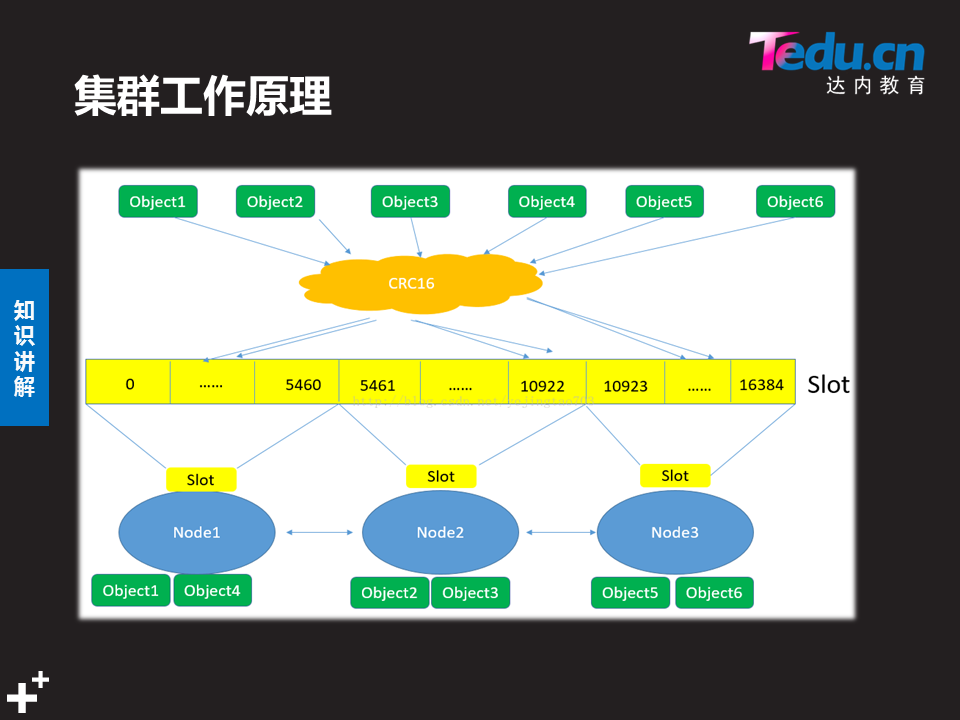
**Morning：**

1. Redis集群原理



**- 数据读写过程**

1. 客户端访问集群(通过任意主库ip、端口)
2. 存入数据时(set)，CRC16计算slot值，跳转到对应主库并存入变量值。

3）获取数据时(get)，CRC16计算slot值，跳转到对应主库返回变量值。

4）从库不允许读、写数据(get)，允许打印(keys \*)，只用于同步主库数据。

**- 工作原理**

hash slot：哈希槽位数，固定为0~16383（总数16384个）

**slot = CRC16(变量名)%16384**

**- slot与变量的关系**

hash slot只是用于标识某变量存储集群中的哪个主库；

不同变量名可能计算出相同slot值；

Redis数据库最大可存储的变量个数，取决于服务器内存大小。

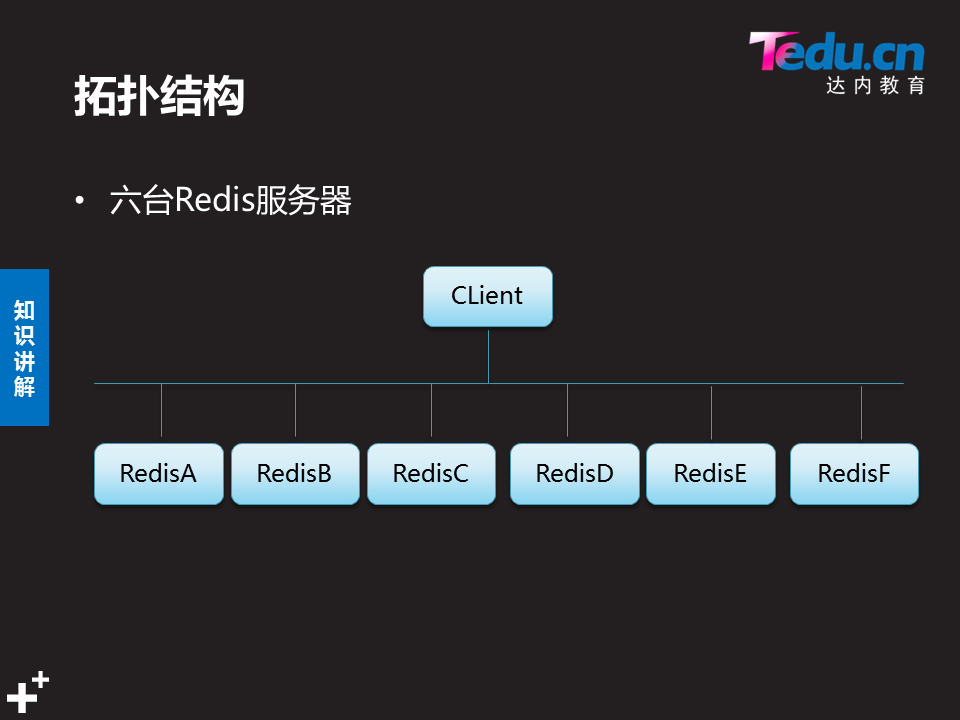
**-Redis集群的优化**

问题：当其中一组主从关系的主机全部宕机，集群停止工作；

解决方案：配置成一主多从结构，实现更高可用集群。

2、创建Redis集群

**步骤1：准备Redis服务器**



Redis服务器规划：

|  |  |  |
| --- | --- | --- |
| 主机名 | 服务器IP | 端口 |
| Redis51(管理主机) | 192.168.4.51 | 6351 |
| Redis52 | 192.168.4.52 | 6352 |
| Redis53 | 192.168.4.53 | 6353 |
| Redis54 | 192.168.4.54 | 6354 |
| Redis55 | 192.168.4.55 | 6355 |
| Redis56 | 192.168.4.56 | 6356 |

要求：所有Redis服务器均不设置密码

port均设置为本机的eth0网卡IP，不设置回环地址

保证所有redis数据库的数据一致(或为空)

**步骤2：调整配置文件**

[root@redis51 ~]# vim /etc/redis/6379.conf

cluster-enabled yes #启用集群

cluster-config-file nodes-6351.conf #指定集群信息文件

cluster-node-timeout 5000 #当前主机与其他主机通信的超时时间(ms)

[root@redis51 ~]# redis-cli -h xx -p xx shutdown

[root@redis51 ~]# /etc/init.d/redis/redis\_6379 start

[root@redis51 ~]# netstat -anputl | grep redis

#重启服务，查看redis-server端口(6351、16351)

Tips：Redis52~Redis56主机都需要配置

**步骤3：查看集群信息**

1）通过集群信息文件查看

[root@redis51 ~]# cat /var/lib/redis/6379/nodes-6351.conf

938df98db49de934 :0@0 myself,master - 0 0 0 connected

本机标识ID 初始状态

2）在redis-cli命令行查看

192.168.4.51:6351> cluster info #查看集群状态

192.168.4.51:6351> cluster nodes #查看集群节点

**步骤4：部署管理主机**

1）配置ruby脚本运行环境

[root@redis51 ~]# yum -y install ruby rubygems

[root@redis51 ~]# cd /root/redis-cluster/

[root@redis51 redis-cluster]# rpm -ivh --nodeps ruby-devel-...rpm

[root@redis51 ~]# gem install redis-3.2.1.gem

1. 拷贝集群管理脚本

[root@redis51 ~]# mkdir /root/bin/

[root@redis51 ~]# cp ./redis/redis-4.0.8/src/redis-trib.rb /root/bin/

[root@redis51 ~]# redis-trib.rb help

Tips：可以使用独立的管理主机，也可以集成于master中。

**步骤5：创建集群**

[root@redis51 ~]# redis-trib.rb create --replicas 1 \

> 192.168.4.51:6351 192.168.4.52:6352 \

> 192.168.4.53:6353 192.168.4.54:6354 \

> 192.168.4.55:6355 192.168.4.56:6356

#--replicas 1，自动为每一个master分配一个slave

#将所提供的主机随机分成3组主从关系

Tips：重复步骤2，查看集群是否创建成功。

**步骤6：测试集群**

[root@redis50 ~]# **redis-cli -c -h 集群主机ip -p 端口**

#-c访问集群中的Redis服务器

192.168.4.51:6351> set x 111

-> Redirected to slot [16287] located at 192.168.4.53:6353

OK

192.168.4.51:6351> get x

-> Redirected to slot [16287] located at 192.168.4.53:6353

"111"

**Afternoon：**

1. 管理集群

**- redis-trib.rb脚本**

语法：redis-trib.rb 选项 参数

|  |  |
| --- | --- |
| 选项 | 备注 |
| check | 检测集群 |
| add-node | 添加master主机 |
| reshard | 重新分片(hasy slot) |
| add-node --slave | 添加slave主机 |
| del-node | 删除主机 |

**- Master选举规则**

1）选举规则

master宕机后对应的slave自动选举为master；

原master修复后，随机设置为最少slave的master的从节点。

2）测试效果

[root@redis52 ~]# redis-cli -h 192.168.4.52 -p 6352 shutdown

[root@redis51 ~]# redis-trib.rb **check** 192.168.4.56:6356

M: 192.168.4.51:6351

M:192.168.4.53:6353

M: 192.168.4.56:6356

S: 192.168.4.54:6354

S: 192.168.4.55:6355

**- 添加新节点**

1. 部署新的redis服务器

192.168.4.57：6357

192.168.4.58：6358

#默认不设密码，启用集群配置

2）添加master主机，重新分片

[root@redis51 ~]# redis-trib.rb **add-node** 192.168.4.57:6357 \

> 192.168.4.52:6352

#将4.57主机加入4.52所在的集群

[root@redis51 ~]# redis-trib.rb check 192.168.4.57:6357

M: 5b1309d9c06e5c5846bd3d81c38ee0 192.168.4.57:6357

slots: (0 slots) master

[root@redis51 ~]# redis-trib.rb **reshard** 192.168.4.57:6357

How many slots do you want to move (from 1 to 16384)? 4096

What is the receiving node ID? 接收slot的主机ID

Source node #1: all 或 (ID值 \n done)

Tips：重新分片后，对应slot范围的数据也会存储到新的master

3）添加slave主机

[root@redis51 ~]# redis-trib.rb **add-node --slave** \

> [--master-id id值] 192.168.4.58:6358 192.168.4.51:6351

#不写--master-id，新节点随机添加为最少slave的master的从节点

**- 移除节点**

1. 移除slave主机

[root@redis51 ~]# redis-trib.rb **del-node** 192.168.4.51:6351 \

> 主机ID值

#移除slave后，该主机的redis服务停止；

#移除slave后，数据仍存在于该主机，但无法读写；

#当主机还原为独立的redis服务器，数据才可读写。

2）移除master主机

[root@redis51 ~]# redis-trib.rb reshard 192.168.4.51:6351

#重新分片，释放该主机的hash slot

[root@redis51 ~]# redis-trib.rb del-node 192.168.4.51:6351 \

> 5b1309d9c06e5c5846bd3dce8b84f97d81c38ee0

#移除master主机

3）还原成独立redis服务器

192.168.4.58:6358> cluster reset

#清空集群数据+信息，不执行则保留数据

[root@redis58 ~]# vim /etc/redis/6379.conf

#cluster-enabled yes

#cluster-config-file nodes-6358.conf

#cluster-node-timeout 5000

[root@redis57 6379]# /etc/init.d/redis\_6379 start

**- 移除后还原**

192.168.4.58:6357> cluster reset

#清空集群信息

[root@redis57 ~]# rm -rf /var/lib/redis/6379/nodes-6357.conf

#清除hash slot的信息

[root@redis51 ~]# redis-trib.rb add-node 192.168.4.57:6357 \

> 192.168.4.51:6351

#在管理节点重新加入到集群