Redis

#### Redis简介

1. 简介

Redis是一种key-value形式的nosql数据库，它是基于内存存储的，所以速度非常快，共有五种数据类型，存储数据可以设置过期时间，而且redis支持持久化，集群搭建等。Redis底层是基于字典存储的而字典又是基于哈希表的。

#### Redis简介

2. 数据结构

1） string

string是一个基础的数据类型，也是任何存储系统必备的类型

常用命令：

set name “james” //存入键值对

get name //取出name的值

del name //删除name

2） list

list及列表，redis的list底层结构为压缩链表（连续内存），链表的优势就是插入和删除快速，而查找慢

常用命令：

lpush mylist “1” //在列表左侧添加

rpush mylist “2” //在列表右侧添加

lrange mylist 0 1 //从列表左侧列出0-1的元素

lrem mylist -2 “1” //从列表尾部向头部删除为1的值，数量为2

作用：

可以用list实现有序队列实现先后顺序，而不必像mysql使用order by排序。

可以用lrange实现分页。

3） set

set集合是一个无序集合，里面的元素没有先后顺序。

常用命令：

sadd myset “one” //向集合中添加一个元素

smembers myset //列出集合中的所有元素

sismember myset “one” //判断集合中是否有one元素

srem myset “one” //从集合中删除元素one

sunion myset yourset //对两个集合求并集

4） zset

zset是一个有序集合，每个元素关联一个score作为排序依据，底层数据结构为跳表。

常用命令：

zadd myzset 1 james //向列表中添加元素

zadd myzset 2 july

zrange myzset 0 -1 withscores //查询列表中的所有元素+scores

zrem myzset july //移除july元素

5） hash

hash数据结构和java中的map相似，存的是字符串和字符串值之间的映射，底层是一个散列表。

常用命令：

hset myhash name james //向列表中存入元素（hmset存多个）

hdel myhash name //删除列表中的元素

hgetall myhash //查询hash中的所有元素

hget myhash name //查询hash中的指定元素

4． redis的事务处理

事务具有原子性，即要么全做要么全不做。

1）multi：用于开启一个事务，后接多个命令组成一个事务

2）exec：执行一个事务

3）discard：放弃一个事务

5. redis的持久化

1） RDB

RDB即为快照方式持久化，redis数据库会创建一个子进程来进行持久化，持久化的过程会把数据都保存到一个临时文件中，待持久化完毕，用该文件替换上次持久化好的文件。

优点：在不需要数据完整性的条件下，适合大规模数据的恢复

缺点：不能保证数据完整性，假若五分钟执行一次快照，那么最近五分钟的数据都将丢失。

2） AOF

AOF是把redis运行的一条条指令追加到AOF文件中，默认每秒执行一次，redis提供重写机制来避免追加导致的文件过大问题，例如调用100次incr指令使数字加100次，就可以重写为一条set指令。

优点：可以尽量保证数据完整性，即使丢失也会丢失最近1s的数据

缺点：AOF文件比RDB大，而且恢复速度也比RDB方式慢。

6. redis的集群学习

实战篇学习搭建了一个redis集群，对集群进行操作发现，只能在master进行读写操作，而slave只能进行读操作，而且写入master的数据在slave中立即可见。稍微了解一下redis读写分离的知识。

1. 主从复制

主从复制分为全量同步与增量同步。

全量同步：

全量同步发生在slave初始化的时候，这时slave需要将master上的所有数据都复制一份，大致过成如下。

slave连接master并发送一条sync指令；

master接收到sync命令即开始执行bgsave命令生产RDB文件并在缓冲区存储所有在此期间的写命令；

master向slave发送快照文件，并继续在缓冲区存储写命令；

slave接受RDB文件丢弃所有旧数据，载入快照文件；

master发送完RDB文件即开始发送缓冲区内的命令

slave载入完快照，开始执行缓冲区中的命令

增量同步：

主要发生在redis集群正常开始工作时，此时master每执行一条写命令都会向slave发送该命令，slave执行该命令以保持同步

主从复制的作用：

高性能：读写分离，提高性能。

高可用：心跳检测，容灾恢复。

1. 哨兵机制-sentinel

redis提供了哨兵机制来保证集群的高可用性，sentinel不断的监视主从数据库的运行状态，当我们的主数据库挂掉之后，sentinel会主动选举出一个新的master来保证集群的正确运行。

window下sentinel的搭建：

在redis-7004根目录中新建sentinel.conf文件

port 27004

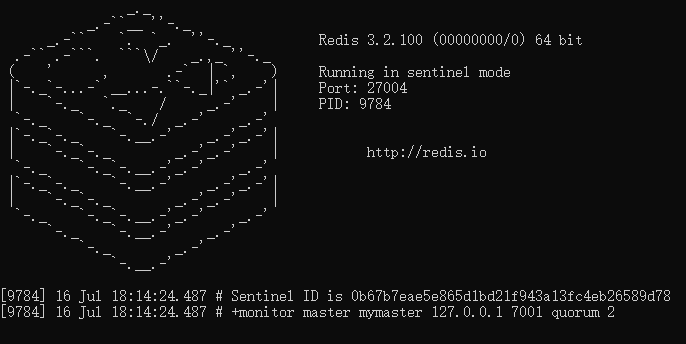
sentinel myid f9c82ea32780a86336c8661c56eb0fb3f3adde9c

sentinel monitor mymaster 127.0.0.1 7001 2

sentinel down-after-milliseconds mymaster 5000

sentinel failover-timeout mymaster 15000

redis-server.exe sentinel.conf --sentinel 启动



7. redis的数据淘汰机制

volatile-lru：使用LRU算法进行数据淘汰（淘汰上次使用时间最早的，且使用次数最少的key），只淘汰设定了有效期的key

allkeys-lru：使用LRU算法进行数据淘汰，所有的key都可以被淘汰

volatile-random：随机淘汰数据，只淘汰设定了有效期的key

allkeys-random：随机淘汰数据，所有的key都可以被淘汰

volatile-ttl：淘汰剩余有效期最短的key

作用：redis的数据淘汰机制可以设置最大存储容量的同时，保证存储的一定是热点数据。

8. mysql+redis

mysql+redis实现读写分离架构，mysql集群提供读与写服务，redis提供读服务，当读取redis没有数据的时候从mysql读，并且把数据写入redis。

也可以在写入mysql数据的时候，把消息传入mq中，等待写入redis。