# 基础题

## 怎么解决Hash冲突；（开放地址法、链地址法、再哈希法、建立公共溢出区等）

<https://blog.csdn.net/qq_27093465/article/details/52269862>

## 写出一个必然会产生死锁的伪代码；

<https://blog.csdn.net/coslay/article/details/45593859>

## Spring IoC涉及到的设计模式；（工厂模式、单利模式。。）

## toString()方法什么情况下需要重写；

## 判断对象相等时，什么情况下只需要重写 equals()，什么情况下需要重写 equals(),hashcode()？

## Set内存放的元素为什么不可以重复，内部是如何保证和实现的？

## 如何保证分布式缓存的一致性(分布式缓存一致性hash算法?)？分布式session实现？

<https://www.cnblogs.com/study-everyday/p/7853145.html>

粘性Session是指将用户锁定到某一个服务器上

## Java 8流式迭代的好处？

<https://www.cnblogs.com/shenlanzhizun/p/6027042.html>

## 项目中用到的JDK的哪些特性？

## 说一下TreeMap的实现原理？红黑树的性质？红黑树遍历方式有哪些？如果key冲突如何解决？setColor()方法在什么时候用？什么时候会进行旋转和颜色转换？

## Spring的bean的创建时机？依赖注入的时机？

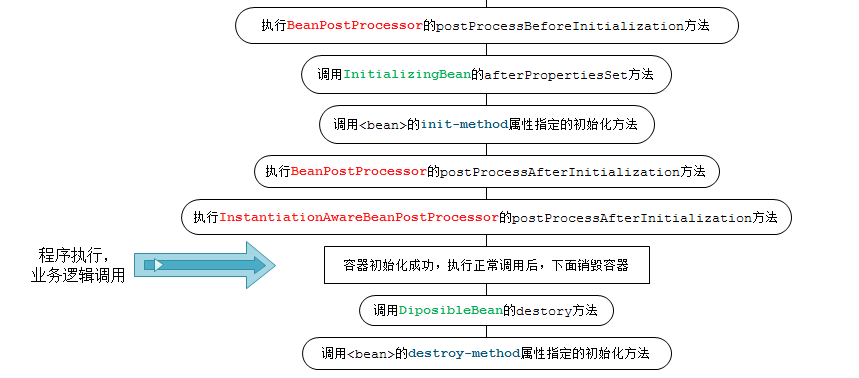
<https://blog.csdn.net/localhost01/article/details/51556423>

<https://www.cnblogs.com/java-class/p/4727775.html>

<https://blog.csdn.net/lisongjia123/article/details/52134396>

<https://www.cnblogs.com/zrtqsk/p/3735273.html>





## ArrayList和LinkList的删除一个元素的时间复杂度；（ArrayList是O(N)，LinkList是O(1)）；

## CopyOnWriteArrayList是什么；

<https://blog.csdn.net/linsongbin1/article/details/54581787>

## 序列化和反序列化底层如何实现的（ObjectOutputStream 、ObjectInputStream、 readObject writeObject）；

## 如何调试多线程的程序；

<https://blog.csdn.net/maritimesun/article/details/7724855>

<https://www.cnblogs.com/rocedu/p/6371262.html>

JDB

## 一个线程连着调用start两次会出现什么情况？（由于状态只有就绪、阻塞、执行，状态是无法由执行转化为执行的，所以会报不合法的状态！）

## HashMap在什么时候时间复杂度是O（1），什么时候是O（n），什么时候又是O（logn）；

## wait方法能不能被重写？（wait是final类型的，不可以被重写，不仅如此，notify和notifyall都是final类型的），wait能不能被中断；

## 一个Controller调用两个Service，这两Service又都分别调用两个Dao，问其中用到了几个数据库连接池的连接？

<https://blog.csdn.net/luhao/article/details/2415513>

<http://developer.51cto.com/art/200906/132336.htm>

# 网络基础

## HTTP、TCP、UDP的区别和联系；

<https://www.cnblogs.com/chengzhengfu/p/4584510.html>

## TCP和UDP各自的优势，知道哪些使用UDP协议的成功案例；

## TCP和UDP各用了底层什么协议；

<https://www.jianshu.com/p/43a25804b2e8>

## 单个UDP报文最大容量；

<https://blog.csdn.net/zdl1016/article/details/8673377>

65535-UDP包头8-IP包头20 = 65507

## 单个TCP报文最大容量；

<https://blog.csdn.net/qq_30667875/article/details/71216281>

以太网Ethernet最大的数据帧是1518字节。以太网帧的帧头14字节和帧尾CRC校验4字节（共占18字节），剩下承载上层协议的地方也就是Data域最大就只剩1500字节. 这个值我们就把它称之为MTU。

MSS=MTU-20字节TCP报头-20字节IP报头，那么在以太网环境下，MSS值一般就是1500-20-20=1460字节。

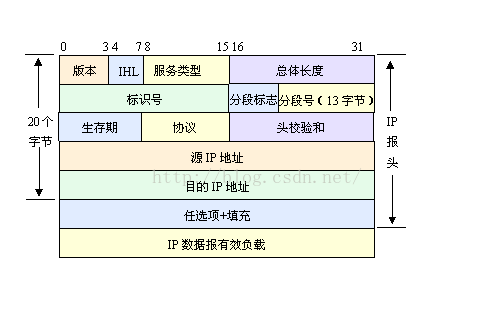
## TCP报头格式、UDP报头格式；

<https://blog.csdn.net/fjhfjhfjh123/article/details/52014978>

<https://blog.csdn.net/lqglqglqg/article/details/48850029>

<https://www.cnblogs.com/Allen-rg/p/7190042.html>

IP



UDP



TCP



## Server遭遇SYN Flood应当怎么处理；

<https://www.cnblogs.com/qiaoconglovelife/p/5713661.html>

3次握手中，服务器大量的半开连接。

## Web开发中如何防范XSS？

## 拆包和粘包的问题，如何解决，如果我们的包没有固定长度的话，我们的应用程序应该如何解决；

# 操作系统

## 为什么要内存对齐；

<https://blog.csdn.net/l_tudou/article/details/51999765>

内存对齐的主要作用：

1、平台原因(移植原因)

A 不是所有的硬件平台都能访问任意地址上的任意数据的；

B 某些硬件平台只能在某些地址处取某些特定类型的数据，否则抛出硬件异常。

2、性能原因：

A 数据结构(尤其是栈)应该尽可能地在自然边界上对齐。

B 原因在于，为了访问未对齐的内存，处理器需要作两次内存访问；而对齐的内存访问仅需要一次访问。

## 为什么会有大端小端，htol这一类函数的作用；

<https://blog.csdn.net/lolita0917/article/details/45770919>

<https://blog.csdn.net/a1414345/article/details/52156766>

<https://www.cnblogs.com/52php/p/6114080.html>

大端模式：是指数据的高字节保存在内存的低地址中，而数据的低字节保存在内存的高地址端。

小端模式，是指数据的高字节保存在内存的高地址中，低位字节保存在在内存的低地址端。

网络序：大端模式

## top显示出来的系统信息都是什么含义；（重要！）

<https://blog.csdn.net/csdn066/article/details/77171018>

<https://blog.csdn.net/lgstudyvc/article/details/53421161>

## Linux地址空间，怎么样进行寻址的；

<http://blog.chinaunix.net/uid-27717694-id-3799538.html>

<https://www.cnblogs.com/wuchanming/p/4339770.html>

## Linux如何查找目录或者文件的；

<https://www.cnblogs.com/wangkongming/p/3588429.html>

<http://www.jb51.net/LINUXjishu/418949.html>

<https://blog.csdn.net/adparking/article/details/38702293>

Find

# 分布式其他

## 分库与分表带来的分布式困境与应对之策；

<https://blog.csdn.net/jiangpingjiangping/article/details/78069480>

## Solr如何实现全天24小时索引更新；

<https://www.cnblogs.com/Zhong-Xin/p/5500599.html>

<https://blog.csdn.net/a7564951/article/details/50372894>

全量

增量

# Redis

## Redis插槽的分配（key的有效部分使用CRC16算法计算出哈希值，再将哈希值对16384取余，得到插槽值）;

<https://blog.csdn.net/it_hejinrong/article/details/79205528>

<https://www.cnblogs.com/wxd0108/p/5729754.html>

很显然，我们需要维护节点和槽之间的映射关系，每个节点需要知道自己有哪些槽，并且需要在结点之间传递这个消息。

为了节省存储空间，每个节点用一个Bitmap来存放其对应的槽:

2k = 2\*1024\*8 = 16384，也就是说，每个结点用2k的内存空间，总共16384个比特位，就可以存储该结点对应了哪些槽。然后这2k的信息，通过Gossip协议，在结点之间传递。

## Redis主从是怎么选取的（一种是主动切换，另一种是使用sentinel自动方式）;

<https://blog.csdn.net/u013516966/article/details/50633925>

## Redis复制的过程;

<https://www.cnblogs.com/yawen/p/7061126.html>

<https://blog.csdn.net/niugang0920/article/details/79737846>

<https://blog.csdn.net/daiyudong2020/article/details/51541557>

<https://www.cnblogs.com/mxf97826/p/8687664.html>

<https://blog.csdn.net/men_wen/article/details/72590550>

## Redis队列应用场景；

<https://www.cnblogs.com/leo_wl/p/3831349.html>

<https://segmentfault.com/q/1010000004486105>

## Redis主节点宕机了怎么办，还有没有同步的数据怎么办;

<https://blog.csdn.net/ligh_sqh/article/details/79383667>

<https://blog.csdn.net/wwd0501/article/details/51307785>

<https://www.cnblogs.com/xuegang/archive/2012/02/21/2361524.html>

<https://blog.csdn.net/sm9sun/article/details/74573792>

# 系统设计开放性题目

## 秒杀系统设计，超卖怎么搞;

<https://www.cnblogs.com/snailding/p/4947803.html>

<https://blog.csdn.net/limitless1113/article/details/46494391>

<https://www.cnblogs.com/Leo_wl/p/4997645.html>

<https://www.jianshu.com/p/c5f94afa57fb>

<https://www.jianshu.com/p/4fefa70464a8?utm_campaign=maleskine&utm_content=note&utm_medium=seo_notes&utm_source=recommendation>

Update table x = x - 1 where (x - 1) >= 0

独立部署

## 你们的图片时怎么存储的，对应在数据库中时如何保存图片的信息的？

<https://blog.csdn.net/haluoluo211/article/details/52080325>

## 假如成都没有一座消防站，现在问你要建立几座消防站，每个消防站要配多少名消防官兵，多少辆消防车，请你拿出一个方案；

## 基于数组实现一个循环阻塞队列；

<https://blog.csdn.net/ditto_zhou/article/details/77330733>

<https://www.cnblogs.com/dolphin0520/p/3932906.html>

## 常见的ipv4地址的展现形式如“168.0.0.1”，请实现ip地址和int类型的相互转换。（使用位移的方式）

<https://www.cnblogs.com/fogcell/p/6681855.html>

## 现网某个服务部署在多台Liunx服务器上，其中一台突然出现CPU 100%的情况，而其他服务器正常，请列举可能导致这种情况发生的原因？如果您遇到这样的情况，应如何定位？内存？CPU？发布？debug？请求量？

<https://blog.csdn.net/qq_20545159/article/details/77775122>

<https://blog.csdn.net/qinshi501/article/details/77442770>

<https://segmentfault.com/q/1010000002741770>

# 大数据量问题

## 给定a、b两个文件，各存放50亿个url，每个url各占64字节，内存限制是4G，让你找出a、b文件共同的url？

<https://blog.csdn.net/zhangtian6691844/article/details/51120772>

<https://www.cnblogs.com/studynote/articles/4920812.html>

<https://diducoder.com/mass-data-topic-6-multi-dividing.html>

<https://blog.csdn.net/caihong0571/article/details/8168536>

<https://blog.csdn.net/fucangzxx/article/details/51576628>

<https://blog.csdn.net/sunmenggmail/article/details/8117610>

Bloom Filter

Hash

Bit-Map

堆(Heap)

双层桶划分

数据库索引

倒排索引（Inverted Index）

外排序

Trie树/Radix Tree

MapReduce

## 海量日志数据，提取出某日访问百度次数最多的那个IP；

<https://blog.csdn.net/cnnumen/article/details/5781226>

<https://blog.csdn.net/tayanxunhua/article/details/20528389>

<https://blog.csdn.net/zhanglei0107/article/details/12137279>

<https://www.cnblogs.com/rav009/p/5131112.html>

## 一个文本文件，大约有一万行，每行一个词，要求统计出其中最频繁出现的前10个词，请给出思想，给出时间复杂度分析。

<https://blog.csdn.net/gamesofsailing/article/details/18040583>

<https://blog.csdn.net/u012328476/article/details/52381891>

<https://www.cnblogs.com/yang91/p/4803154.html>

<https://blog.csdn.net/u012289441/article/details/45192775>

此话题后边会有专门的文章探讨，如果有等不及的小伙伴，可以移步参考：

1、https://blog.csdn.net/v\_july\_v/article/details/6279498

2、https://blog.csdn.net/v\_july\_v/article/details/7382693

# 逻辑思维题

## 有两根粗细均匀的香（烧香拜佛的香），每一根烧完都花一个小时，怎么样能够得到15min？

## 假定你有8个撞球，其中有1个球比其他的球稍重,如果只能利用天平来断定哪一个球重,要找到较重的球,要称几次?（2次）；

## 实验室里有1000个一模一样的瓶子，但是其中的一瓶有毒。可以用实验室的小白鼠来测试哪一瓶是毒药。如果小白鼠喝掉毒药的话，会在一个星期的时候死去，其他瓶子里的药水没有任何副作用。请问最少用多少只小白鼠可以在一个星期以内查出哪瓶是毒药；（答案是10只）

<https://blog.csdn.net/zhu418766417/article/details/47256121>

<https://blog.csdn.net/a130737/article/details/44726395>

<https://blog.csdn.net/qianlizhima1/article/details/10267901>

<https://www.cnblogs.com/xiaoka/archive/2012/10/30/2747160.html>

bitmap

## 假设有一个池塘，里面有无穷多的水。现有2个空水壶，容积分别为5升和6升。问题是如何只用这2个水壶从池塘里取得3升的水；

<https://zhidao.baidu.com/question/715643857703492325.html>