NoSql

RDBMS (Relational DataBase Management System) 关系数据库管理系统

按照预先设置的组织结构,将数据存储在物理介质上

数据之间可以做关联操作

RDBMS转件 (主流)

Mysql

MariaDB

Oracle

DB2

SQL server

NoSql (NoSql = Not Only SQL)

--意思是不仅仅是sql

--泛指非关系型数据库

--不需要预先定义数据存储结构

--每条记录可以有不同的数据类型和字段类型

NoSql软件 (主流软件)

Memcached

Redis

MongoDB

CouchDB

Neo4j

FlockDB

Redis介绍

---Remota Dicionary Server (远程字典服务器)

---是一款高性能的分布式内存数据库

--支持数据持久化(定期把内存里的数据存储到硬盘)

--支持多种数据类型 string list hash....

--支持master-slave模式数据备份

配置服务器运行参数

]# ./utils/install\_server.sh //初始化配置

---端口 6379

---主配置文件 /etc/redis/6379.conf

---日志文件 /var/log/redis\_6379.log

---数据库目录 /var/lib/redis/6379

---服务器启动程序 /usr/local/bin/redis-server

---命令行连接命令 /usr/local/bin/redis-cli

管理服务

]# /etc/init.d/redis\_6379 stop //停止服务

Redis pid文件: /var/run/redis\_6379.pid

]# /etc/init.d/redis\_6379 start //启动服务

!!!若修改了 ip(默认回环地址127.0.0.1) 端口号 (默认6379) 添加了密码(requirepass) 都无法通过默认脚本关闭与开启

!!!若仍要通过脚本管控 可以修改脚本参数

43 $CLIEXEC -p 6350 -h 192.168.4.50 -a 123456 shutdown

]# ps -C redis-server //查看进程

]# netstat -ntulp|grep :6379 //查看端口

常用命令

mset key 值 .. //存储多个key

mget //获取多个key值

select 数据库编号0-15 //切换库

exists key名 //测试key名是否存在

ttl key名 //查看key的生存时间 -2 永久生效 ; -1 已经失效

type key名 //查看key的类型

move key名 库编号 //移动key到指定库

expire key名 数字 //设置key的有效时间

del key名 //删除指定的key

flushall //删除内存里所有的key

flushdb //删除所有库的所有key

save //保存所有key到硬盘

内存管理

内存清楚策略

-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(针对所有的key)

-allkeys-lfu //从所有key中清楚使用频率最少的key

-volatile-lfu //从所有配置了过期时间的key中清除使用频率最少的key

-volatile-random //在设置了TTL的key里随机移除

-allkeys-random //随意移除key

-volatile-ttl(minor TTL) //移除最近过期的key

-noeviction //不删除,写满时报错

优化设置

--maxmemory //最大内存

--maxmemory-policy //定义使用策略

--maxmemory-samples //选取key模板的个数(针对lru和ttl的策略)

phpize 生成配置文件与 ./configure 命令

######################################################################################################

Redis 集群

1.加入的集群的Redis需要开启集群功能 (集群通信端口 1XXXX(原端口的基础上 加 1万))

2.连接集群中任意服务器必须加 -c 参数(包括本地登录),否则获取集群中数据会报错

3.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,并且会自动同步当前Master的数据

4.存储规则:crc16 算法 与 变量名 进行 hash 计算 得出的数字 与 总槽数取余 获得一个槽值然后对应存储

5.添加时不指定主机角色,默认为Master,并且需要手动分配hash槽

6.添加Slave主机时不指定Master节点的id默认把新节点随机添加到从节点最少的主库,[--Master-id id值]参数指定主机

7.添加服务器时集群信息必须为空(rm -rf /var/lib/redis/6379/\* 或者进入Redis数据库后执行 cluster reset 清空集群信息) (存在nodes-6379.conf文件记录了老的集群信息)

8.移除Slave服务器 (从服务器没有hash槽,直接移除即可,移除时指定从服务器id值,会停止移除主机的Redis服务)

9.移除Master服务器,必须先释放战友的hash槽,然后移除主机

10.若当前集群配置不平衡(例如Slave分布状态 1 0 0 2 ),不指定Master添加服务器时(Redis-trib.rb add-node --slave ip:port ip:port),会重新分配当前集群配置(Slave分布状态为 1 1 1 1).

11.查看当前数据库的主从信息命令:info replication

1.创建集群

redis-trib.rb create --replicas 1 (定义每台主库的从库个数) 192.168.4.51:6351....(等)

2.查看集群信息(在管理主机查看)

redis-trib.rb info 192.168.4.51:6351(集群中任意一台) //查看集群信息

redis-trib.rb check 192.168.4.51:6351(集群中任意一台) //查看集群主机对应角色

3.添加master服务器(添加时不指定主机角色,默认为Master,并且需要手动分配hash槽)

redis-trib.rb add-node 192.168.4.58:6358(新的开启集群功能的服务器) 192.168.4.51:6351(集群中任意一台)

redis-trib.rb reshard 192.168.4.51:6351(集群中任意一台) //重新分片

--移出的hash槽个数

--接受的hash槽主机id

--移出的hash槽主机id

How many slots do you want to move (from 1 to 16384)? 4096 //移出的hash槽个数

What is the receiving node ID? f9c26dafbfc85e29f16c639351cd4b7e4709d5d7 //接受的hash槽主机id(check命令查看)

Please enter all the source node IDs.

Type 'all' to use all the nodes as source nodes for the hash slots.

Type 'done' once you entered all the source nodes IDs.

Source node #1:all //移出的hash槽主机id (全部 all)

4添加从服务器(添加Slave主机时不指定Master节点的id默认把新节点随机添加到从节点最少的主库,[--Master-id id值]参数指定主机)

redis-trib.rb add-node --slave --master-id f9c26dafbfc85e29f16c639351cd4b7e4709d5d7 192.168.4.59:6359 192.168.4.51:6351(任意一台)

5移除Slave服务器 (从服务器没有hash槽,直接移除即可,移除时指定从服务器id值,会停止移除主机的Redis服务)

redis-trib.rb del-node 192.168.4.51:6351(集群中任意一台) 4807e16c8dd6395944665a1554753d96ec4dbe2a(对应的id值 check查看)

6移除Master服务器(必须先释放战友的hash槽,然后移除主机)

redis-trib.rb reshard 192.168.4.51:6351(任意一台)

--移出的hash槽个数

--接受的hash槽主机id

--移出的hash槽主机id

How many slots do you want to move (from 1 to 16384)? 4096 //移出的hash槽个数

What is the receiving node ID? f9c26dafbfc85e29f16c639351cd4b7e4709d5d7 //接受的hash槽主机id(check命令查看)

Please enter all the source node IDs.

Type 'all' to use all the nodes as source nodes for the hash slots.

Type 'done' once you entered all the source nodes IDs.

Source node #1:f9c26dafbfc85e29f16c639351cd4b7e4709d5d7 //指定移出的Master主机id

Source node #2:doen

############################################################################################################################################################