Redis集群

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

Redis 集群中内置了 16384 个哈希槽，当需要在 Redis 集群中放置一个 key-value 时，redis 先对 key 使用 crc16 算法算出一个结果，然后把结果对 16384 求余数，这样每个 key 都会对应一个编号在 0-16383 之间的哈希槽，redis 会根据节点数量大致均等的将哈希槽映射到不同的节点

示例如下：

Server3

10001-16383

Server2

5001-10000

Server1

0-5000

## redis-cluster投票:容错



(1)集群中所有master参与投票,如果半数以上master节点与其中一个master节点通信超过(cluster-node-timeout),认为该master节点挂掉.

(2):什么时候整个集群不可用(cluster\_state:fail)?

* 如果**集群任意master挂掉**,**且当前master没有slave，则集群进入fail状态**。也可以理解成集群的[0-16383]slot映射不完全时进入fail状态。
* **如果集群超过半数以上master挂掉，无论是否有slave，集群进入fail状态。**

## 安装ruby

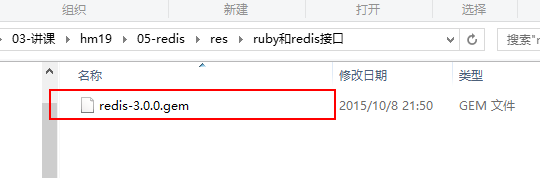
**集群管理工具（redis-trib.rb）是使用ruby脚本语言编写的。**

第一步：**安装ruby**

[root@itheima bin2]# yum install ruby

[root@itheima bin2]# yum install rubygems

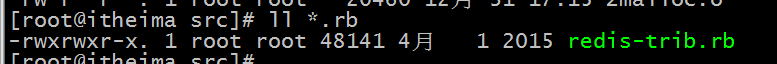
第二步：将以下文件上传到**linux系统**



第三步：**安装ruby和redis接口**

[root@itheima ~]# gem install redis-3.0.0.gem

第四步：将redis-3.0.0包下src目录中的以下文件拷贝到redis19/redis-cluster/



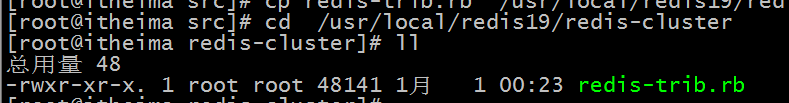
[root@itheima src]# cd /usr/local/redis19/

[root@itheima redis19]# mkdir redis-cluster

[root@itheima redis19]# cd /root/redis-3.0.0/src/

[root@itheima src]# cp redis-trib.rb /usr/local/redis19/redis-cluster

第五步：查看是否拷贝成功



## 搭建集群

**搭建集群最少也得需要3台主机，如果每台主机再配置一台从机的话，则最少需要6台机器。**

端口设计如下：**7001-7006**

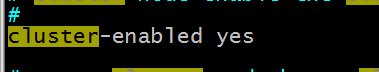
第一步：复制出一个7001机器

[root@itheima redis19]# cp bin ./redis-cluster/7001 –r

第二步：如果存在持久化文件，则删除

[root@itheima 7001]# rm -rf appendonly.aof dump.rdb

第三步：设置集群参数



第四步：修改端口



第五步：复制出7002-7006机器

[root@itheima redis-cluster]# cp 7001/ 7002 -r

[root@itheima redis-cluster]# cp 7001/ 7003 -r

[root@itheima redis-cluster]# cp 7001/ 7004 -r

[root@itheima redis-cluster]# cp 7001/ 7005 -r

[root@itheima redis-cluster]# cp 7001/ 7006 –r

第六步：修改7002-7006机器的端口

第七步：启动7001-7006这六台机器



第八步：修改start-all.sh文件的权限

[root@itheima redis-cluster]# chmod u+x start-all.sh

[root@itheima redis-cluster]# ./start-all.sh

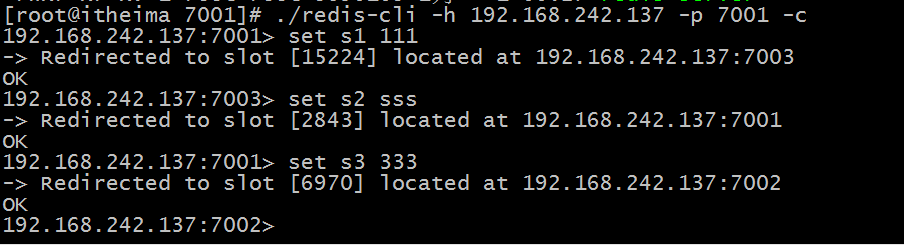
第九步：创建集群

|  |
| --- |
| [root@itheima redis-cluster]# **./redis-trib.rb create --replicas 1 192.168.242.137:7001 192.168.242.137:7002 192.168.242.137:7003 192.168.242.137:7004 192.168.242.137:7005 192.168.242.137:7006**  >>> Creating cluster  Connecting to node 192.168.242.137:7001: OK  Connecting to node 192.168.242.137:7002: OK  Connecting to node 192.168.242.137:7003: OK  Connecting to node 192.168.242.137:7004: OK  Connecting to node 192.168.242.137:7005: OK  Connecting to node 192.168.242.137:7006: OK  >>> Performing hash slots allocation on 6 nodes...  Using 3 masters:  192.168.242.137:7001  192.168.242.137:7002  192.168.242.137:7003  Adding replica 192.168.242.137:7004 to 192.168.242.137:7001  Adding replica 192.168.242.137:7005 to 192.168.242.137:7002  Adding replica 192.168.242.137:7006 to 192.168.242.137:7003  M: 8240cd0fe6d6f842faa42b0174fe7c5ddcf7ae24 192.168.242.137:7001  slots:0-5460 (5461 slots) master  M: 4f52a974f64343fd9f1ee0388490b3c0647a4db7 192.168.242.137:7002  slots:5461-10922 (5462 slots) master  M: cb7c5def8f61df2016b38972396a8d1f349208c2 192.168.242.137:7003  slots:10923-16383 (5461 slots) master  S: 66adf006fed43b3b5e499ce2ff1949a756504a16 192.168.242.137:7004  replicates 8240cd0fe6d6f842faa42b0174fe7c5ddcf7ae24  S: cbb0c9bc4b27dd85511a7ef2d01bec90e692793b 192.168.242.137:7005  replicates 4f52a974f64343fd9f1ee0388490b3c0647a4db7  S: a908736eadd1cd06e86fdff8b2749a6f46b38c00 192.168.242.137:7006  replicates cb7c5def8f61df2016b38972396a8d1f349208c2  Can I set the above configuration? (type 'yes' to accept): **yes**  >>> 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 192.168.242.137:7001)  M: 8240cd0fe6d6f842faa42b0174fe7c5ddcf7ae24 192.168.242.137:7001  slots:0-5460 (5461 slots) master  M: 4f52a974f64343fd9f1ee0388490b3c0647a4db7 192.168.242.137:7002  slots:5461-10922 (5462 slots) master  M: cb7c5def8f61df2016b38972396a8d1f349208c2 192.168.242.137:7003  slots:10923-16383 (5461 slots) master  M: 66adf006fed43b3b5e499ce2ff1949a756504a16 192.168.242.137:7004  slots: (0 slots) master  replicates 8240cd0fe6d6f842faa42b0174fe7c5ddcf7ae24  M: cbb0c9bc4b27dd85511a7ef2d01bec90e692793b 192.168.242.137:7005  slots: (0 slots) master  replicates 4f52a974f64343fd9f1ee0388490b3c0647a4db7  M: a908736eadd1cd06e86fdff8b2749a6f46b38c00 192.168.242.137:7006  slots: (0 slots) master  replicates cb7c5def8f61df2016b38972396a8d1f349208c2  [OK] All nodes agree about slots configuration.  >>> Check for open slots...  >>> Check slots coverage...  [OK] All 16384 slots covered.  [root@itheima redis-cluster]# |

## 连接集群

[root@itheima 7001]# ./redis-cli -h 192.168.242.137 -p 7001 **–c**

**-c：指定是集群连接**



## 查看集群信息

* 查看集群信息

|  |
| --- |
| * 192.168.242.137:7002> cluster info * cluster\_state:ok * cluster\_slots\_assigned:16384 * cluster\_slots\_ok:16384 * cluster\_slots\_pfail:0 * cluster\_slots\_fail:0 * cluster\_known\_nodes:6 * cluster\_size:3 * cluster\_current\_epoch:6 * cluster\_my\_epoch:2 * cluster\_stats\_messages\_sent:2372 * cluster\_stats\_messages\_received:2372 * 192.168.242.137:7002> |

* 查看集群节点

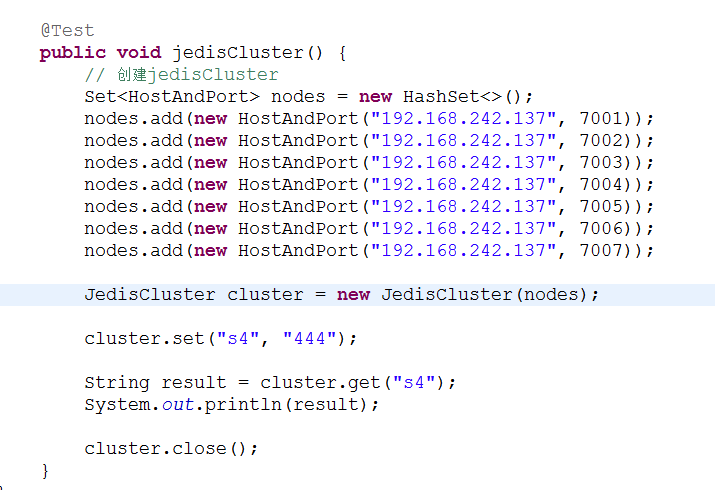
|  |
| --- |
| * 192.168.242.137:7002> cluster nodes * 8240cd0fe6d6f842faa42b0174fe7c5ddcf7ae24 192.168.242.137:7001 master - 0 1451581348093 1 connected 0-5460 * cb7c5def8f61df2016b38972396a8d1f349208c2 192.168.242.137:7003 master - 0 1451581344062 3 connected 10923-16383 * 66adf006fed43b3b5e499ce2ff1949a756504a16 192.168.242.137:7004 slave 8240cd0fe6d6f842faa42b0174fe7c5ddcf7ae24 0 1451581351115 1 connected * a908736eadd1cd06e86fdff8b2749a6f46b38c00 192.168.242.137:7006 slave cb7c5def8f61df2016b38972396a8d1f349208c2 0 1451581349101 3 connected * 4f52a974f64343fd9f1ee0388490b3c0647a4db7 192.168.242.137:7002 myself,master - 0 0 2 connected 5461-10922 * cbb0c9bc4b27dd85511a7ef2d01bec90e692793b 192.168.242.137:7005 slave 4f52a974f64343fd9f1ee0388490b3c0647a4db7 0 1451581350108 5 connected |

# jedis连接集群

## 设置防火墙

|  |
| --- |
| [root@itheima redis-cluster]# vim /etc/sysconfig/iptables  -A INPUT -m state --state NEW -m tcp -p tcp --dport 6379 -j ACCEPT  -A INPUT -m state --state NEW -m tcp -p tcp --dport 6379 -j ACCEPT  -A INPUT -m state --state NEW -m tcp -p tcp --dport 6379 -j ACCEPT  -A INPUT -m state --state NEW -m tcp -p tcp --dport 6379 -j ACCEPT  # Firewall configuration written by system-config-firewall  # Manual customization of this file is not recommended.  \*filter  :INPUT ACCEPT [0:0]  :FORWARD ACCEPT [0:0]  :OUTPUT ACCEPT [0:0]  -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT  -A INPUT -p icmp -j ACCEPT  -A INPUT -i lo -j ACCEPT  -A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT  -A INPUT -m state --state NEW -m tcp -p tcp --dport 3306 -j ACCEPT  -A INPUT -m state --state NEW -m tcp -p tcp --dport 8080 -j ACCEPT  **-A INPUT -m state --state NEW -m tcp -p tcp --dport 6379 -j ACCEPT**  **-A INPUT -m state --state NEW -m tcp -p tcp --dport 7001 -j ACCEPT**  **-A INPUT -m state --state NEW -m tcp -p tcp --dport 7002 -j ACCEPT**  **-A INPUT -m state --state NEW -m tcp -p tcp --dport 7003 -j ACCEPT**  **-A INPUT -m state --state NEW -m tcp -p tcp --dport 7004 -j ACCEPT**  **-A INPUT -m state --state NEW -m tcp -p tcp --dport 7005 -j ACCEPT**  **-A INPUT -m state --state NEW -m tcp -p tcp --dport 7006 -j ACCEPT**  **-A INPUT -m state --state NEW -m tcp -p tcp --dport 7007 -j ACCEPT**  -A INPUT -j REJECT --reject-with icmp-host-prohibited  -A FORWARD -j REJECT --reject-with icmp-host-prohibited  COMMIT  ~  ~  ~  ~  "/etc/sysconfig/iptables" 23L, 1146C 已写入  [root@itheima redis-cluster]# service iptables restart  iptables：清除防火墙规则： [确定]  iptables：将链设置为政策 ACCEPT：filter [确定]  iptables：正在卸载模块： [确定]  iptables：应用防火墙规则： [确定]  [root@itheima redis-cluster]# |

## 代码



## 使用spring

* 配置applicationContext.xml

|  |
| --- |
| <!-- 连接池配置 -->  <bean id=*"jedisPoolConfig"* class=*"redis.clients.jedis.JedisPoolConfig"*>  <!-- 最大连接数 -->  <property name=*"maxTotal"* value=*"30"* />  <!-- 最大空闲连接数 -->  <property name=*"maxIdle"* value=*"10"* />  <!-- 每次释放连接的最大数目 -->  <property name=*"numTestsPerEvictionRun"* value=*"1024"* />  <!-- 释放连接的扫描间隔（毫秒） -->  <property name=*"timeBetweenEvictionRunsMillis"* value=*"30000"* />  <!-- 连接最小空闲时间 -->  <property name=*"minEvictableIdleTimeMillis"* value=*"1800000"* />  <!-- 连接空闲多久后释放, 当空闲时间>该值 且 空闲连接>最大空闲连接数 时直接释放 -->  <property name=*"softMinEvictableIdleTimeMillis"* value=*"10000"* />  <!-- 获取连接时的最大等待毫秒数,小于零:阻塞不确定的时间,默认-1 -->  <property name=*"maxWaitMillis"* value=*"1500"* />  <!-- 在获取连接的时候检查有效性, 默认false -->  <property name=*"testOnBorrow"* value=*"true"* />  <!-- 在空闲时检查有效性, 默认false -->  <property name=*"testWhileIdle"* value=*"true"* />  <!-- 连接耗尽时是否阻塞, false报异常,ture阻塞直到超时, 默认true -->  <property name=*"blockWhenExhausted"* value=*"false"* />  </bean>  <!-- redis集群 -->  <bean id=*"jedisCluster"* class=*"redis.clients.jedis.JedisCluster"*>  <constructor-arg index=*"0"*>  <set>  <bean class=*"redis.clients.jedis.HostAndPort"*>  <constructor-arg index=*"0"* value=*"192.168.101.3"*></constructor-arg>  <constructor-arg index=*"1"* value=*"7001"*></constructor-arg>  </bean>  <bean class=*"redis.clients.jedis.HostAndPort"*>  <constructor-arg index=*"0"* value=*"192.168.101.3"*></constructor-arg>  <constructor-arg index=*"1"* value=*"7002"*></constructor-arg>  </bean>  <bean class=*"redis.clients.jedis.HostAndPort"*>  <constructor-arg index=*"0"* value=*"192.168.101.3"*></constructor-arg>  <constructor-arg index=*"1"* value=*"7003"*></constructor-arg>  </bean>  <bean class=*"redis.clients.jedis.HostAndPort"*>  <constructor-arg index=*"0"* value=*"192.168.101.3"*></constructor-arg>  <constructor-arg index=*"1"* value=*"7004"*></constructor-arg>  </bean>  <bean class=*"redis.clients.jedis.HostAndPort"*>  <constructor-arg index=*"0"* value=*"192.168.101.3"*></constructor-arg>  <constructor-arg index=*"1"* value=*"7005"*></constructor-arg>  </bean>  <bean class=*"redis.clients.jedis.HostAndPort"*>  <constructor-arg index=*"0"* value=*"192.168.101.3"*></constructor-arg>  <constructor-arg index=*"1"* value=*"7006"*></constructor-arg>  </bean>  </set>  </constructor-arg>  <constructor-arg index=*"1"* ref=*"jedisPoolConfig"*></constructor-arg>  </bean> |

* 测试代码

|  |
| --- |
| **private** ApplicationContext applicationContext;  @Before  **public** **void** init() {  applicationContext = **new** ClassPathXmlApplicationContext(  "classpath:applicationContext.xml");  }  // redis集群  @Test  **public** **void** testJedisCluster() {  JedisCluster jedisCluster = (JedisCluster) applicationContext  .getBean("jedisCluster");  jedisCluster.set("name", "zhangsan");  String value = jedisCluster.get("name");  System.*out*.println(value);  } |