redis

1.什么叫高可用性

由于某种原因服务端崩溃或者需要维护升级的过程，在这个过程中使之影响的时间达到最短

2.redis的数据结构

- key–string　set key value get setnx
- key–map　hset key field value　 hget/hsetnx
- key–list　 lpush/rpush/lpop/rpop　key value
- key–set　 sadd/smembers/spop　key value

- key-zset　zadd/scores/zrem　key　score value

3.redis集成springboot

redis核心依赖：Jedis

redis存储两种方式：

- 字符串JSON　　redis存储，对象---->字符串JSON　fastjson/databind
- 字节数组　redis存储，对象------>字节数组　　spring-context　有个工具类可以序列化反序列化：serializationutils

Jedis连接池操作　JedisPool--->jedis　避免什么被频繁的创建和销毁么

redis管道操作

4.redis的搭建配置

YAML　复制代码

```yaml
valumes:
  -  ./conf/redis.conf : /usr/local/redis/redis.conf
  yml当前目录下：容器redis配置的内部目录　　数据卷映射
  redis的密码：
  redis.conf:requirepass密码
```

5.redis的持久化机制

1. RDB

   RDB持久化机制是Redis默认的持久化机制，存储形式是以数据快照的形式存在后缀为rdb的文件中，RDB的存储条件：900s内 次数达到一次，300s 次数达到10次，60s 次数达到10000次存在问题：当我们的数据操作达到9次单又未到900s时，这个时候redis宕机了，数据丢失掉了

2. AOF

   AOF持久化机制是Redis的另一种持久化机制，存储形式是以指令快照的形式存储在AOF的文件中，我们存储指令的时候可能与当时的指令有些地方不同，但是最终执行的结果redis存储的数据是一致的

注意：我们可以对AOF和RDB文件进行数据移至，因为Redis在启动的时候都会去加载这个文件。在开发过程中我们一般是两种机制都混合使用，当然redis会先以AOF文件为标准。

6.redis小知识

1. redis事务:一次事务操作，将所要执行的命令放在一个队列中，如果你执行全部执行，要么全部取消。
2. redis主从架构：主负责写读，从负责读（存在单点故障问题）

3. redis哨兵模式：主负责读写，从负责读，解决主从架构单点故障问题，哨兵投票的选举方式决定
4. redis集群：通过提供16384个hash槽，CRC16算法%16384平均分配到每个节点上

7.rediskey的生存时间到了，redis会立即删除吗?

不会，redis删除key一般为两种：

- 定期删除：redis每隔一段时间就会去查询过了期的key在100ms内默认查看3个key
- 惰性删除：我们每次操作redis都要操作key,redis会先去看key过期没，如果过期了，直接剔除掉，返回一个null

8.redis的淘汰机制

redis的淘汰机制有8种，我在这说我记得住的几种

- redis内存满的时候，当我们添加一个key的时候，干掉一个过期的key
- 干掉一个最近最少使用的key
- 直接抛异常（默认的）

9.redis遇到的一些线上问题

1. 缓存穿透----->请求一些redis没有的数据：列如id:-1,2.30频繁请求不断查询数据库：解决方案将id范围限制，或者在redis中存一份，先查redis在走数据库
2. 缓存击穿----->一个时间点，热点数据key的生存期到了，解决方案：分布式锁
3. 缓存雪崩----->在一个时间点，大量的热点key同时到期了,解决方案：控制生存期不同
4. 缓存倾斜------>大量的热点数据在一台redis上，redis撑不住宕机：解决方案：集群

10.redis单线程工作，为什么工作效率非常高

redis是基于内存操作，多路复用（NIO机制）,优秀的数据结构：redis存储的key方式是基于hash表存储的，都知道hash表存储查询效率一般比b树存储效率还要高，不仅key有自己的数据结构我们的value存储也有大量的数据结构做支撑，如动态字符串，hash表，压缩列表，双向链表，数组，跳表等

11.redis的string数据结构你在项目中如何使用

- 单值缓存
- 多值缓存
- 分布式锁
- 计数器
- 主键id自增

12.让你实现一个购物车功能你怎样实现

可以使用redis存储进行实现

- hset key field value　key代表一个购物车id:cartId,field代表一个商品id,value代表商品数量

- 添加购物车：hset cartid productid 1
- 获取购物车：hgetall cartid
- 删除购物车：hdel cartid productid
- 商品总数： hlen cartid
- 修改商品数量：都可以直接操作命令

13.实现一个栈和队列的功能你怎样实现

- stack(栈)=lpush+Lpop=FILO
- Queue(队列)=lpush+rpop=FIFO
- blockingMQ(阻塞队列)=lpush+brpop

注意为什么不适用jdk提供的阻塞队列数据结构，分布式情况下jdk的数据结构是不可以使用的

14.实现一个微博和微信公众号消息流

例：

用户关注了macTalk 备胎说车

以用户id为key 发布的消息为value

1. macTalk发布消息

Lpush msg{user-id} 消息id

2. 备胎说车发布消息

Lpush msg{user-id} 消息id

3. 用户查看最新消息

LRANGE msg{user-id} 0,4

15.实现微信抽奖小程序

set结构:以key为actionid,以value为用户id

- 点击参与抽奖加入集合 sadd actionid user_id
- 查看参与抽奖的所有用户 smembers actionid
- 开奖功能，不剔除中奖用户 srandmember actionid 3
- 分等级开奖功能，剔除中奖用户 spop actionid

16.微博等点赞功能

- 点赞：sadd like:{消息id} 用户id
- 取消点赞： srem like:{消息id} 用户id
- 检查用户是否点过赞：sismember like:{消息id} 用户id
- 获取所有的点赞用户：smembers like:{消息id}
- 获取点赞数： scard like:{消息id}

17.实现微博微信朋友圈关注模型

- zhangsan关注的人 zhangset ---->{guihaole,aliyun}
- wangwu关注的人 wangwuset------>{guihaole,xushu,xishi,diaochan}
- lisi关注的人 lisiset----->{zhangsan,diaochan,wangwu}
- zhangsan和wangwu 共同关注：sinter zhangset wangwuset
- zhangsan可能认识的人：sdiff zhangsanset wangwuset

18.某一天热点新闻排行榜

1. 点击新闻 zincrby hotNew:2019.8.19 1 守护中国
2. 展示排行榜前十 zrevrange hotNew:2019.8.19 0 9 withscores

19.跳表的数据结构

跳表实际上是在链表上建立了索引层，我们通过二分查找发遍历链表，提高效率也叫折半查找法，zset默认是用压缩列表存储的，当数据达到一定阈值128，压缩列表变为跳表存储

20.redis的value数据存储结构

- string 动态字符串 int str
- hash 哈希表和压缩列表
- list 压缩列表和双向链表
- set 数组
- zset 压缩列表和跳表