**redis安装**

**redis软件安装**

**官网上的安装步骤**

Installation

Download, extract and compile Redis with:

$ wget http://download.redis.io/releases/redis-4.0.1.tar.gz //可离线下载。

$ tar xzf redis-4.0.1.tar.gz //建议将安装包移动到自己指定的目录，在该目录解压。

$ cd redis-4.0.1

$ make

The binaries that are now compiled are available in the src directory.

**创建redis需要的文件夹**

# cd /usr

#mkdir all

#cd all

#mkdir redis

#cd redis

#mkdir conf

#mkdir clusters

#cd clusters

#mkdir 7001 [7002、7003、7004、7005、7006] //建立6个文件夹，对应六个节点。

**添加redis路径到系统路径**

vi ~/.bashrc

REDIS\_HOME=/usr/master/a\_run/d\_service/redis/redis-4.0.1

PATH=${REDIS\_HOME}/src:$PATH

//安装(解压)地址。

source ~/.bashrc

**启动redis服务**

Run Redis with:

$ redis-server

$ redis-server 7001.conf

You can interact with Redis using the built-in client:

**停止redis服务**

redis-cli -p 7001 shutdown

**使用redis的客户端使用服务**

$ src/redis-cli

redis> set foo bar

OK

redis> get foo

"bar"

**分布式安装**

**创建集群节点**

**创建redis节点配置文件**

测试我们选择3台服务器，分别为：192.168.221.136，192.168.175.233, 192.168.175.244每分服务器有6个节点。

我先在192.168.221.136创建6个节点：

cd /usr/all

mkdir conf //创建集群目录

配置7001配置文件为例:

cd /usr/master/a\_run/d\_service/redis/redis-4.0.1 //进入redis安装(解压)目录。

cp redis.conf /usr/all/redis/conf/redis\_7001.conf

修改其中的配置：

daemonize yes //redis后台运行

pidfile /usr/all/redis/clusters/7001/redis.pid //pidfile文件对应7000,7002,7003

bind 192.168.221.168 //绑定需要访问这个节点的主机IP地址。

port 7001 //端口7000,7002,7003

cluster-enabled yes //开启集群 把注释#去掉

cluster-config-file nodes.conf //集群配置，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，它每次写操作都记录一条日志

拷贝7001的配置，然后稍作修改。这4个文件修改对应的配置。

cp redis\_7001.conf redis\_7002.conf

cp redis\_7001.conf redis\_7003.conf

cp redis\_7001.conf redis\_7004.conf

cp redis\_7001.conf redis\_7005.conf

cp redis\_7001.conf redis\_7006.conf

同样的方法在其他机器上再创建节点：对应的端口改为7005,7006,...配置对应的改一下就可以了。

**启动各个节点**

几台机器启动各节点(几台服务器方式一样)

master机器上：

cd /usr/all/redis

cd clusters/7001

redis-server redis\_7001.conf

注意：在执行redis-server 启动服务的目录下，会生成报告文件等，为了不混乱，这里每个节点新建一个目录。

slaveC、slaveD机器：

cd /usr/all/redis

cd clusters/7011

redis-server redis\_7011.conf

为了方便启动，可以写一个sh脚本runserver.sh放到redis目录下:

runserver.sh:

cd clusters/7001

redis-server redis\_7001.conf

cd ../../

...后面几条类似，改一下7001就行。你可以写好后存起来，方便其他节点使用，我放在共享文件夹了，跟配置放在一起。

**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

**安装ruby**

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

本机：sudo apt-get install ruby2.0

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

/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.

**创建集群**

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)

执行redis-trib.rb脚本，创建集群。

redis-trib.rb create --replicas 1 192.168.221.168:7001 192.168.221.168:7002 192.168.221.168:7003 192.168.221.168:7004 192.168.221.168:7005 192.168.221.168:7006

  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)

集群新建时候遇到的问题：

①：异常信息:

Node 192.168.221.168:7001 is not empty. Either the node already knows other nodes (check with CLUSTER NODES) or contains some key in database 0.

② ：异常原因：如果新集群的节点以前创建过集群，会在每个节点下创建dump.rdb和nodes.conf，这会给新集群带来异常。

③ ：解释一下dump.rdb: dump.rdb是甶Redis服务器自动生成的，默认情 况下，每隔一段时间redis服务器会自动对数据库进行遍历，把内存快照写在 -个叫"dump.rdb^文件里，这个持久化机制叫做：SNAPSHOT (快照的意 思，快速拍照）有了 SNAPSHOT后，如果服务器挂了，坏了，重启redis 服务器程序的时候，会自动恢复到上一次SNAPSHOT的状态，类似于隔段 时间进行备份数据。

④ ：解决办法：将每个虚拟节点下的dump.rdb、appendonly.aof和nodes.conf删除，因为 redis-cluster (redis集群）需要一个干净的环境。此处可以编写脚本实现， ./clustersclean.sh

⑤ ：删除完毕后，全部重启，./clustersrun.sh

⑥ ：重启完之后，重新连接，即可。

⑦ ：再次添加新节点如果还是报错，则登录新Node,./redis-cli -h ip -p port对数据库进行清除：

redis-cli -h 192.168.221.168 -p 7001

192.168.221.168:7001> flushdb #清空当前数据

**注意点**

  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 7001……

    回到Server1，已经创建完毕了。

    查看一下 redis-trib.rb check 192.168.211.136:7001

    到这里集群已经初步搭建好了。

redis-cli -c -h 192.168.221.168 -p 7001 cluster meet 192.168.221.168 7005

redis-trib.rb create --replicas 1 192.168.221.168:7001 192.168.221.168:7002 192.168.221.168:7003 192.168.221.168:7004 192.168.221.168:7005 192.168.221.168:7006

redis-trib.rb create --replicas 1 192.168.221.153:7001 192.168.221.153:7002 192.168.221.153:7003 192.168.221.153:7004 192.168.221.153:7005 192.168.221.153:7006

redis-trib.rb create --replicas 1 192.168.221.154:7001 192.168.221.154:7002 192.168.221.154:7003 192.168.221.154:7004 192.168.221.154:7005 192.168.221.154:7006