redis3.0.2 分布式集群安装详细步骤

--(centos5.8 X64 系统)

版本历史

时间	版本	说明	编写者
2015-06-5	1.0	redis3.0.2 分布式集群安装详细步骤	崔四超

一: redis cluster 介绍篇

1:redis cluster 的现状

目前 redis 支持的 cluster 特性(已亲测):

1):节点自动发现

2):slave->master 选举,集群容错

3):Hot resharding:在线分片

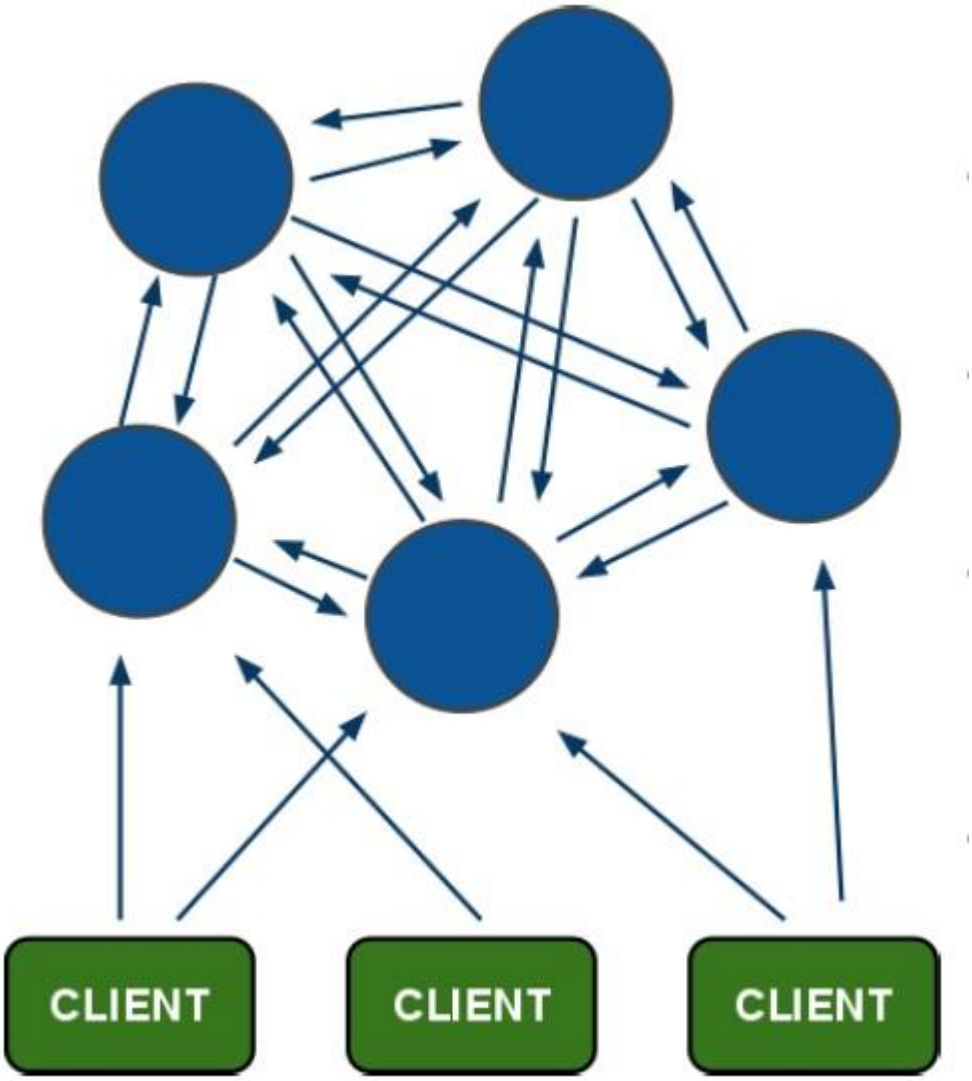
4):进群管理:cluster xxx

5):基于配置(nodes-port.conf)的集群管理

6):ASK 转向/MOVED 转向机制.

2:redis cluster 架构

1)redis-cluster 架构图



架构细节:

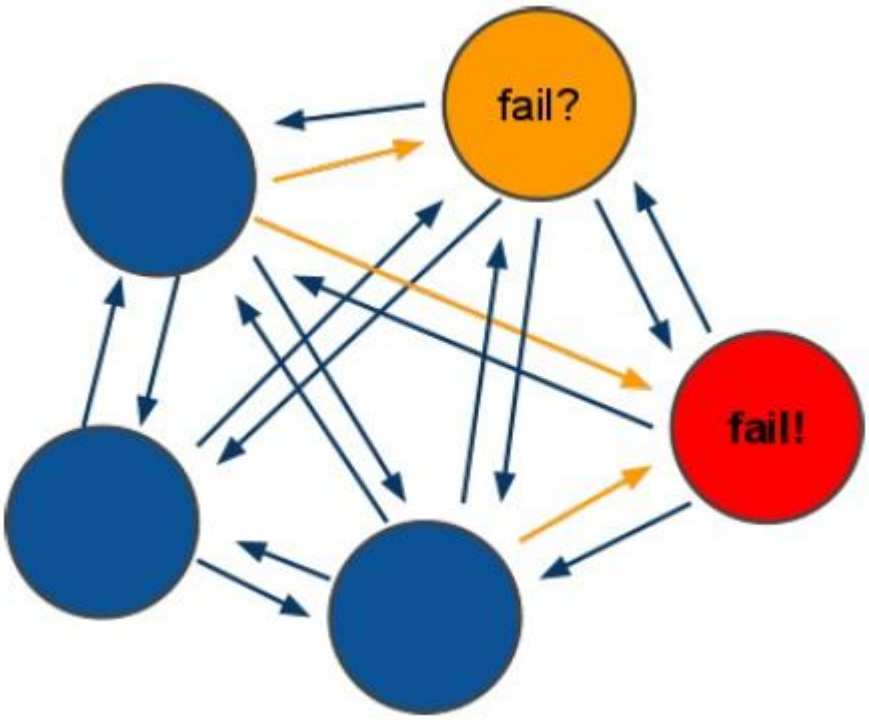
(1)所有的 redis 节点彼此互联(PING-PONG 机制),内部使用二进制协议优化传输速度和带宽.

(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 集群安装篇(centos5.8 X64 系统)

（要让集群正常工作至少需要 3 个主节点，在这里我们要创建 6 个 redis 节点，其中三个为主节点,三个为从节点,对应的 redis 节点的 ip 和端口对应关系如下）

127.0.0.1:7000

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

1：下载 redis。

官网下载 3.0.0 版本，之前 2.几的版本不支持集群模式

下载地址：http://download.redis.io/releases/redis-3.0.2.tar.gz

2：上传服务器，解压，编译

tar -zxvf redis-3.0.2.tar.gz.tar.gz

mv redis-3.0.2.tar.gz.tar.gz redis3.0

cd /usr/local/redis3.0

make

make install

3：创建集群需要的目录

mkdir -p /usr/local/cluster

cd /usr/local/cluster

mkdir 7000

mkdir 7001

mkdir 7002

mkdir 7003

mkdir 7004

mkdir 7005

4：修改配置文件 redis.conf

cp /usr/local/redis3.0/redis.conf /usr.local/cluster

vi redis.conf

##修改配置文件中的下面选项

port 7000

daemonize yes

cluster-enabled yes

cluster-config-file nodes.conf

cluster-node-timeout 5000

appendonly yes

##修改完 redis.conf 配置文件中的这些配置项之后把这个配置文件分别拷贝到 7000/7001/7002/7003/7004/7005 目录下面

cp /usr/local/cluster/redis.conf /usr/local/cluster/7000

cp /usr/local/cluster/redis.conf /usr/local/cluster/7001

cp /usr/local/cluster/redis.conf /usr/local/cluster/7002

cp /usr/local/cluster/redis.conf /usr/local/cluster/7003

cp /usr/local/cluster/redis.conf /usr/local/cluster/7004

cp /usr/local/cluster/redis.conf /usr/local/cluster/7005

##注意：拷贝完成之后要修改 7001/7002/7003/7004/7005 目录下面 redis.conf 文件中的 port 参数，分别改为对应的文件夹的名称

5：分别启动这 6 个 redis 实例

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

##启动之后使用命令查看 redis 的启动情况 ps -ef|grep redis

如下显示则说明启动成功

```
 # ps -ef|grep redis
root      13703     1  0 10:03 ?        00:00:00 redis-server *:7000 [cluster]
root      14015     1  0 10:04 ?        00:00:00 redis-server *:7002 [cluster]
root      14133     1  0 10:04 ?        00:00:00 redis-server *:7003 [cluster]
root      14172     1  0 10:04 ?        00:00:00 redis-server *:7004 [cluster]
root      14187     1  0 10:04 ?        00:00:00 redis-server *:7005 [cluster]
root      14323     1  0 10:04 ?        00:00:00 redis-server *:7001 [cluster]
```

6.升级 ruby 安装 gem

安装gem 需要ruby的版本在 1.8.7 以上，默认的centos5 上都是1.8.5 版本，所以首先你的升级你的 ruby ，

rpm -ivh http://yum.puppetlabs.com/el/5/products/x86_64/puppetlabs-release-5-6.noarch.rpm

yum install ruby ruby-devel rubygems rpm-build

检查 ruby 版本：

#ruby -v

ruby 1.8.7 (2013-06-27 patchlevel 374) [x86_64-linux]

是否安装 rubygems：

rpm -qa|grep ruby

ruby-rdoc-1.8.7.374-2.el5

ruby-1.8.7.374-2.el5

ruby-devel-1.8.7.374-2.el5

ruby-devel-1.8.7.374-2.el5

ruby-mode-1.8.5-24.el5

ruby-irb-1.8.7.374-2.el5

ruby-libs-1.8.7.374-2.el5

ruby-libs-1.8.7.374-2.el5

rubygems-1.3.7-1.el5

7.gem 安装 redis ruby 接口

gem install redis

8：执行 redis 的创建集群命令创建集群

#redis-trib.rb 的 create 子命令构建

#--replicas 则指定了为 Redis Cluster 中的每个 Master 节点配备几个 Slave 节点

#节点角色由顺序决定,先 master 之后是 slave

创建方式：

cd /usr/local/redis3.0/src

./redis-trib.rb create --replicas 1 127.0.0.1:7000 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

错误笔记备注：

8.1 执行上面的命令的时候会报错，因为是执行的 ruby 的脚本，需要 ruby 的环境

错误内容：/usr/bin/env: ruby: No such file or directory

所以需要安装 ruby 的环境，这里推荐使用 yum install ruby 安装

yum install ruby

8.2 然后再执行第 6 步的创建集群命令，还会报错，提示缺少 rubygems 组件，使用 yum 安装

错误内容：

./redis-trib.rb:24:in `require': no such file to load -- rubygems (LoadError)

from ./redis-trib.rb:24

yum install rubygems

8.3 再次执行第 8 步的命令，还会报错，提示不能加载 redis，是因为缺少 redis 和 ruby 的接口，使用 gem 安装

错误内容：

/usr/lib/ruby/site_ruby/1.8/rubygems/custom_require.rb:31:in

`gem_original_require': no such file to load -- redis (LoadError)

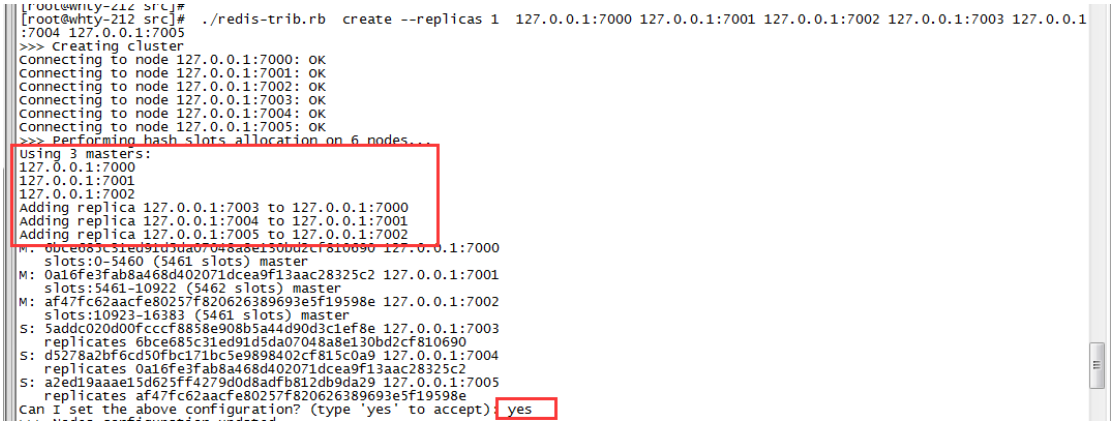
from /usr/lib/ruby/site_ruby/1.8/rubygems/custom_require.rb:31:in `require'

from ./redis-trib.rb:25

gem install redis

8.4 再次执行第 8 步的命令，正常执行

输入 yes，然后配置完成。



注意观察 主从的配置：

默认是前三个节点 7000 7001 7002 是主，

后 3 个节点 7003 7004 7005 是从

如果是部署在不同的服务器，请根据主从分部规则，分开在不同的服务器

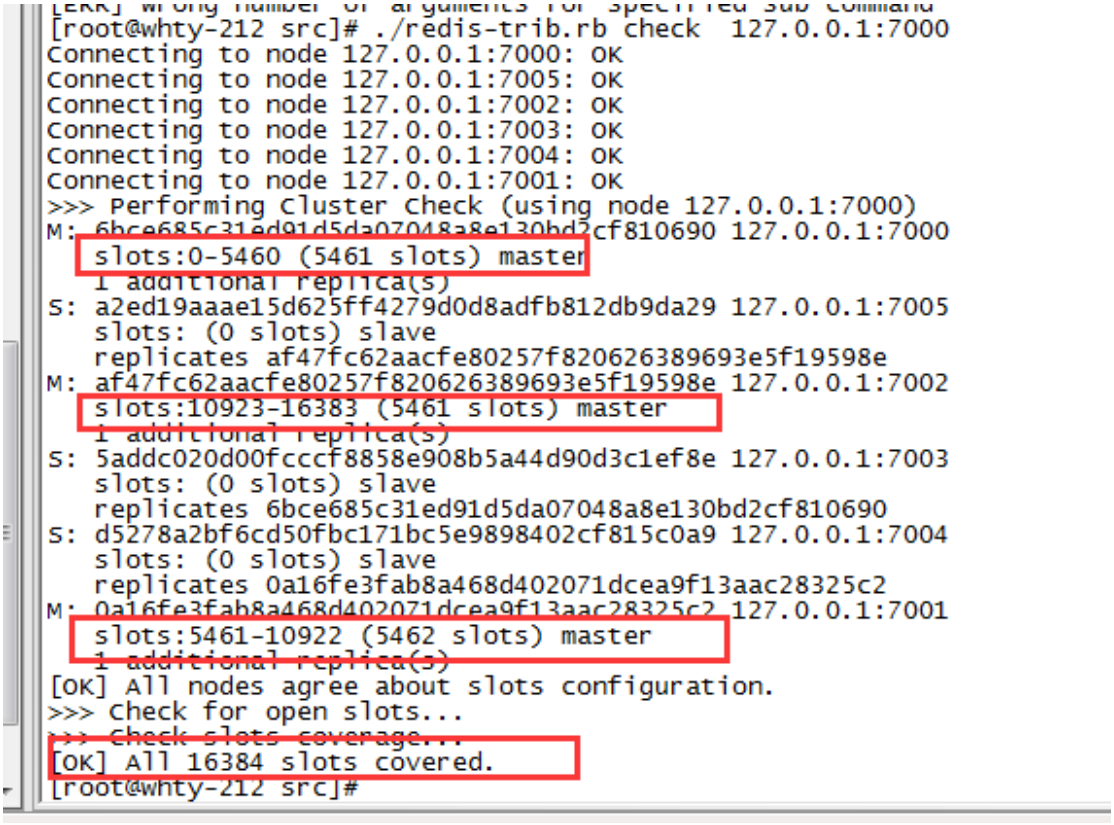
至此 redis 集群即搭建成功！

9：使用 redis-cli 命令进入集群环境

```
redis-cli  -c  -p  7000
```

三. 测试篇

1). 检查集群状态,



#redis-trib.rb 的 check 子命令构建

#ip:port 可以是集群的任意节点

./redis-trib.rb check 1 127.0.0.1:7000

最后输出如下信息,没有任何警告或错误，表示集群启动成功并处于 ok 状态

2):添加新 master 节点

(1)添加一个 master 节点:创建一个空节点（empty node），然后将某些 slot 移动到这个空节点上,这个过程目前需要人工干预

a):根据端口生成配置文件(ps:establish_config.sh 是我自己写的输出配置脚本)

sh establish_config.sh 6386 > conf/redis-6386.conf

b):启动节点

nohup redis-server /opt/redis/conf/redis-6386.conf > /opt/redis/logs/redis-6386.log 2>&1 &

c):加入空节点到集群

add-node 将一个节点添加到集群里面， 第一个是新节点 ip:port, 第二个是任意一个已存在节点 ip:port

redis-trib.rb add-node 10.10.34.14:6386 10.10.34.14:6381

node:新节点没有包含任何数据， 因为它没有包含任何 slot。新加入的加点是一个主节点， 当集群需要将某个从节点升级为新的主节点时， 这个新节点不会被选中

d):为新节点分配 slot

redis-trib.rb reshard 10.10.34.14:6386

#根据提示选择要迁移的 slot 数量(ps:这里选择 500)

How many slots do you want to move (from 1 to 16384)? 500

#选择要接受这些 slot 的 node-id

What is the receiving node ID? f51e26b5d5ff74f85341f06f28f125b7254e61bf

#选择 slot 来源:

#all 表示从所有的 master 重新分配，

#或者数据要提取 slot 的 master 节点 id,最后用 done 结束

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:all

#打印被移动的 slot 后，输入 yes 开始移动 slot 以及对应的数据.

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

#结束

3):添加新的 slave 节点

a):前三步操作同添加 master 一样

b)第四步:redis-cli 连接上新节点 shell,输入命令:cluster replicate 对应 master 的 node-id

cluster replicate 2b9ebcbd627ff0fd7a7bbcc5332fb09e72788835

note:在线添加 slave 时，需要 dump 整个 master 进程，并传递到 slave，再由 slave 加载 rdb 文件到内存，rdb 传输过程中 Master 可能无法提供服务,整个过程消耗大量 io,小心操作.

例如本次添加 slave 操作产生的 rdb 文件

-rw-r--r-- 1 root root 34946 Apr 17 18:23 dump-6386.rdb

-rw-r--r-- 1 root root 34946 Apr 17 18:23 dump-7386.rdb

4):在线 reshard 数据:

对于负载/数据均匀的情况，可以在线 reshard slot 来解决,方法与添加新 master 的 reshard 一样，只是需要 reshard 的 master 节点是老节点.

5):删除一个 slave 节点

#redis-trib del-node ip:port '<node-id>'

redis-trib.rb del-node 10.10.34.14:7386 'c7ee2fca17cb79fe3c9822ced1d4f6c5e169e378'

6):删除一个 master 节点

a):删除 master 节点之前首先要使用 reshard 移除 master 的全部 slot,然后再删除当前节点(目前只能把被删除
master 的 slot 迁移到一个节点上)

#把 10.10.34.14:6386 当前 master 迁移到 10.10.34.14:6380 上

redis-trib.rb reshard 10.10.34.14:6380

#根据提示选择要迁移的 slot 数量(ps:这里选择 500)

How many slots do you want to move (from 1 to 16384)? 500(被删除 master 的所有 slot 数量)

#选择要接受这些 slot 的 node-id(10.10.34.14:6380)

What is the receiving node ID? c4a31c852f81686f6ed8bcd6d1b13accdc947fd2 (ps:10.10.34.14:6380 的 node-id)

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:f51e26b5d5ff74f85341f06f28f125b7254e61bf(被删除 master 的 node-id)

Source node #2:done

#打印被移动的 slot 后，输入 yes 开始移动 slot 以及对应的数据.

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

b):删除空 master 节点

redis-trib.rb del-node 10.10.34.14:6386 'f51e26b5d5ff74f85341f06f28f125b7254e61bf'

四:redis cluster 客户端(Jedis)

1:客户端基本操作使用

```
<span style="color: #333333; font-family: Arial, sans-serif;"><span style="color: #333333; font-family: Arial, sans-serif;"> private static BinaryJedisCluster jc;
   static {
         //只给集群里一个实例就可以
          Set<HostAndPort> jedisClusterNodes = new HashSet<HostAndPort>();
          jedisClusterNodes.add(new HostAndPort("10.10.34.14", 6380));
          jedisClusterNodes.add(new HostAndPort("10.10.34.14", 6381));
          jedisClusterNodes.add(new HostAndPort("10.10.34.14", 6382));
          jedisClusterNodes.add(new HostAndPort("10.10.34.14", 6383));
          jedisClusterNodes.add(new HostAndPort("10.10.34.14", 6384));
          jedisClusterNodes.add(new HostAndPort("10.10.34.14", 7380));
          jedisClusterNodes.add(new HostAndPort("10.10.34.14", 7381));
          jedisClusterNodes.add(new HostAndPort("10.10.34.14", 7382));
          jedisClusterNodes.add(new HostAndPort("10.10.34.14", 7383));
          jedisClusterNodes.add(new HostAndPort("10.10.34.14", 7384));
          jc = new BinaryJedisCluster(jedisClusterNodes);
      }
@Test
      public void testBenchRedisSet() throws Exception {
          final Stopwatch stopwatch = new Stopwatch();
          List list = buildBlogVideos();
          for (int i = 0; i < 1000; i++) {
```

```
            String key = "key:" + i;
            stopwatch.start();
            byte[] bytes1 = protostuffSerializer.serialize(list);
            jc.setex(key, 60 * 60, bytes1);
            stopwatch.stop();
        }
        System.out.println("time=" + stopwatch.toString());
    }</span></span>
```

2:jedis 客户端的坑.

1)cluster 环境下 redis 的 slave 不接受任何读写操作，

2)client 端不支持 keys 批量操作,不支持 select dbNum 操作，只有一个 db:select 0

3)JedisCluster 的 info()等单机函数无法调用,返回(No way to dispatch this command to Redis Cluster)错误，.

4)JedisCluster 没有针对 byte[]的 API，需要自己扩展(附件是我加的基于 byte[]的 BinaryJedisCluster api)