**redis介绍（远程字典服务器）**

使用c语言编写

是一款高性能的(key/values)分布式内存数据库

支持数据持久化，可以把内存里的数据保存到硬盘中

也支持list、hash、set、zset数据类型

支持master-salve模式备份数据

中文网站www.redis.cn

**常用指令操作**

set keyname keyvalues //存储变量

get keyname //获取变量

select 数据库编号0-15 //切换库

keys \* //打印所有变量

keys a? //打印指定变量

exists keyname //测试变量是否存在

ttl keyname //查看变量生存时间

type keyname //查看变量类型

moved keyname dbname //移动变量

expire keyname 10 //设置变量有效时间

del keyname //删除变量

flushall //删除内存所有变量

save //保存所有变量

shutdown //关闭redis服务

**配置文件解析**

常用配置选项

port 6379 //端口

bind 127.0.0.1 //ip地址

tcp-backlog 511 //tcp连接总数

timeout 0 //连接超时时间

tcp-keepalive 300 //长时间连接

daemonize yes //守护进程方式运行

databases 16 //数据库个数

logfile /var/log/redis\_6379.log //pid文件

maxclients 10000 //并发连接数量

dir /var/lib/redis/6379 //数据库目录

**内存管理**

内存清除策略

volatile-lru //最近最少使用（针对设置了TTL的key）

allkeys-lru //删除最少使用的key

volatile-random //在设置了TTL的key里随机移除

allkeys-random //随机移除key

volatile-ttl(minor TTL) //移除最近国企的key

noeviction //不删除，写满时报错

选项默认值

maxmemory <bytes> //最大内存

maxmemory-policy noeviction //定义使用策略

maxmemory-samples 5 //选取模版数据的个数（针对lru和ttl策略）

**安装**

从源码包编译安装

tar -xf redis-4.0.8.tar.gz

cd redis-4.0.8

make && make install

初始化配置（初始化完成默认开启服务）

端口：6379

主配置文件：/etc/redis/6379.conf

日志：/var/log/redis\_6379.log

数据库目录（内存往硬盘存储的目录）：/var/lib/redis/6379

pid文件：/var/run/redis\_6379.pid

启动程序：/usr/local/bin/redis-server

utils/install\_server.sh //初始化

启动/停止服务

启动服务：/etc/init.d/redis\_<portnumber> start

停止服务：/etc/init.d/redis\_<portnumber> stop

redis-cli -h ip地址 -p 端口 shutdown

redis-cli -h ip地址 -p 端口

>shutdown

**连接数据库服务**

ps -C redis

netstat -utnlp | grep redis

redis-cli //连接本机的redis数据库服务

redis-cli -h ip地址 -p 端口 //连接远程的redis数据库服务器

**设置连接密码**

vim /etc/redis/6379.conf

requirepass 密码

/etc/init.d/redis\_<portnumber> restart

redis-cli -h ip地址 -p 端口

>auth 密码

redis-cli -h ip地址 -p 端口 -a 密码

**部署LNMP+redis**

部署LNMP

安装nginx和PHP

yum -y install gcc gcc-c+ pcre-devel zlib-devel

tar -xf nginx-1.12.2.tar.gz

cd nginx-1.12.2

./configure --prefix=/usr/local/nginx

make && make install

yum -y install php-common

yum -y install php-fpm-5.4.16-42.el7.x86\_64.rpm

修改nginx配置文件，使其支持PHP

启动服务

/usr/local/nginx/sbin/nginx

netstat -utnlp | grep nginx

systemctl start php-fpm

netstat -utnlp | grep 9000

配置PHP支持redis

运行redis服务

装包并启动服务

tar -xf redis-4.0.8.tar.gz

cd redis-4.0.8

make

make install

./utils/install\_server.sh //初始化

service redis\_6379 start

配置PHP支持redis

安装php扩展

yum -y install autoconf automake

yum -y install php-devel-5.4.16-46.el7.x86\_64.rpm

tar -xf phpredis-2.2.4.tar.gz

cd phpredis-2.2.4

phpize

./configure --with-php-config=/usr/bin/php-config

make && make install

vim /etc/php.ini

extension\_dir=”扩展路径”

extension=”redis.so”

systemctl start php-fpm

测试配置

vim /usr/local/nginx/html/redis.php

<?php

$redis=new redis();

$redis->connect(‘127.0.0.1’,6379);

$redis->set(‘redistest’,’666666’);

echo $redis->get(redistest);

?>

**redis高可用集群**

准备环境

六台redis

IP及端口规划

A：192.168.1.221 6351

B：192.168.1.238 6352

C：192.168.1.213 6353

D：192.168.1.83 6354

E：192.168.1.252 6355

F：192.168.1.72 6356

工作原理

用户存数据到数据库根据hash slot，放在不同的数据库服务器内，取得时候，在根据相应的公式进行计算，再去相应的数据库读出来，从库是同步主库的数据，所以不需要hash slot，当主机宕机后，从库自动提升为主库，原来主库重新启动后，将配置为当前主库的从库。当主库宕机后，没有从库，集群将不能正常运转。存储：key/CRC16 值%16384查看余数在哪个hash槽的范围内，就存到哪个节点，hash槽范围0-16383。

创建redis集群

装包

为6台主机安装并运行redis服务

yum -y install gcc gcc-g++

tar -xf redis-4.0.8.tar.gz

cd redis-4.0.8

make && make install

调整配置文件

vim /etc/redis/redis.conf

bind IP地址 //只写物理接口IP地址

daemonize yes//守护进程运行方式

port xxxx//端口号不要使用默认的6379

cluster-enabled yes//启用集群

cluster-config-file nodes-xxxx.conf//指定集群信息

cluster-node-timeout 5000//请求超时5秒

查看服务信息

/etc/init.d/redis\_6379 status

netstat -utnlp | grep redis-server

查看集群信息

redis-cli -c -h ip地址 -p 端口

>cluster info

>cluster nodes

创建集群

在选中的一台redis服务器上，执行创建集群脚本

部署ruby脚本运行环境

yum -y install ruby rubygems

rpm -ivh --nodeps ruby-devel-2.0.0.648-30.el7.x86\_64.rpm

gem install redis-3.2.1.gem

cd redis-3.2.0/src

./redis-trib.rb create --replicas 1 host:port host:port ......

--replicas 1，自动为每一个master节点分配1个slave节点

创建集群

./redis-trib.rb create --replicas 1 192.168.1.221:6351 192.168.1.238:6352 192.168.1.213:6353 192.168.1.83:6354 192.168.1.252:6355 192.168.1.72:6356

查看集群信息

任意一台主机访问本机的redis服务即可查看

cluster info//查看集群信息

cluster nodes//查看集群节点信息

测试集群

在客户端访问任意一台masster主机存数据

redis-cli -c -h ip地址 -p 端口

set key values//存数据

在客户端访问任意一台master主机取数据（从库不允许查看键值）

redis-cli -c -h ip地址 -p 端口

get key//取数据

管理集群

管理命令

redis-cli命令

查看命令帮助

redis-cli -h

常用选项

-h ip地址

-p 端口

-c 集群模式

redis-trib-rb脚本

语法格式

redis-trib.rb 选项 参数

选项

create 创建集群

add-node 添加master主机

check 检测集群

reshard 重新分片

add-noide --slave 添加slave主机

del-node 删除主机

master选举测试

选举master主机

停止master主机的redis服务

master宕机后对应的slave自动被选举为master

原master启动后 会自动配置为当前master的slave

查看集群主机的信息

redis-cli -h master\_ip -p master\_port

redis-trib.rb check IP地址:端口

添加新节点

添加master主机

部署一台新的redis服务器

装包

初始化

启用集群配置重启服务

添加master主机步骤

添加master主机

添加时指定主机角色，默认新主机为master

./redis-trib.rb add-node 新主机IP:端口 192.168.1.221:63511

检查主机

./redis-trib.rb check 192.168.1.221:6351

新主机角色为master

无槽位数量

重新分片

./redis-trib.rb reshard 192.168.1.221:6351

从当前的集群中拿出多少个hash slots

新主库的ID值

all从当前的集群中所有主机里获取hash slots

yes确认信息

添加slave主机（哪个主机少往哪个添加，如果一个，随机分配，可以指定，如果不指定主节点的id的话，会吧新节点随机添加为节点最少的主的从）

部署一台新redis服务器

装包

初始化

启用集群配置

运行服务

添加slave主机

./redis-trib.rb add-node --slave [--master-id id值] 新主机ip地址:端口 192.168.1.221:6351

移除节点

移除master主机

配置步骤

重新分片释放占用的hash槽、

指定移出hash槽个数

指定接收hash槽主机ID

指定移出hash槽主机ID

rdis-trib.rb reshard 192.168.1.221:6351

move

rece ID

source ID（如果不需要的话写done）

移除master主机

redis-trib.rb del-node 192.168.1.221:6351 要移出master主机ID

移除slave主机

从节点主机没有槽位范围，直接移出即可

rediis-trib.rb delnode 192.168.1.221.6351 要移除主机ID值