# Java基础

## 为什么JVM调优经常会将-Xms和-Xmx参数设置成一样；

<http://unixboy.iteye.com/blog/174173/>

## Java线程池的核心属性以及处理流程；

<https://www.cnblogs.com/dolphin0520/p/3932921.html>

<https://blog.csdn.net/zhangliangzi/article/details/52389766>

## Java内存模型，方法区存什么；

<https://blog.csdn.net/hxpjava1/article/details/55189077>

<https://www.cnblogs.com/MazeHong/p/JVM.html>

在方法区中，存储了每个类的信息（包括类的名称、方法信息、字段信息）、静态变量、常量以及编译器编译后的代码等。

## Java GC ROOT

<https://www.zhihu.com/question/50381439>

<https://www.cnblogs.com/w-wfy/p/6415768.html>

在Java语言中，可以作为GC Roots的对象包括下面几种：

虚拟机栈（栈帧中的本地变量表）中引用的对象；

方法区中类静态属性引用的对象；

方法区中常量引用的对象；

本地方法栈中JNI（即一般说的Native方法）引用的对象；

总结就是，方法运行时，方法中引用的对象；类的静态变量引用的对象；类中常量引用的对象；Native方法中引用的对象。

## CMS垃圾回收过程；

<https://blog.csdn.net/mawming/article/details/51967472>

<https://blog.csdn.net/hqq2023623/article/details/50993165>

<http://www.importnew.com/27822.html>

<https://blog.csdn.net/zqz_zqz/article/details/70568819>

CMS 处理过程有七个步骤：

1. 初始标记(CMS-initial-mark) ,会导致swt；

2. 并发标记(CMS-concurrent-mark)，与用户线程同时运行；

3. 预清理（CMS-concurrent-preclean），与用户线程同时运行；

4. 可被终止的预清理（CMS-concurrent-abortable-preclean） 与用户线程同时运行；

5. 重新标记(CMS-remark) ，会导致swt；

6. 并发清除(CMS-concurrent-sweep)，与用户线程同时运行；

7. 并发重置状态等待下次CMS的触发(CMS-concurrent-reset)，与用户线程同时运行；

## Full GC次数太多了，如何优化；

<https://blog.csdn.net/endlu/article/details/51144918>

<https://blog.csdn.net/n8765/article/details/50911742>

System.gc()方法的调用

老年代代空间不足

永生区空间不足

CMS GC时出现promotion failed和concurrent mode failure

统计得到的Minor GC晋升到旧生代的平均大小大于老年代的剩余空间

堆中分配很大的对象

## 直接内存如何管理的；

<http://www.importnew.com/21998.html>

直接内存（堆外内存）

<https://www.cnblogs.com/z-sm/p/6235157.html?utm_source=itdadao&utm_medium=referral>

ByteBuffer.allocateDirect(1000);

## Java线程池的几个参数的意义和实现机制；

## Java线程池使用无界任务队列和有界任务队列的优劣对比；

<https://blog.csdn.net/kusedexingfu/article/details/72491864>

当线程池的任务缓存队列已满并且线程池中的线程数目达到maximumPoolSize，如果还有任务到来就会采取任务拒绝策略。

无界队列导致资源耗尽。

## CountDownLatch和CyclicBarrier的区别；

<https://blog.csdn.net/tolcf/article/details/50925145>

<https://blog.csdn.net/xianymo/article/details/46865469>

## Java中有哪些同步方案（重量级锁、显式锁、并发容器、并发同步器、CAS、volatile、AQS等）

## 如果你的项目出现了内存泄露，怎么监控这个问题呢；

<https://blog.csdn.net/honlin/article/details/2749687>

<https://blog.csdn.net/caomiao2006/article/details/48454329>

<https://blog.csdn.net/dreamxiang68/article/details/7615903>

<https://blog.csdn.net/chwshuang/article/details/44202491>

<https://www.cnblogs.com/200911/p/3965108.html>

日志

Dump文件

Eclipse MAT

## 标记清除和标记整理的区别和优缺点，为何标记整理会发生stop the world；

<https://blog.csdn.net/u010841296/article/details/50945390>

标记-清除算法：

先标记出所有需要回收的对象，在标记完成后统一回收所有被标记的对象。

缺点：回收了被标记的对象后，由于未经过整理，所以导致很多内存碎片

标记-整理算法：

标记整理算法的“标记”过程和标记-清除算法一致，只是后面并不是直接对可回收对象进行整理，而是让所有存活的对象都向一段移动，然后直接清理掉端边界意外的内存。

## 线程池，如何根据CPU的核数来设计线程大小，如果是计算机密集型的呢，如果是IO密集型的呢？

<https://blog.csdn.net/lc0817/article/details/50706720>

<https://blog.csdn.net/q_l_s/article/details/51538039>

<https://www.cnblogs.com/dennyzhangdd/p/6909771.html?utm_source=itdadao&utm_medium=referral>

线程数=Ncpu/（1-阻塞系数）

IO密集型=2Ncpu（可以测试后自己控制大小，2Ncpu一般没问题）（常出现于线程中：数据库数据交互、文件上传下载、网络数据传输等等）

计算密集型=Ncpu（常出现于线程中：复杂算法）

## 让你设计一个cache如何设计；

<https://www.cnblogs.com/davidwang456/p/4001342.html>

<https://blog.csdn.net/xuchishao/article/details/48292197>

<http://www.importnew.com/18758.html>

LinkedHashMap.removeEldestEntry

## String中hashcode是怎么实现的；

## JDK中哪些实现了单例模式？

<https://blog.csdn.net/qqqqq1993qqqqq/article/details/77324914>

<https://blog.csdn.net/applepie1/article/details/7265779>

java.lang.reflect.Proxy

java.lang.Runtime

## 多个线程同时读写，读线程的数量远远⼤于写线程，你认为应该如何解决并发的问题？你会选择加什么样的锁？

<https://blog.csdn.net/deng375235329/article/details/4864546>

<https://blog.csdn.net/xiemeitx/article/details/1881204>

<https://blog.csdn.net/zlb824/article/details/7637303>

<https://blog.csdn.net/mink2010/article/details/51490707>

ReadWriteLock 写优先

## 线程池内的线程如果全部忙，提交⼀个新的任务，会发⽣什么？队列全部塞满了之后，还是忙，再提交会发⽣什么？

## synchronized关键字锁住的是什么东西？在字节码中是怎么表示的？在内存中的对象上表现为什么？

<https://www.cnblogs.com/javaminer/p/3889023.html>

<https://blog.csdn.net/liu88010988/article/details/50799759>

<https://blog.csdn.net/bohu83/article/details/51141836>

## wait/notify/notifyAll⽅法需不需要被包含在synchronized块中？这是为什么？

<https://blog.csdn.net/honganboy/article/details/47061441>

<https://www.cnblogs.com/xiohao/p/7118102.html>

<https://blog.csdn.net/haluoluo211/article/details/49558155>

## ExecutorService你一般是怎么⽤的？是每个Service放一个还是个项目放一个？有什么好处？

# 数据库

## InnoDB的插入缓冲和两次写的概率和意义；

<https://www.cnblogs.com/wyy123/p/7286311.html>

<https://www.cnblogs.com/snake-hand/archive/2013/06/12/3132942.html>

<https://blog.csdn.net/dbanote/article/details/9077113>

<https://blog.csdn.net/dbanote/article/details/9078091>

<http://blog.sina.com.cn/s/blog_c2839d2a0102wnsi.html>

<https://www.jianshu.com/p/6738e1dce97c>

插入缓冲：插入数据时，辅助索引不直接更新

Double write：写部分失效

## 如果建了⼀个单列索引，查询的时候查出2列，会⽤到这个单列索引吗？（会用到）

## 如果建了⼀个包含多个列的索引，查询的时候只⽤了第⼀列，能不能⽤上这个索引？查三列呢？

## 接上题，如果where条件后⾯带有⼀个 i + 5 < 100 会使⽤到这个索引吗？

## like %aaa%会使⽤索引吗? like aaa%呢?

<https://q.cnblogs.com/q/103498/>

<https://bbs.csdn.net/topics/290007200>

like 'aaa%' 会, like '%aaa' 不会。

## drop、truncate、delete的区别？

<https://blog.csdn.net/johnson_mar/article/details/74352515>

## 平时你们是怎么监控数据库的? 慢SQL是怎么排查的？（慢查询日志）

## 你们数据库是否⽀持emoji表情，如果不⽀持，如何操作?选择什么编码方式？如果支持一个表情占几个字节?(utf8mb4)；

## 如果查询很慢，你会想到的第⼀个⽅式是什么？（数据库索引）

# Linux基础

## Linux下可以在/proc目录下可以查看CPU的核心数等；cat /proc/下边会有很多系统内核信息可供显示；

<https://blog.csdn.net/scdxmoe/article/details/37560893>

<https://blog.csdn.net/hzrandd/article/details/50778517>

## 说一下栈的内存是怎么分配的；

<https://blog.csdn.net/shanchangyi/article/details/51854795>

<https://www.cnblogs.com/houjun/p/4909413.html>

由编译器自动分配释放

## Linux各个目录有了解过吗？/etc、/bin、/dev、/lib、/sbin这些常见的目录主要作用是什么？

<https://blog.csdn.net/u013239236/article/details/48845251>

<https://www.cnblogs.com/sytfyf/p/6364691.html>



## 说一下栈帧的内存是怎么分配的；

<https://blog.csdn.net/djinglan/article/details/7266488>

## Linux下排查某个死循环的线程；

<https://blog.csdn.net/zb872676223/article/details/48135757>

<https://www.cnblogs.com/caosiyang/archive/2012/10/15/2724585.html>

## 动态链接和静态链接的区别；

<https://blog.csdn.net/macmacip/article/details/54645927>

<https://www.cnblogs.com/dirge/p/6443317.html>

## 进程的内存分布；

<https://www.cnblogs.com/diegodu/p/4552490.html>

## 如何查找一个进程打开所有的文件；

<https://blog.csdn.net/Betty2017/article/details/78634327>

<https://blog.csdn.net/matrixyy/article/details/49132843>

## 说一下常使用的协议及其对应的端口；

<https://blog.csdn.net/u010817474/article/details/48311609>

<https://blog.csdn.net/qhairen/article/details/46746247>

## 为什么会有内核态，保护模式你知道吗?

<https://www.cnblogs.com/immortal-worm/p/5867418.html>

## 文件是怎么在磁盘上存储的？

## 有了进程为何还要线程呢，不同进程和线程他们之间有什么不同。

（进程是资源管理的最小单位，线程是程序执行的最小单位。在操作系统设计上，从进程演化出线程，最主要的目的就是更好的支持SMP以及减小（进程/线程）上下文切换开销。）

## InnoDB聚集索引B+树叶子节点和磁盘什么顺序相同;

<https://blog.csdn.net/m0_37899949/article/details/78991700>

## 文件系统，进程管理和调度，内存管理机制、虚地址保护模式；

# 网络基础

## HTTP1.0和HTTP1.1的区别；

<https://blog.csdn.net/forgotaboutgirl/article/details/6936982>

<https://www.cnblogs.com/bass6/p/5945014.html>

## DHCP如何实现分配IP的；

<https://blog.csdn.net/qq_35370263/article/details/79506676>

发现阶段（DHCP客户端在网络中广播发送DHCP DISCOVER请求报文，发现DHCP服务器，请求IP地址租约）、提供阶段（DHCP服务器通过DHCP OFFER报文向DHCP客户端提供IP地址预分配）、选择阶段（DHCP客户端通过DHCP REQUEST报文确认选择第一个DHCP服务器为它提供IP地址自动分配服务）和确认阶段（被选择的DHCP服务器通过DHCP ACK报文把在DHCP OFFER报文中准备的IP地址租约给对应DHCP客户端）。

## OSI七层模型，每层都说下自己的理解和知道的，说的越多越好；

<https://blog.csdn.net/superjunjin/article/details/7841099>

<https://blog.csdn.net/superjunjin/article/details/7841099>

<https://blog.csdn.net/qq_32528231/article/details/57415350>

<https://blog.csdn.net/JeremyZJM/article/details/78184775>

# 框架相关

## Servlet如何保证单例模式,可不可以编程多例的哪？

<https://blog.csdn.net/hollboy/article/details/7786805>

<https://blog.csdn.net/abrazen_zz/article/details/80543943>

## Dubbo请求流程以及原理；

## Spring框架如何实现事务的；

<https://www.cnblogs.com/WJ-163/p/6035462.html>

<https://blog.csdn.net/zhuxinquan61/article/details/71075051>

<https://blog.csdn.net/trigl/article/details/50968079>

## 如果一个接⼝有2个不同的实现, 那么怎么来Autowire一个指定的实现？

(可以使用Qualifier注解限定要注入的Bean，也可以使用Qualifier和Autowire注解指定要获取的bean，也可以使用Resource注解的name属性指定要获取的Bean)

## Spring框架中需要引用哪些jar包，以及这些jar包的用途；

<https://www.cnblogs.com/BensonHe/p/3903050.html>

## Spring Boot没有放到web容器⾥为什么能跑HTTP服务？

<https://blog.csdn.net/zhangxl0113/article/details/52672452>

<https://blog.csdn.net/liuchuanhong1/article/details/76667343?utm_source=gold_browser_extension>

<https://blog.csdn.net/liaokailin/article/details/48948093>

<https://blog.csdn.net/github_35169934/article/details/78573971>

## Spring中循环注入是什么意思，可不可以解决，如何解决；

<https://blog.csdn.net/chengxu2011/article/details/8478290>

<https://blog.csdn.net/icarus_wang/article/details/51588337>

<https://blog.csdn.net/jijianshuai/article/details/78122738>

## Spring的声明式事务 @Transaction注解⼀般写在什么位置? 抛出了异常会⾃动回滚吗？有没有办法控制不触发回滚?

## MyBatis怎么防止SQL注入；

<https://blog.csdn.net/yizhenn/article/details/52384601>

<https://blog.csdn.net/u013417227/article/details/70597871>

## Tomcat本身的参数你⼀般会怎么调整？

## 了解哪几种序列化协议？如何选择合适的序列化协议；

<https://blog.csdn.net/junzixing1985/article/details/79599348>

<https://blog.csdn.net/baiye_xing/article/details/73249819>

## Redis渐进式rehash过程？

<https://www.jianshu.com/p/9c84856cd5c0>

<https://blog.csdn.net/baiye_xing/article/details/76088425>

<http://redisbook.com/preview/dict/incremental_rehashing.html>

<https://blog.csdn.net/qq_29108585/article/details/58614447>

## 比如我有个电商平台，做每日订单的异常检测，服务端代码应该写；