并发编程与高并发解决方案

首先，你得对并发，与高并发的概念了解，什么是并发，什么是高并发？

并发：是指多个线程直接操作相同的资源，保证线程安全，合理利用资源。

高并发：是指服务能同时处理很多请求，提高程序性能。

要写好并发程序你需要具备哪些知识？

1. 线程安全： 这是并发编程最基本的要求。那么怎么来控制我们的线程安全呢？
   1. 针对线程安全与单机系统资源利用，java为我们提供了并发包。我们需要注意的地方有以下几点。
      1. 线程封闭：什么是线程封闭？ 线程封闭有什么意义？
         1. 简单来说，线程是一种微进程，拥有独立的指令执行序列，工作内存。

由于在多核处理器上，可以同时并发执行多个线程，由于主内存与工作内存及时同步性问题，会导致线程安全问题，要开发出多线程安全的程序，需要我们人为的去封闭线程。保证线程安全。

Ii 线程调度：为啥要线程调度？ 有什么意义？

1. 线程调度是在线程封闭的情况下，来调度线程，是我们的多线程程序能够灵活的处理业务。比如一个计算任务，需要多个线程来完成，那么结果统计的线程就必须等子计算全部计算完成，这个就设计到线程的调度。

Iii 其他：

我们需要掌握的工具，

1. 同步容器：Collections.**synchronizedXXX(),vector,hashtable,stack等。**
2. 并发容器：CopyOnwriteXXX,ConcurrentSkipXXX,ConcurrentHashXXX
3. AQS: sync Queue,ConditionQueue,等
4. J.U.C：
   1. aotmic：原子包：包含了很多原子更新的类。
      1. 如：AotmicReference, CAS原理， Unsafe类，CAS的ABA问题，等。
   2. Tools：
      1. 线程调度工具：线程闭锁CountDownLatch，线程张珊：CyclicBarroer,信号量：Smamphore,执行器Exexutors 工具类，交换器Exchequer。
   3. Collections：
      1. Queue：阻塞队列，并发链表队列，双端队列。
   4. LOCK：
      1. Reentrantlock，可重入锁，读锁，写锁。Condition等
   5. Executor：
      1. 定时任务执行器:schedluledXXX
      2. FutureTask等。
5. 等等

在上面的并发编程的基础上，那我们如何来实现高并发呢？

对于高并发，我们重要的是掌握计算机原理，根据业务状况来处理各种并发问题。

那我们会涉及到那些知识点呢？

1. 扩容：
   1. 垂直扩容: （纵向扩展），来提高系统部件的能力
   2. 水平扩容：（横向扩展） 也就是增加系统成员，来实现系统高性能。
2. 缓存：
3. 队列：
4. 拆分：
5. 服务降级与熔断：
6. 数据库切库以及分表分库
7. 等等。

多线程编程的实践总结：

1. 使用本地变量
2. 使用不可变类
3. 最小化锁的作用域，细化锁粒度
4. 不要newThread 用executor
5. 宁可使用同步手段也不要用线程的wait、notify方法
6. 使用blockingQueue实现生产消费者模式
7. 使用并发容器，而不是同步容器
8. 使用semaphore并发控制
9. 宁可使用同步代码块而不是同步方法
10. 避免使用静态变量。