.2.6.tar.gz

上传以上两个文件到/usr/local/src/ ， 运行如下命令：

tar -zxvf redis-3.2.6.tar.gz

cd redis-3.2.6/

mkdir  -p /usr/local/cluster (cluster目录是用来放redis集群的**[测试](http://lib.csdn.net/base/softwaretest" \o "软件测试知识库" \t "http://blog.csdn.net/weiguolong0306/article/details/_blank)**目录)

make MALLOC=libc

make install PREFIX=/usr/local/cluster

注意：首先需要按装GCC， yum install gcc

2、创建节点机器目录

cp /usr/local/src/redis-3.2.0/redis.conf  /usr/local/cluster

cd /usr/local/cluster

mv bin/  7000

cp -rf 7000 7001 ; cp -rf 7000 7002; cp -rf 7000 7003 ;cp -rf 7000 7004 ;cp -rf 7000 7005

3、编辑配置文件

vim /usr/local/cluster/redis.conf , 修改的数据如下：

port 7000

cluster-enabled yes

appendonly yes

cluster-config-file nodes.conf

cluster-node-timeout 5000

完成之后，保存退出，并将reids.conf配置文件分别复制到以上700X的目录中，并修改port为对应的目录名称。

4、安装配置工具

yum install ruby

yum install rubygems

gem install redis

5、启动节点服务

cd /usr/local/cluster/7000 ; ./redis-server redis.conf &

cd /usr/local/cluster/7001 ; ./redis-server redis.conf &

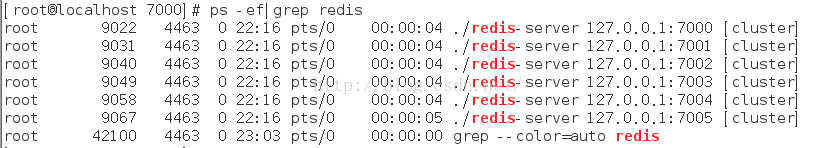
cd /usr/local/cluster/7002 ; ./redis-server redis.conf &

cd /usr/local/cluster/7003; ./redis-server redis.conf &

cd /usr/local/cluster/7004; ./redis-server redis.conf &

cd /usr/local/cluster/7005 ; ./redis-server redis.conf &

ps -ef|grep redis, 如果出现以下图片内容，则说明OK!



6、配置redis集群

cd /usr/local/src/redis-3.2.6/src

./redis-trib.rb create --replicas 1 192.168.0.189:7000192.168.0.189:7001 192.168.0.189:7002 192.168.0.189:7003 192.168.0.189:7004 192.168.0.189:7005

如出现如下所示，则表示集群搭建OK:



1. 检查redis集群连接

./redis-cli -h 192.168.0.189 -p 7000

flushall

# redis优化配置和redis.conf说明

时间 2014-10-29 01:01:30 [CSDN博客](http://www.tuicool.com/sites/vQZRB3" \t "http://www.tuicool.com/articles/_blank)

*原文*  [http://blog.csdn.net/luozhonghua2014/article/details/40568707](http://blog.csdn.net/luozhonghua2014/article/details/40568707?utm_source=tuicool&utm_medium=referral)

主题 [Redis](http://www.tuicool.com/topics/11030032" \t "http://www.tuicool.com/articles/_blank)

1. redis.conf 配置参数：

#是否作为守护进程运行

daemonize yes

#如以后台进程运行，则需指定一个pid，默认为/var/run/redis.pid

pidfile redis.pid

#绑定主机IP，默认值为127.0.0.1

#bind 127.0.0.1

#Redis默认监听端口

port 6379

#客户端闲置多少秒后，断开连接，默认为300（秒）

timeout 300

#日志记录等级，有4个可选值，debug，verbose（默认值），notice，warning

loglevel verbose

#指定日志输出的文件名，默认值为stdout，也可设为/dev/null屏蔽日志

logfile stdout

#可用数据库数，默认值为16，默认数据库为0

databases 16

#保存数据到disk的策略

#当有一条Keys数据被改变是，900秒刷新到disk一次

save 900 1

#当有10条Keys数据被改变时，300秒刷新到disk一次

save 300 10

#当有1w条keys数据被改变时，60秒刷新到disk一次

save 60 10000

#当dump .rdb数据库的时候是否压缩数据对象

rdbcompression yes

#本地数据库文件名，默认值为dump.rdb

dbfilename dump.rdb

#本地数据库存放路径，默认值为 ./

dir /var/lib/redis/

########### Replication #####################

#Redis的复制配置

# slaveof <masterip> <masterport> 当本机为从服务时，设置主服务的IP及端口

# masterauth <master-password> 当本机为从服务时，设置主服务的连接密码

#连接密码

# requirepass foobared

#最大客户端连接数，默认不限制

# maxclients 128

#最大内存使用设置，达到最大内存设置后，Redis会先尝试清除已到期或即将到期的Key，当此方法处理后，任到达最大内存设置，将无法再进行写入操作。

# maxmemory <bytes>

#是否在每次更新操作后进行日志记录，如果不开启，可能会在断电时导致一段时间内的数据丢失。因为redis本身同步数据文件是按上面save条件来同步的，所以有的数据会在一段时间内只存在于内存中。默认值为no

appendonly no

#更新日志文件名，默认值为appendonly.aof

#appendfilename

#更新日志条件，共有3个可选值。no表示等操作系统进行数据缓存同步到磁盘，always表示每次更新操作后手动调用fsync()将数据写到磁盘，everysec表示每秒同步一次（默认值）。

# appendfsync always

appendfsync everysec

# appendfsync no

################ VIRTUAL MEMORY ###########

#是否开启VM功能，默认值为no

vm-enabled no

# vm-enabled yes

#虚拟内存文件路径，默认值为/tmp/redis.swap，不可多个Redis实例共享

vm-swap-file /tmp/redis.swap

# 将所有大于vm-max-memory的数据存入虚拟内存,无论vm-max-memory设置多小,所有索引数据都是内存存储的 (Redis的索引数据就是keys),也就是说,当vm-max-memory设置为0的时候,其实是所有value都存在于磁盘。默认值为0。

vm-max-memory 0

vm-page-size 32

vm-pages 134217728

vm-max-threads 4

############# ADVANCED CONFIG ###############

glueoutputbuf yes

hash-max-zipmap-entries 64

hash-max-zipmap-value 512

#是否重置Hash表

activerehashing yes

注意：Redis官方文档对VM的使用提出了一些建议:

当你的key很小而value很大时,使用VM的效果会比较好.因为这样节约的内存比较大.

当你的key不小时,可以考虑使用一些非常方法将很大的key变成很大的value,比如你可以考虑将key,value组合成一个新的value.

最好使用linux ext3 等对稀疏文件支持比较好的文件系统保存你的swap文件.

vm-max-threads这个参数,可以设置访问swap文件的线程数,设置最好不要超过机器的核数.如果设置为0,那么所有对swap文件的操作都是串行的.可能会造成比较长时间的延迟,但是对数据完整性有很好的保证.

2. 调整系统内核参数

如果内存情况比较紧张的话，需要设定内核参数：

echo 1 > /proc/sys/vm/overcommit\_memory

这里说一下这个配置的含义：/proc/sys/vm/overcommit\_memory

该文件指定了内核针对内存分配的策略，其值可以是0、1、2。

0，表示内核将检查是否有足够的可用内存供应用进程使用；如果有足够的可用内存，内存申请允许；否则，内存申请失败，并把错误返回给应用进程。

1，表示内核允许分配所有的物理内存，而不管当前的内存状态如何。

2，表示内核允许分配超过所有物理内存和交换空间总和的内存

Redis 在dump数据的时候，会fork出一个子进程，理论上child进程所占用的内存和parent是一样的，比如parent占用的内存为 8G，这个时候也要同样分配8G的内存给child, 如果内存无法负担，往往会造成redis服务器的down机或者IO负载过高，效率下降。所以这里比较优化的内存分配策略应该设置为 1（表示内核允许分配所有的物理内存，而不管当前的内存状态如何）

一、 运行服务

# redis-server /etc/redis/redis.conf 开启

# redis-cli shutdown 关闭

二、 测试

1) 可在后台启动redis服务后，用redis-benchmark命令测试

2) 通过redis-cli命令实际操作测试

三、 保存/备份

数据备份可以通过定期备份该文件实现。

因为redis是异步写入磁盘的，如果要让内存中的数据马上写入硬盘可以执行如下命令：

redis-cli save 或者 redis-cli -p 6380 save（指定端口）

注意，以上部署操作需要具备一定的权限，比如复制和设定内核参数等。

执行redis-benchmark命令时也会将内存数据写入硬盘。

四、 开启端口号

1) 打开/etc/sysconfig/iptables，

2) 在-【A INPUT -p tcp -m state --state NEW -m tcp --dport 22 -j ACCEPT】后面，加上 【-A INPUT -p tcp -m state --state NEW -m tcp --dport 6379 -j ACCEPT】  //这里的6379是Redis默认端口号

3) 保存，重启防火墙：/etc/init.d/iptables restart

redis 官方提供的 conf

https://raw.github.com/antirez/redis/2.2/redis.conf

中6中过期策略的具体方式。  
redis 中的默认的过期策略是volatile-lru 。设置方式

config set maxmemory-policy volatile-lru

maxmemory-policy 六种方式

1. volatile-lru：只对设置了过期时间的key进行LRU（默认值）
2. allkeys-lru ： 删除lru算法的key
3. volatile-random：随机删除即将过期key
4. allkeys-random：随机删除
5. volatile-ttl ： 删除即将过期的
6. noeviction ： 永不过期，返回错误