

# 给我介绍一下 redis

## (what redis 是什么)

redis 是一个 NOSQL 类型的数据库，是一个内存级别的数据库，内部存储是以 key value 方式进行数据存储的。

Redis 可以存储的数据类型也比较丰富，比如有字符串类型，列表类型，还是散列数据类型，比如也就是 hash，这个 hash 的存储方式我们在项目中使用的还是比较频繁。

Redis 有两种数据保存的方式，第一种：是 rdb 形式，快照的形式，redis 默认开启的也是这种方式，这种保存方式是每隔一段时间就会保存一下 key 值，性能上会高一些。

第二种：是 aof 方式，这种方式是每隔一秒钟就会保存一次数据，但是这种保存数据的方式性能上会差一些，不推荐使用这种。

## (why 为什么用 redis)

如果我们做的项目中，有访问非常频繁，数据修改稍微少的信息时，这个时候我们就会使用 redis，因为 MySQL 是将信息存储到磁盘中，读写都会涉及到 IO 流，这样效率就比较低，而 redis 存储到内存中，读写效率比较高。

当然除了 redis，还有其他一些 NO SQL 类型的数据库，像 Memcached，Mongo Db

redis 和 Memcached 作比较的话，memcache 存储的数据类型比较单一，只支持一种数据类型，而 redis 存储的数据类型比较多，支持 hash，集合，和字符串等类型。Memcache 只是做内存的存储，数据不能够缓存到本地，到服务器宕机，数据就会丢失。而 redis 支持数据缓存到本地。从性能上，memcache 存储数据量超过 100k，效率要高于 redis，而对于小数据量的存储，redis 性能要高于 memcache，在多数项目中，我们存储内容并不多，综合考虑的话会使用 redis

对于 mongoDb，它的优势也是很强的，可以存储大数据量的数据，擅长 json 数据的查询，适合做文档类型的数据存储，但是不支持事物，一般我们数据量小的时候，还是会选择 redis（或者说我没有用过，不过网上有做比较，mongoDb 可以存储大数据量的数据，擅长 json 数据的查询）

## (how 怎样使用 redis)

### 使用场景

Redis 的应用场景比较多，我们门户界面一般会用到 redis 做一个缓存，像广告位，包括门户界面中商品展示内容，这些如果网站并发量比较大，或者是访问非常频繁，如果每次都从 MySQL 数据库中查询数据，一个是对数据库压力非常大，另外查询效率也比较低，门户界面做得其实类似一个二级缓存，比如商品信息先到 redis 中查询，redis 中没有就到 mysql 中查询，mysql 中查询到就存储到 redis 中一份，下一个用户访问就可以从 redis 中查询到了，当然对于商品我们有一个时效性，redis 中就有了一个设置信息过期的 expire 方法。

除了门户，像商品详情信息，我们也会放置到 redis 中。

还有在项目中，我们做搜索的时候也用到了 redis，我们把品牌信息和规格参数存入到 redis 中，我们在用户搜索的时候可以根据用户搜索的内容，确定该类商品的分类模板，确定有哪些品牌，和哪些规格参数，

这是我们在这个项目中应用到 redis 的场景。

当然 redis 除了这些，还要其他的，比如聊天室的在线好友列表。

网站访问统计。这些操作都可以使用 redis 来操作完成。不过这些我们项目中没有涉及到。

（这个如果能弄明白就说，不然就是给自己挖坑，人家接着会问到在线好友列表怎么实现，排行榜怎么实现的，网站访问统计怎么统计）

像聊天室好友列表，这个业务很少，大多数聊天不是网页版本，我们可以创建一个 hash 类型的数据存储方式，比如 key 是聊天室的 id，field 是用户 id，value 是用户信息。当上线的时候，就可以往 redis 中存一个数据，当下线时就从 redis 中清除。网站访问统计，可以定义一个简单的 String 类型，每次有用户访问界面，做一个 incr 的递增就可以统计。

### 使用步骤

在使用 redis 时，我们是这样来使用的，首先呢，我们需要在服务器上配置一下 redis 的环境，这个我说一下大体的步骤吧，

把 redis 的压缩包 copy 到 linux 上

解压 tar 包(tar -xvf )

进入到 redis 的目录下执行 make 指令进行编译

make install 将编译内容安装到某个目录下

直接./redis-server 就可以前台启动

配置 redis.conf 文件可以配置成后端启动方式 （能记住就记住，记不住就这样来说，

copy tar 包，解压，编译，安装，配置成后端启动，然后./redis-server redis.conf 启动）

讲师：闫洪波

微信：769828176

## Redis 3.0 支持集群搭建,集群搭建大体流程是这样的

配置多台服务器 redis 环境

安装 rubby 环境

使用 rubby 脚本来搭建集群

在代码使用 redis 时,我们是这样来使用的,在 pom 文件中配置 redis 的 jar 包信息,一般,会定义一个 redis 操作的接口类,把我们工作中操作 redis 的方法(get set hset hget hvals expire incr decr)封装到接口类中,当然一家家公司下来,我们倒是积攒了不少的工具类。

在开发环境下,我们是使用单机版的服务器,在公司内网的服务器上配置 reids 环境,更惨的时候,公司连个服务器都没有,那我们就搞虚拟机,反正总是能玩。使用单机版时,会在 spring 的配置文件中,将 redis 配置成单机形式,其实就是 redis 框架提供的一个类,配置 JedisPool,就跟 MySQL 中 C3PO 连接池类似。

在上线部署时,我们会配置 reids 的集群,reids 集群配置也并不复杂,主要用到一个 [HostAndPort 的 API 类来做](#),集群的优点就是效率高,即便有一台 redis 服务器挂掉也不会影响用户访问。

集群储存的原理是有一个 hash 槽的概念,redis 默认总共有 16383 哈希槽,当存 key value 时,会根据 key 计算出 hash 值,然后将数据存到具体的某台服务器上,取得时候还是会根据 key 值计算 hash 值,还是会到存的那台服务器上取数据。

## 存在问题

我们使用 redis,默认是使用 rdb 方式进行持久化,这种形式保存数据有可能数据丢失,比如 redis 刚刚存了数据,还没来的及往 rdb 中保存,服务器宕机,如果再恢复的话,最后那点存的数据就丢失了。不过数据丢失我倒是没有遇到过,解决方案的话,可以把 rdb 存储的频率调高,也可以搞集群,另一方面,特别重要的数据我们会持久化到 mysql 数据库中

## 一、redis 非关系型数据库

### 1.1.给我介绍一下 redis

Redis 是一个非关系数据库,我们项目中主要用它来存储热点数据的,减轻数据库的压力.我们项目中很多地方用到了 redis;比如说商品的秒杀,首页热点数据的展示,还要提交订单的处理,都有去用过 redis,还有在以前的项目中我们也用过 redis 来实现单点登录,由于他是单线程的,所以在电商平台里边也经常用他做“防止超卖”,还有就是购物车也用到了 redis

## 1.2.你们在项目中是怎么操作 redis 的?有过封装么?

项目中,我们一般都通过 springData 操作的 redis,他底层也是整合 redis 官方提供的 jedis 工具包来操作 redis,我们开发的时候用的是单机版。

## 1.3.redis 持久化方式有几种?

有 RDB 和 AOF 这两种,RDB 是一种快照的方式来存储的,这也是 redis 的默认的持久化方式,每隔一段对数据进行一次存储,默认是 15S,这个也可以通过配置文件里修改,这种存储方式性能比较高

还有一种是 AOF 是即时性的持久化方式,只要数据发生改变都会保存到硬盘一份,这种方式对数据的保存完整性比较高,但是性能比较差。而 RDB 存在的问题主要是服务器宕机或者断电,会造成数据丢失

## 1.4.使用 redis 有没有遇到过缓存穿透和缓存雪崩?

有遇到过,穿透就是,因为 redis 都是按照 key 去缓存查询,如果不存在对应的 value,就应该去数据库查找。如果 key 对应的 value 是一定不存在的,并且对该 key 并发请求量很大,就会对后端系统造成很大的压力。这就叫做**缓存穿透**。当缓存服务器重启或者大量缓存集中在某一个时间段失效,这样在失效的时候,也会给后端系统(比如 DB)带来很大压力,这就叫**“缓存雪崩”**。解决办法是,对查询结果为空的情况也进行缓存,并且给缓存设置不同的有效期。当然 redis 容灾的最有效的方法还是搭建集群。在缓存失效后,通过加锁或者队列来控制读数据库写缓存的线程数量。比如对某个 key 只允许一个线程查询数据和写缓存,其他线程等待。不同的 key,设置不同的过期时间,让缓存失效的时间点尽量均匀。

## 1.5.说一下 redis 集群吗?

Redis 本身就支持集群操作 redis\_cluster,另外 redis 还支持主从复制,以前的老版本中有一个哨兵模式,在主服务器宕机时,从服务器可以自动转换为主服务器。我们公司搭建的 redis 集群是用的 ruby 脚本配合搭建的,我们一共搭建了 6 台服务器,3 主 3 备,他们之间通信的原理是有一个乒乓协议进行通信的,他们判断一个节点的状态是用投票选举机制判断的,半数以上判断一个接口是宕机了的话,备用节点就会启动,对,我再给你说下他们往里存储数据的机制吧,其实这个 redis 搭建好集群以后每个节点都存放着一个 hash 槽,每次往里存储数据的时候,redis 都会根据存储进来的 key 值算出一个 hash 值,通过这个 hash 值可以判断到底应该存储到哪一个哈希槽中,取的时候也是这么取的,这就是我了解的 redis 集群。

## 哨兵 (sentinel)

### 什么是哨兵

顾名思义,哨兵的作用就是对 Redis 的系统的运行情况的监控,它是一个独立进程。它的功能有 2 个:

- 1、 监控主数据库和从数据库是否运行正常；
- 2、 主数据出现故障后自动将从数据库转化为主数据库；

### 使用集群需要注意的事项

- 1、 集群中的节点只能使用 0 号数据库，如果执行 SELECT 切换数据库会提示错误。

## 1.6.除了用 Redis,你用过 memcached 的吗?

我知道他俩本质的区别就是 Redis 除了在内存中保存数据,还会把数据保存到硬盘中重启了服务器 redis 里的数据还有,但是 memcached 只保存到内存中,重启了服务器数据就没有了还有就是应用场景不一样: Redis 除了作为 NoSQL 数据库使用外,还能用做消息队列、数据堆栈和数据缓存等; Memcached 适合于缓存 SQL 语句、数据集、用户临时性数据、延迟查询数据和 session 等。

## 1.7.在项目那一块用过,使用场景?

我们在项目的秒杀功能用过,还有购物车的时候也用了 redis,还有搜索的时候我们存储常用的类别的时候也用过,反正就是项目中访问量比较大的数据都会考虑用 redis.

## 1.8.redis 的数据类型?

我了解的有

字符串类型、

list, 我们可以向 list 的两端添加数据

集合 set, 存放的数据是无序的, 集合中的数据是不重复的, 由于它是无序的, 所以不能通过下标来获取制定元素、

order set 有序集合、

hash

实际上我们常用的也就是 string 和 hash

redis 是通过 key-value 存储的 set key value string

hset key value hget key value

## 1.9.Redis 分布式锁如何实现?

红色字体用于理解,不用背

### 一、什么是分布式锁?

要介绍分布式锁, 首先要提到与分布式锁相对应的是线程锁、进程锁。

线程锁: 主要用来给方法、代码块加锁。当某个方法或代码使用锁, 在同一时刻仅有一个线程执行该方法或该代码段。线程锁只在同一 JVM 中有效果, 因为线程锁的实现在根本上是依靠线程之间共享内存实现的, 比如 synchronized 是共享对象头, 显示锁 Lock 是共享某个变

量 (state)。

**进程锁：**为了控制同一操作系统中多个进程访问某个共享资源，因为进程具有独立性，各个进程无法访问其他进程的资源，因此无法通过 `synchronized` 等线程锁实现进程锁。

**分布式锁：**当多个进程不在同一个系统中，用分布式锁控制多个进程对资源的访问

这个分布式锁，我大概了解过，其实我们原来传统的项目都在单台服务器上部署用 java 里的锁 `synchronized` 这个同步锁就行，但是他这个是针对对象的锁，但是我们分布式的项目需要把项目部署到多台服务器上，每台服务器的对象都不同，所以就得考虑用分布式锁，我们项目里有个代码，我大概看过里面用到过几个方法 `setnx(key, value)`：“set if not exists”，若该 `key-value` 不存在，则成功加入缓存并且返回 1，否则返回 0。`get(key)`：获得 `key` 对应的 `value` 值，若不存在则返回 `nil`。`getset(key, value)`：先获取 `key` 对应的 `value` 值，若不存在则返回 `nil`，然后将旧的 `value` 更新为新的 `value`。`expire(key, seconds)`：设置 `key-value` 的有效期为 `seconds` 秒。我们公司用的是 `Redisson` 这个框架去实现的分布式锁，基本原理就是对操作的 `key` 进行判断，要加锁的这个 `key` 不存在的话就要进行加锁，用那个 `hset mylock`，然后设置锁生命周期是 30 秒，大概就是这样，具体要操作的话，我那有现成的代码。

## 1.10.如何使用 redis 去实现一个消息队列

我知道可以用那个 `list` 类型去实现，他里面有一个 `rpoplpush` 方法，可以把一个 `list` 里的尾部元素放到另一个 `list` 里头去，比如说是 `rpoplpush cht cctv`，从一个 `cht` 尾部中拿出来一个元素放到 `cctv` 的头部，

## 1.11.redis 和 mongoDB 的区别？

`mongodb` 是非关系型数据库最像关系型数据库的一种数据库，有数据库，集合和文档的刚年，集合就相当于咱们关系型数据库里的表，文档就是一条数据，他存储数据的方式是以 `bson` 方式进行存储的。但是 `Redis` 是 `key` 和 `value` 方式存储的，在项目中我们可以把一些频繁访问的数据存到 `redis` 中，不用频繁的访问数据库了，给 `mysql` 数据库减轻压力。

## 1.12.Redis 在你们项目中起到的主要作用是什么？

主要就是缓存热点数据的，减轻服务器压力。

## 1.13.Redis 是如何同步的？

读取数据的时候先从 `redis` 里面查，若没有，再去数据库查，同时写到 `redis` 里面，并且要设置失效时间。存数据的时候要具体情况具体分析，可以选择同时插到数据库和 `redis`（要是存放到 `redis` 中，最好设置失效时间），也可以选择直接插到数据库里面，少考虑一些问题，添加内容的，把对应的 `redis` 里的数据的删除掉。第一个人查的时候从数据库里查询，把数据放到的缓存中，第二个人访问就可以直接从缓存中访问数据了



## 1.14 如何保证缓存与数据库双写时的数据一致性？

你只要用缓存，就可能会涉及到缓存与数据库双存储双写，你只要是双写，就一定会有数据一致性的问题，那么如何解决一致性问题？

一般来说，就是如果你的系统不是严格要求缓存+数据库必须一致性的话，最好不要做这个方案。如果不得不做这种方案时，那么可以使用读请求和写请求串行化，串到一个内存队列里去，这样就可以保证一定不会出现不一致的情况串行化之后，就会导致系统的吞吐量会大幅度的降低，用比正常情况下多几倍的机器去支撑线上的一个请求。

## 1.15 Redis 主从复制（读写分离）

主从复制的好处有 2 点：

- 1、 避免 redis 单点故障
- 2、 构建读写分离架构，满足读多写少的应用场景

### 复制的过程原理

- 1、 当从库和主库建立 MS 关系后，会向主数据库发送 SYNC 命令；
- 2、 主库接收到 SYNC 命令后会开始在后台保存快照（RDB 持久化过程），并将期间接收到的写命令缓存起来；
- 3、 当快照完成后，主 Redis 会将快照文件和所有缓存的写命令发送给从 Redis；
- 4、 从 Redis 接收到后，会载入快照文件并且执行收到的缓存的命令；
- 5、 之后，主 Redis 每当接收到写命令时就会将命令发送从 Redis，从而保证数据的一致；

## 复制架构中出现宕机情况，怎么办？

如果在主从复制架构中出现宕机的情况，需要分情况看：

- 1、 从 Redis 宕机
  - a) 这个相对而言比较简单，在 Redis 中从库重新启动后会自动加入到主从架构中，自动完成同步数据；
  - b) **问题？** 如果从库在断开期间，主库的变化不大，从库再次启动后，主库依然会将所有的数据做 RDB 操作吗？还是增量更新？（从库有做持久化的前提下）
    - i. 不会的，因为在 Redis2.8 版本后就实现了，主从断线后恢复的情况下实现增量复制。
- 2、 主 Redis 宕机
  - a) 这个相对而言就会复杂一些，需要以下 2 步才能完成
    - i. 第一步，在从数据库中执行 SLAVEOF NO ONE 命令，断开主从关系并且提升为主库继续服务；
    - ii. 第二步，将主库重新启动后，执行 SLAVEOF 命令，将其设置为其他库的从库，这时数据就能更新回来；
  - b) 这个手动完成恢复的过程其实是比较麻烦的并且容易出错，有没有好办法解决呢？当前有的，Redis

提高的哨兵（sentinel）的功能。