

## Redis笔记

web1.0 2000左右, 单机的web, 人家可能只是请求, 看数据

比如看新闻, 看博客, 但是没有互相评论

web2.0 2008- 2020 --> 随着业务增长, 我们需求变大

不仅仅是要看, 我们还要去评论

web3.0 人和机器之间交互

nosql

not only sql (不仅仅是sql语句), 它有各自的操作api 不像我们之前操作关系型数据库, select update, delete

cpu》内存》硬盘

redis mysql/sqlserver

关系型数据库

```
1 |  
2 | 他有自己的规范, 你学会了mysql, 你可能还会oracle, server  
3 | 它的优点不就是它的缺点
```

redis优点:

方便扩展: 不需要提前去有规范的字段, 只要是数据都可以往里面存

打个预防针

有几个注意点

1.单线程一定要比多线程性能高, 不一定

单线程的原子性操作

不需要锁+不需要上下文文的切换

单线程的原子性: 就是一个任务一个人做

单线程多进程 PK 多线程

一个任务有好多个人去做，肯定有一个去指导这些人去合理安排的做事

多线程要实现原子性，涉及到各种锁，上下文的切换性能的消耗

典型的单线程的服务，比如nginx

单线程多进程

我们可以根据我们的服务，开启多个实例

其实在我们业务中经常去这么使用的，

io多路复用/异步

打个单子，然后把单子给后厨，然后然后你稍等，菜做完了喊你，你来拿

(io多路复用)

打个单子，让你留下自己地址，然后把单子给后厨，菜做完了，直接送到你家去了（异步）

redis单线程不代表整个redis服务中只有一个线程

fd1存放了客户端信息

freebsd 开源系统：centos

秒杀 todo2

1.订单数+1

2.库存-1

困境：可能超卖

如果是多线程去做，把他打包原子性+锁

+锁是让他从多线程变成单线程

用string还是小菜鸟

---留着最后解决

///用string存了一个对象

怎么存，先json，取回来之后，然后json反序列化

//现在需要把这个对象的某一个属性的值改一下下的

//首先取出来json反序列化，然后改值，然后序列化存进去

如果你是大神，你到最后会发现，所有的那些底层都是算法+算法+算法

