高可用架构篇

Redis 集群的安装（Redis3+CentOS）

参考文档

Redis 官方集群指南：http://redis.io/topics/cluster-tutorial

Redis 官方集群规范：http://redis.io/topics/cluster-spec

Redis 集群指南(中文翻译，紧供参考)：http://redisdoc.com/topic/cluster-tutorial.html

Redis 集群规范(中文翻译，紧供参考)： http://redisdoc.com/topic/cluster-spec.html

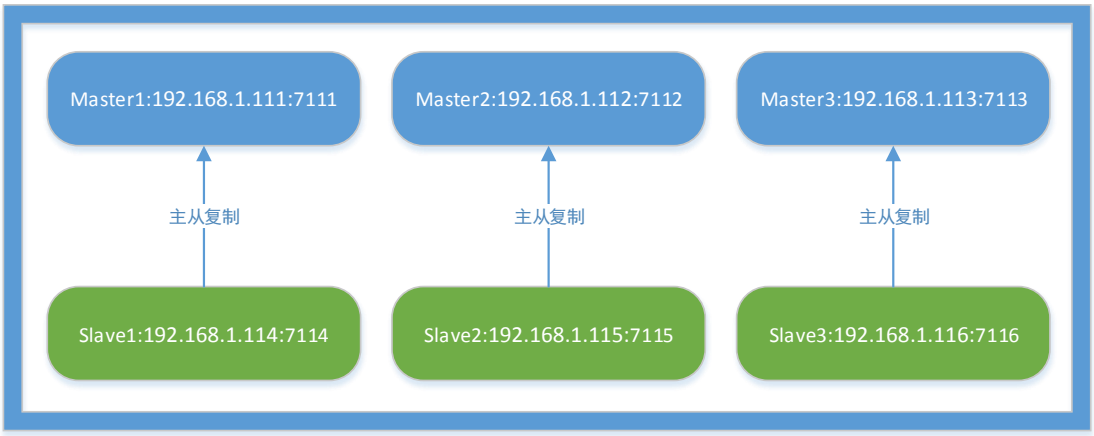
（建议学员们在观看视频前，请先对以上参考文档中的内容有个大概的了解）

Redis 集群介绍、特性、规范等（可看提供的参考文档+视频解说）

Redis 集群的安装（Redis3.0.3 + CentOS6.6_x64）

要让 Redis3.0 集群正常工作至少需要 3 个 Master 节点，要想实现高可用，每个 Master 节点要配备至少 1 个 Slave 节点。根据以上特点和要求，进行如下的集群实施规划：

使用 6 台服务器（物理机或虚拟机）部署 3 个 Master + 3 个 Slave；



Redis3.0集群规划，吴水成，wu-sc@foxmail.com

主机名	IP	服务端口 默认 6379	集群端口 服务端口数+10000	主/从
edu-redis-01	192.168.1.111	7111	17111	Master
edu-redis-02	192.168.1.112	7112	17112	Master
edu-redis-03	192.168.1.113	7113	17113	Master
edu-redis-04	192.168.1.114	7114	17114	Slave
edu-redis-05	192.168.1.115	7115	17115	Slave
edu-redis-06	192.168.1.116	7116	17116	Slave

按规划：防火墙中打开相应的端口

192.168.1.111

-A INPUT -m state --state NEW -m tcp -p tcp --dport 7111 -j ACCEPT

-A INPUT -m state --state NEW -m tcp -p tcp --dport 17111 -j ACCEPT

192.168.1.112

-A INPUT -m state --state NEW -m tcp -p tcp --dport 7112 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 17112 -j ACCEPT
192.168.1.113
-A INPUT -m state --state NEW -m tcp -p tcp --dport 7113 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 17113 -j ACCEPT
192.168.1.114
-A INPUT -m state --state NEW -m tcp -p tcp --dport 7114 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 17114 -j ACCEPT
192.168.1.115
-A INPUT -m state --state NEW -m tcp -p tcp --dport 7115 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 17115 -j ACCEPT
192.168.1.116
-A INPUT -m state --state NEW -m tcp -p tcp --dport 7116 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 17116 -j ACCEPT

安装目录：/usr/local/redis3
用户：root

编译和安装所需的包:
```
# yum install gcc tcl
```

下载（或上传）Redis3 最新稳定版（当前最新版 redis-3.0.3.tar.gz）
```
# cd /usr/local/src
# wget http://download.redis.io/releases/redis-3.0.3.tar.gz
```

创建安装目录:
```
# mkdir /usr/local/redis3
```

解压:
```
# tar -zxvf redis-3.0.3.tar.gz
# cd redis-3.0.3
```

安装(使用 PREFIX 指定安装目录)：
```
# make PREFIX=/usr/local/redis3 install
```
安装完成后，可以看到/usr/local/redis3 目录下有一个 bin 目录，bin 目录里就是 redis 的命令脚本：
redis-benchmark redis-check-aof redis-check-dump redis-cli redis-server

创建集群配置目录，并拷贝 redid.conf 配置文件到各节点配置目录：
192.168.1.111
```
# mkdir -p /usr/local/redis3/cluster/7111
# cp /usr/local/src/redis-3.0.3/redis.conf /usr/local/redis3/cluster/7111/redis-7111.conf
```
192.168.1.112
```
# mkdir -p /usr/local/redis3/cluster/7112
# cp /usr/local/src/redis-3.0.3/redis.conf /usr/local/redis3/cluster/7112/redis-7112.conf
```

192.168.1.113

```
# mkdir -p /usr/local/redis3/cluster/7113
# cp /usr/local/src/redis-3.0.3/redis.conf /usr/local/redis3/cluster/7113/redis-7113.conf
```

192.168.1.114

```
# mkdir -p /usr/local/redis3/cluster/7114
# cp /usr/local/src/redis-3.0.3/redis.conf /usr/local/redis3/cluster/7114/redis-7114.conf
```

192.168.1.115

```
# mkdir -p /usr/local/redis3/cluster/7115
# cp /usr/local/src/redis-3.0.3/redis.conf /usr/local/redis3/cluster/7115/redis-7115.conf
```

192.168.1.116

```
# mkdir -p /usr/local/redis3/cluster/7116
# cp /usr/local/src/redis-3.0.3/redis.conf /usr/local/redis3/cluster/7116/redis-7116.conf
```

修改配置文件中的下面选项：

6 个节点的 redis.conf 配置文件内容，注意修改下红色字体部分的内容即可，其他都相同：

配置选项	选项值	说明
daemonize	yes	是否作为守护进程运行
pidfile	/var/run/redis-7111.pid	如以后台进程运行，则需指定一个 pid，默认为/var/run/redis.pid
port	7111	监听端口，默认为 6379 注意：集群通讯端口值默认为此端口值+10000, 如17111
databases	1	可用数据库数，默认值为 16，默认数据库存储在 DB 0 号 ID 库中，无特殊需求，建议仅设置一个数据库 databases 1
cluster-enabled	yes	打开 redis 集群
cluster-config-file	/usr/local/redis3/cluster/7111/nodes.conf	集群配置文件(启动自动生成)，不用人为干涉
cluster-node-timeout	15000	节点互连超时时间。毫秒
cluster-migration-barrier	1	数据迁移的副本临界数，这个参数表示的是，一个主节点在拥有多少个好的从节点的时候就要割让一个从节点出来给另一个没有任何从节点的主节点。
cluster-require-full-coverage	yes	如果某一些 key space 没有被集群中任何节点覆盖，集群将停止接受写入。
appendonly	yes	启用 aof 持久化方式 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
dir	/usr/local/redis3/cluster/7111	节点数据持久化存放目录（建议配置）

包含了最少选项的集群配置文件示例如下：

```
port 7000
cluster-enabled yes
cluster-config-file nodes.conf
cluster-node-timeout 5000
```

```
appendonly yes
```

使用如下命令启动这 6 个 Redis 节点实例：

192.168.1.111

```
# /usr/local/redis3/bin/redis-server /usr/local/redis3/cluster/7111/redis-7111.conf
```

192.168.1.112

```
# /usr/local/redis3/bin/redis-server /usr/local/redis3/cluster/7112/redis-7112.conf
```

192.168.1.113

```
# /usr/local/redis3/bin/redis-server /usr/local/redis3/cluster/7113/redis-7113.conf
```

192.168.1.114

```
# /usr/local/redis3/bin/redis-server /usr/local/redis3/cluster/7114/redis-7114.conf
```

192.168.1.115

```
# /usr/local/redis3/bin/redis-server /usr/local/redis3/cluster/7115/redis-7115.conf
```

192.168.1.116

```
# /usr/local/redis3/bin/redis-server /usr/local/redis3/cluster/7116/redis-7116.conf
```

启动之后用 PS 命令查看实例启动情况：

```
[root@edu-redis-01 cluster]# ps -ef | grep redis
root  5443  1  0 22:49 ?  00:00:00 /usr/local/redis3/bin/redis-server *:7111 [cluster]
[root@edu-redis-02 cluster]# ps -ef | grep redis
root  5421  1  0 22:49 ?  00:00:00 /usr/local/redis3/bin/redis-server *:7112 [cluster]
[root@edu-redis-03 cluster]# ps -ef | grep redis
root  5457  1  0 22:49 ?  00:00:00 /usr/local/redis3/bin/redis-server *:7113 [cluster]
[root@edu-redis-04 cluster]# ps -ef | grep redis
root  5379  1  0 22:50 ?  00:00:00 /usr/local/redis3/bin/redis-server *:7114 [cluster]
[root@edu-redis-05 cluster]# ps -ef | grep redis
root  5331  1  0 22:50 ?  00:00:00 /usr/local/redis3/bin/redis-server *:7115 [cluster]
[root@edu-redis-06 cluster]# ps -ef | grep redis
root  5687  1  0 22:50 ?  00:00:00 /usr/local/redis3/bin/redis-server *:7116 [cluster]
```

注意：启动完毕后，6 个 Redis 实例尚未构成集群。

接下来准备创建集群

安装 ruby 和 rubygems（注意：需要 ruby 的版本在 1.8.7 以上）

```
# yum install ruby rubygems
```

检查 ruby 版本：

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

gem 安装 redis ruby 接口：

```
# gem install redis
Successfully installed redis-3.2.1
```

```
1 gem installed
Installing ri documentation for redis-3.2.1...
Installing RDoc documentation for redis-3.2.1...
```

执行 Redis 集群创建命令（只需要在其中一个节点上执行一次则可）

```
# cd /usr/local/src/redis-3.0.3/src/
# cp redis-trib.rb /usr/local/bin/redis-trib
# redis-trib create --replicas 1 192.168.1.114:7114 192.168.1.115:7115 192.168.1.116:7116
192.168.1.111:7111 192.168.1.112:7112 192.168.1.113:7113

>>> Creating cluster
Connecting to node 192.168.1.114:7114: OK
Connecting to node 192.168.1.115:7115: OK
Connecting to node 192.168.1.116:7116: OK
Connecting to node 192.168.1.111:7111: OK
Connecting to node 192.168.1.112:7112: OK
Connecting to node 192.168.1.113:7113: OK
>>> Performing hash slots allocation on 6 nodes...
Using 3 masters:
192.168.1.113:7113
192.168.1.112:7112
192.168.1.111:7111
Adding replica 192.168.1.116:7116 to 192.168.1.113:7113
Adding replica 192.168.1.115:7115 to 192.168.1.112:7112
Adding replica 192.168.1.114:7114 to 192.168.1.111:7111
S: 007a3fe8d7451d3d0a78fffd2653c8641809499c 192.168.1.114:7114
   replicates 94e140b9ca0735040ae3428983835f1d93327aeb
S: ea69b6b6e2e7723eed50b1dabea9d244ccf3f098 192.168.1.115:7115
   replicates c642b3071c4b2b073707ed3c3a2c16d53a549eff
S: 5f09dc0671732cf06a09f28631c90e0c68408520 192.168.1.116:7116
   replicates 896a3c99da4fcf680de1f42406fccb551d8c40c3
M: 94e140b9ca0735040ae3428983835f1d93327aeb 192.168.1.111:7111
   slots:10923-16383 (5461 slots) master
M: c642b3071c4b2b073707ed3c3a2c16d53a549eff 192.168.1.112:7112
   slots:5461-10922 (5462 slots) master
M: 896a3c99da4fcf680de1f42406fccb551d8c40c3 192.168.1.113:7113
   slots:0-5460 (5461 slots) master
Can I set the above configuration? (type 'yes' to accept): yes
```

（输入 yes 并按下回车确认之后，集群就会将配置应用到各个节点，并连接起（join）各个节点，也就是让各个节点开始互相通讯）

```
>>> 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 192.168.1.114:7114)
M: 007a3fe8d7451d3d0a78fffd2653c8641809499c 192.168.1.114:7114
   slots: (0 slots) master
   replicates 94e140b9ca0735040ae3428983835f1d93327aeb
M: ea69b6b6e2e7723eed50b1dabea9d244ccf3f098 192.168.1.115:7115
   slots: (0 slots) master
   replicates c642b3071c4b2b073707ed3c3a2c16d53a549eff
M: 5f09dc0671732cf06a09f28631c90e0c68408520 192.168.1.116:7116
   slots: (0 slots) master
   replicates 896a3c99da4fcf680de1f42406fccb551d8c40c3
M: 94e140b9ca0735040ae3428983835f1d93327aeb 192.168.1.111:7111
   slots:10923-16383 (5461 slots) master
M: c642b3071c4b2b073707ed3c3a2c16d53a549eff 192.168.1.112:7112
   slots:5461-10922 (5462 slots) master
M: 896a3c99da4fcf680de1f42406fccb551d8c40c3 192.168.1.113:7113
   slots:0-5460 (5461 slots) master
```

一切正常的情况下输出以下信息：

```
[OK] All nodes agree about slots configuration.
>>> Check for open slots...
>>> Check slots coverage...
[OK] All 16384 slots covered.
```

最后一行信息表示集群中的 16384 个槽都有至少一个主节点在处理， 集群运作正常。

集群创建过程说明：

(1) 给定 redis-trib 程序的命令是 create ， 这表示我们希望创建一个新的集群；

(2) 这里的 --replicas 1 表示每个主节点下有一个从节点；

(3) 之后跟着的其它参数则是实例的地址列表，程序使用这些地址所指示的实例来创建新集群；

总的来说，以上命令的意思就是让 redis-trib 程序创建一个包含三个主节点和三个从节点的集群。

接着，redis-trib 会打印出一份预想中的配置给你看，如果你觉得没问题的话（注意核对主从关系是否是你想要的），就可以输入 yes ， redis-trib 就会将这份配置应用到集群当中。

集群简单测试

使用 redis-cli 命令进入集群环境

```
[root@edu-redis-04 bin]# ./redis-cli -c -p 7114
127.0.0.1:7114> set wusc WuShuicheng
-> Redirected to slot [8559] located at 192.168.1.112:7112
OK

[root@edu-redis-01 bin]# ./redis-cli -c -p 7111
127.0.0.1:7111> get wusc
-> Redirected to slot [8559] located at 192.168.1.112:7112
```

"WuShuicheng"

[root@edu-redis-02 bin]# ./redis-cli -c -p 7112
127.0.0.1:7112> get wusc
"WuShuicheng"
127.0.0.1:7112>

[root@edu-redis-01 bin]# ./redis-cli -p 7111 cluster nodes

```
[root@edu-redis-01 bin]# ./redis-cli -p 7111 cluster nodes
5f09dc0671732cf06a09f28631c90e0c68408520 192.168.1.116:7116 slave 896a3c99da4fcf680de1f42406fccb551d8c40c3 0 143819
1571045 6 connected
896a3c99da4fcf680de1f42406fccb551d8c40c3 192.168.1.113:7113 master - 0 1438191565033 6 connected 0-5460
007a3fe8d7451d3d0a78fffd2653c8641809499c 192.168.1.114:7114 slave,fail 94e140b9ca0735040ae3428983835f1d93327aeb 143
8188320834 1438188315822 4 connected
ea69b6b6e2e7723eed50b1dabea9d244ccf3f098 192.168.1.115:7115 slave c642b3071c4b2b073707ed3c3a2c16d53a549eff 0 143819
1572046 5 connected
c642b3071c4b2b073707ed3c3a2c16d53a549eff 192.168.1.112:7112 master - 0 1438191570042 5 connected 5461-10922
94e140b9ca0735040ae3428983835f1d93327aeb 192.168.1.111:7111 myself,master - 0 0 4 connected 10923-16383
[root@edu-redis-01 bin]#
```

将 Redis 配置成服务

（非伪集群适用，也就是每个节点都单独物理机部署的情况下）：

按上面的操作步骤，Redis 的启动脚本为：/usr/local/src/redis-3.0.3/utils/redis_init_script

将启动脚本复制到/etc/rc.d/init.d/目录下，并命名为 redis：

cp /usr/local/src/redis-3.0.3/utils/redis_init_script /etc/rc.d/init.d/redis

编辑/etc/rc.d/init.d/redis，修改相应配置，使之能注册成为服务：

vi /etc/rc.d/init.d/redis

```
#!/bin/sh
#
# Simple Redis init.d script conceived to work on Linux systems
# as it does use of the /proc filesystem.

REDISPORT=6379
EXEC=/usr/local/bin/redis-server
CLIEXEC=/usr/local/bin/redis-cli

PIDFILE=/var/run/redis_${REDISPORT}.pid
CONF="/etc/redis/${REDISPORT}.conf"

case "$1" in
    start)
        if [ -f $PIDFILE ]
        then
                echo "$PIDFILE exists, process is already running or crashed"
        else
                echo "Starting Redis server..."
                $EXEC $CONF
        fi
        ;;
```

```
    stop)
        if [ ! -f $PIDFILE ]
        then
                echo "$PIDFILE does not exist, process is not running"
        else
                PID=$(cat $PIDFILE)
                echo "Stopping ..."
                $CLIEXEC -p $REDISPORT shutdown
                while [ -x /proc/${PID} ]
                do
                    echo "Waiting for Redis to shutdown ..."
                    sleep 1
                done
                echo "Redis stopped"
        fi
        ;;
    *)
        echo "Please use start or stop as first argument"
        ;;
esac
```

查看以上 redis 服务脚本，关注标为橙色的几个属性，做如下几个修改的准备：

(1) 在脚本的第一行后面添加一行内容如下：

#chkconfig: 2345 80 90

（如果不添加上面的内容，在注册服务时会提示：service redis does not support chkconfig）

(2) REDISPORT 端口修改各节点对应的端口；（注意，端口名将与下面的配置文件名有关）

(3) EXEC=/usr/local/bin/redis-server 改为 EXEC=/usr/local/redis3/bin/redis-server

(4) CLIEXEC=/usr/local/bin/redis-cli 改为 CLIEXEC=/usr/local/redis3/bin/redis-cli

(5) 配置文件设置，对 CONF 属性作如下调整：

CONF="/etc/redis/${REDISPORT}.conf"

改为 CONF="/usr/local/redis3/cluster/${REDISPORT}/redis-${REDISPORT}.conf"

(6) 更改 redis 开启的命令，以后台运行的方式执行：

$EXEC $CONF &　　#“&”作用是将服务转到后面运行

修改后的/etc/rc.d/init.d/redis 服务脚本内容为（注意各节点的端口不同）：

```
#!/bin/sh
#chkconfig: 2345 80 90
#
# Simple Redis init.d script conceived to work on Linux systems
# as it does use of the /proc filesystem.

REDISPORT=7111
EXEC=/usr/local/redis3/bin/redis-server
CLIEXEC=/usr/local/redis3/bin/redis-cli
```

```
PIDFILE=/var/run/redis-${REDISPORT}.pid
CONF="/usr/local/redis3/cluster/${REDISPORT}/redis-${REDISPORT}.conf "

case "$1" in
    start)
        if [ -f $PIDFILE ]
        then
                echo "$PIDFILE exists, process is already running or crashed"
        else
                echo "Starting Redis server..."
                $EXEC $CONF &
        fi
        ;;
    stop)
        if [ ! -f $PIDFILE ]
        then
                echo "$PIDFILE does not exist, process is not running"
        else
                PID=$(cat $PIDFILE)
                echo "Stopping ..."
                $CLIEXEC -p $REDISPORT shutdown
                while [ -x /proc/${PID} ]
                do
                    echo "Waiting for Redis to shutdown ..."
                    sleep 1
                done
                echo "Redis stopped"
        fi
        ;;
    *)
        echo "Please use start or stop as first argument"
        ;;
esac
```

以上配置操作完成后，便可将 Redis 注册成为服务：

```
# chkconfig --add redis
```

防火墙中打开对应的端口，各节点的端口不同（前面已操作则可跳过此步）

```
# vi /etc/sysconfig/iptables
```

添加：

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 7111 -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 17111 -j ACCEPT
```

重启防火墙：

```
# service iptables restart
```

启动 Redis 服务

```
# service redis start
```

将 Redis 添加到环境变量中：

```
# vi /etc/profile
```

在最后添加以下内容：

```
## Redis env
export PATH=$PATH:/usr/local/redis3/bin
```

使配置生效：

```
# source /etc/profile
```

现在就可以直接使用 redis-cli 等 redis 命令了：

```
[root@edu-cache-01 bin]# redis-cli
127.0.0.1:6379> set wusc WuShuicheng
OK
127.0.0.1:6379> get wusc
"WuShuicheng"
127.0.0.1:6379>
```

关闭 Redis 服务

```
# service redis stop
```

默认情况下，Redis 未开启安全认证，可以通过/usr/local/redis3/cluster/7111/redis-7111.conf 的 requirepass 指定一个验证密码。

其它供参考资料

Redis 3.0 集群搭建测试（一）：http://blog.csdn.net/zhu_tianwei/article/details/44928779

Redis 3.0 集群搭建测试（二）：http://blog.csdn.net/zhu_tianwei/article/details/45009647

Redis 集群要点：http://5i.io/redis-3-0-cluster-configuration/