重入锁

ReentrantLock，是对synchronized的升级，synchronized是通过JVM实现的，ReentrantLock是通过JDK实现的。

重入锁的特点：

重入锁指可以给同一个资源添加多个锁，并且解锁的方式与synchronized也不同，synchronized的锁是当线程执行完业务逻辑之后自动释放，ReentrantLock的锁必须手动释放。

1.可重入，可以给同一个资源同时添加多把锁，对应的解锁的次数必须与上锁的次数相同，否则就会出现程序不继续执行的情况。

2.可中断，指某个线程在等待获取锁的过程中可主动终止线程。

3.限时性，指可以判断某个线程在一定的时间段内能否获取锁，通过boolean tryLock(long time,TimeUnit unit)，time表示时间数值，unit表示时间单位，返回值为boolean类型，true表示在该时间段内获取了锁，false表示在该时间段内没有获取锁。

高并发

并行和并发。

并发concurrency 并行parallelism

- 并行：指多个操作同时执行，判断程序是否处于并行状态，就看同一个时刻是否有一个以上的工作单元在运行，单线程是永远无法达到并行状态的。
- 并发：指的是程序的结构，处理并行的能力。如果一个系统称之为并发系统，则表示该系统采用了支持并发的设计模式，所以并发并不是指多个线程同时执行，它指的是一种人为设计的程序结构，可以处理多线程的能力。

高并发是指我们设计的程序，可以支持海量的任务在同一时间段内同时执行。

高并发的标准：

1.QPS：每秒响应的http请求数量，QPS不等于并发量，并发数是指某时刻同时到到达服务器的请求数量。

2.吞吐量：单位时间内可以处理的请求数。

3.平均响应时间：系统对一个请求做出响应的平均时间。

QPS = 并发数/平均响应时间。

4.并发用户数量：系统在正常运行情况可以承载的用户数量。

提供系统并发能力的两种方式：1.垂直扩展，水平扩展。

垂直扩展

提升单机的处理能力。

1.增强单机硬件性能。

2.提升单机架构性能。

水平扩展

系统集群（分层架构，nginx反向代理分担web应用服务器的压力，数据层主从复制，读写分离，分表分库，减轻数据库服务器的压力）

线程池

优点：

1.线程的创建和销毁是需要耗费资源的，使用线程池可以有效的减少创建和销毁线程的次数，每个工作线程都可以重复使用。 2.可以根据系统的承受能力，调整线程池中工作线程的数量，防止因为消耗过多内存导致服务器崩溃。

工作流程

当提交一个任务时，线程池会创建一个新的线程执行任务，直到当前线程池中的线程数等于corePoolSize：线程池的大小。

如果当前线程数量已经到达了corePoolSize，继续提交的任务被保存到阻塞队列中，等待获取执行完成的任务释放线程，从而执行等待任务。

如果阻塞队列满了，那就再创建新的线程去执行任务，直到线程池中的线程数达到maximumPoolSize，这时候如果再有任务来，只能执行拒绝方案。

ThreadPoolExecutor类

java.util.concurrent.ThreadPoolExecutor类是Java描述线程池中最核心的一个类。

核心参数：

- corePoolSize：核心池大小。
- maxmumPoolSize：线程池最大线程数，线程池中线程数量的上限。

ThreadPoolExecutor提供了动态修改线程池容量的方法：

setCorePoolSize()修改corePoolSize

setMaxmumPoolSize()修改maxmumPoolSize

- keepAliveTime：10s

  corePoolSize：5，线程池中有6个线程，如果某个线程的闲置时间达到了10s，则释放该线程。

  直到线程池中的线程数量小于等于corePoolSize，keepAliveTime失效。
- unit：参数keepAliveTime的时间单位，TimeUnit中有7个常量来表示7种不同的时间单位。

TimeUnit.DAYS 天

TimeUnit.HOURS 小时

TimeUnit.MINUTES 分钟

TimeUnit.SECONDS 秒

TimeUnit.MILLSECONDS 毫秒

TimeUnit.MICROSECONDS 微秒

TimeUnit.NANOSECONDS 纳秒