1.redis cluster集群是什么?

redis cluster集群是一个由多个主从节点群组成的分布式服务器群，它具有复制、高可用和分片特 性。Redis cluster集群不需要sentinel哨兵也能完成节点移除和故障转移的功能。需要将每个节点 设置成集群模式，这种集群模式没有中心节点，可水平扩展，据官方文档称可以线性扩展到 1000节点。redis cluster集群的性能和高可用性均优于之前版本的哨兵模式，且集群配置非常简单

2.redis cluster集群搭建

/usr/local/bin/redis-cli --cluster help

1.原生搭建

1.配置开启cluster节点

cluster-enabled yes（启动集群模式）

cluster-config-file nodes-8001.conf（这里800x最好和port对应上）

2.meet

cluster meet ip port

3.指派槽

查看crc16 算法算出key的槽位命令 cluster keyslot key

16384/3 0-5461 5462-10922 10923-16383 16384/4 4096

cluster addslots slot（槽位下标）

4.分配主从

cluster replicate node-id

2.使用redis提供的rb脚本

redis cluster集群需要至少要三个master节点，我们这里搭建三个master节点，并且给每个 master再搭建一个slave节点，总共6个redis节点，由于节点数较多，这里采用在一台机器 上创建6个redis实例，并将这6个redis实例配置成集群模式，所以这里搭建的是伪集群模 式，当然真正的分布式集群的配置方法几乎一样，搭建伪集群的步骤如下： 第一步：在/usr/local下创建文件夹redis-cluster，然后在其下面分别创建6个文件夹如下 （1）mkdir -p /usr/local/redis-cluster （2）mkdir 8001、 mkdir 8002、 mkdir 8003、 mkdir 8004、 mkdir 8005、 mkdir 8006 第一步：把之前的redis.conf配置文件copy到8001下，修改如下内容： （1）daemonize yes （2）port 8001（分别对每个机器的端口号进行设置） （3）bind 127.0.0.1（如果只在本机玩则可以指定为127.0.0.1 如果需要外网访问则需要指定本机真实ip） 定可能会出现循环查找集群节点机器的情况） （4）dir /usr/local/redis-cluster/8001/（指定数据文件存放位置，必须要指定不同的目 录位置，不然会丢失数据） （5）cluster-enabled yes（启动集群模式） （6）cluster-config-file nodes-8001.conf（这里800x最好和port对应上） （7）cluster-node-timeout 5000 （8）appendonly yes 第三步：把修改后的配置文件，分别 copy到各个文夹下，注意每个文件要修改第2、4、6 项里的端口号，可以用批量替换：:%s/源字符串/目的字符串/g 第四步：由于 redis集群需要使用 ruby命令，所以我们需要安装 ruby（redis5.0之后省略） （1）yum install ruby （2）yum install rubygems （3）gem install redis --version 3.0.0（安装redis和 ruby的接口） 第五步：分别启动6个redis实例，然后检查是

否启动成功 （1）/usr/local/redis/bin/redis-server /usr/local/redis-cluster/800*/redis.conf （2）ps -ef | grep redis 查看是否启动成功

第六步：在redis3的安装目录下执行 redis-trib.rb命令创建整个redis集群 （1）cd /usr/local/redis3/src （2）./redis-trib.rb create --replicas 1 127.0.0.1:9000 127.0.0.1:9001 127.0.0.1:9002 127.0.0.1:9003 127.0.0.1:9004 127.0.0.1:9005

redis5.0使用/usr/local/bin/redis-cli --cluster create 192.168.0.104:7000 192.168.0.104:7001 192.168.0.104:7002 192.168.0.104:7003 192.168.0.104 :7004 192.168.0.104:7005 --cluster-replicas 1

第七步：验证集群： （1）连接任意一个客户端即可：./redis-cli -c -h -p (-c表示集群模式，指定ip地址和端口 号) 如：/usr/local/redis/bin/redis-cli -c -h 127.0.0.1 -p 800* （2）进行验证： cluster info（查看集群信息）、cluster nodes（查看节点列表） （3）进行数据操作验证 （4）关闭集群则需要逐个进行关闭，使用命令：/usr/local/redis/bin/redis-cli -c -h 127.0.0.1 -p 800* shutdown

3.集群伸缩

1.扩容集群

1.准备新节点

2.加入集群

使用redis-cli 语法：add-node 新节点ip 端口 已存在节点ip 端口

使用原生命令 语法：cluster meet ip port

指定主从

使用redis-cli 语法（加入时指定）：add-node 新节点ip 端口 已存在节点ip 端口 --cluster-slave --cluster-master-id masterID

使用原生命令 语法：cluster replicate node-id

3.迁移槽和数据

1.槽迁移计划

语法：/redis-cli --cluster reshard 已存在节点ip： 端口

/usr/local/bin/redis-cli --cluster reshard 192.168.204.188:7000

2.迁移数据

执行流程：提示要分配多少槽-》接收节点ID-》all/done

3.添加从节点

2.缩容集群

1.下线迁移槽

语法：redis-cli --cluster reshard --cluster-from 要迁出节点ID --cluster-to 接收槽节点ID --cluster-slots 迁出槽数量 已存在节点ip 端口

/usr/local/bin/redis-cli --cluster reshard --cluster-from a2fdd1359d03acacf2a6e558acbc006639445d53 --cluster-to 1794864d5f8af79e88cfc0f699f02b6341c78b5c --cluster-slots 1366 192.168.0.104 7000

2.忘记节点.关闭节点

语法： redis-cli --cluster del-node 已存在节点IP：端口 要删除的节点ID

/usr/local/bin/redis-cli --cluster del-node 192.168.0.104:7000
8de55e2a7419983184cede9daab5d36ee9da1fa3

4.cluster客户端

1.moved重定向：指我们发送命令时，会对发送的key进行crc16算法，得到一个数字，然而我们连接的客户端并不是管理这个数字的范围，所以会返回错误并告诉你此key应该对应的槽位，然后客户端需要捕获此异常，重新发起请求到对应的槽位

2.asx重定向：指在我们送发命令时，对应的客户端正在迁移槽位中，所以此时我们不能确定这个key是还在旧的节点中还是新的节点中

3.smart客户端

1.从集群中选取一个可运行节点，使用cluster slots初始化槽和节点映射。

2.将cluster slots的结果映射到本地，为每个节点创建jedispool

3.准备执行命令

5.故障转移（与哨兵相似）

1.故障发现： 通过ping/pong消息实现故障发现（不依赖sentinel）

2.故障恢复

1.检查资格

1.每个从节点检查与主节点的断开时间

超过cluster-node-timeout * cluster-replica-validity-factor 时间取消资格

2.选择偏移量最大的

替换主节点

1.当前从节点取消复制变为主节点（slaveof no one）

2.撤销以前主节点的槽位，给新的主节点

3.向集群广播消息，表明已经替换了故障节点