# 知识点大全

# 链接-常用知识点：

https://m.baidu.com/?ref=www\_ptl&ms=1&from=1013377a&rsv\_pq=9662766133084091005&rsv\_t=7b9eaYh31HCBtniYN3RpQSBx80cGx8LNpKnUIT63u9D6yECkjSwenP7LdVO3ncw#iact=wiseindex%2Ftabs%2Fnews%2Factivity%2Fnewsdetail%3D%257B%2522linkData%2522%253A%257B%2522name%2522%253A%2522iframe%252Fmib-iframe%2522%252C%2522id%2522%253A%2522feed%2522%252C%2522index%2522%253A0%252C%2522url%2522%253A%2522https%253A%252F%252Fmbd.baidu.com%252Fnewspage%252Fdata%252Flandingpage%253Fs\_type%253Dnews%2526dsp%253Dwise%2526context%253D%25257B%252522nid%252522%25253A%252522news\_8859487327701719988%252522%25257D%2526pageType%253D1%2526n\_type%253D1%2526p\_from%253D-1%2526innerIframe%253D1%2522%252C%2522isThird%2522%253Afalse%252C%2522title%2522%253Anull%257D%257D

1 redis集群（主从复制是怎么触发的，数据如何分配）

2 mysql分库分表（分库有什么好处）

3 安全方面

4 加密算法

5 ca证书

6 sql注入

7 还用过哪些缓存框架

8 用过哪些签名和加密算法

9 Syncronized reentrantLock区别；

10 索引字段的建立；性别 加索引和不加索引的区别；

11 慢查询如何优化；

12 jvm如何进行优化，怎么去设置各个参数；垃圾回收算法及垃圾回收期如何选型及优化；

13 HashMap和ConcurrentHashMap的区别，链表查询和红黑树查询的复杂度是多少？红黑树是如何查询的？

1 session与cookie的区别及原理

2 NIO的原理及使用场景

3 springmvc原理和流程

4 在装用tomcat,java的linux环境下如何解决oom及过多load的问题

5 多线程中的ABA问题如何解决

6 工作中用到的数据库以及数据库优化的常用措施

7 如何设计一个高并发高可用架构?和如何设计秒杀系统?二选一下

1 多线程线程安全问题如何解决?

2 spring框架中的缓存体系工作有没有用到,如何使用的?

3 设计一个异构系统首先要考虑的问题是什么问题?

4 如何保证分布式事务的一致性?

5 工作中是否用到过分库分表,如何设计的?

1. Java内存结构，spring的aop的实现方法，java数据库问题定位和性能调优；

2. 关于Java异常的续承层次结构，讲述异常的续承关系；

3. java中重载和重写有什么区别，分别用什么关键字；

4. 关于分布式消息队列，分布式缓存；

5. 关于hashmap源码实现， jdk

6. 关于设计模式，uml，jvm 内存回收机制问题

7. java线程如何启动？java中加锁的方式有哪些，怎么个写法？

8. 对乐观锁和悲观锁的理解；

9. ORACLE中的SQL如何进行优化，都有哪些方式？事务有哪些特性，在ORACLE中隔离有哪些级别？

10. 介绍一下自己最近做的一个典型的项目；

11. 在项目中遇到了哪些问题，自己是如何解决的 ；

12. 目前系统支撑的用户量是多少，假如用户量提升10倍，系统会出现什么样的问题，如何重新设计系统【这里主要是想了解您的问题预见能力以及问题解决能力，考查思路】

13. 使用memcached是一个什么样的原理

14. 如何存放数据到memcached集群中，介绍一下这个过程。跟进的问题，讲一下一致性哈希算法的实现原理。

15. JVM中堆是如何管理的，JVM的内存回收机制，介绍一下

16. 分布式事务实现方式

17. 热点账户问题(项目中有就会问)

1 状态模式和策略模式的区别:

2 netty怎么设置写的比较快读得比较慢的问题

3 redis读取数据哪些情况会出现超时

4 一致性hash

5 读写分离读从库怎么写注解

6 dubbo客户端如何去调服务端

7内部类的为什么只会加载一次

8 MySQL读库注解

9 threadExecutor四中拒绝策略.

1.Core Java，就是Java基础、JDK的类库，很多童鞋都会说，JDK我懂，但是懂还不足够，知其然还要知其所以然，JDK的源代码写的非常好，要经常查看，对使用频繁的类，比如String，集合类（List，Map，Set）等数据结构要知道它们的实现，不同的集合类有什么区别，然后才能知道在一个具体的场合下使用哪个集合类更适合、更高效，这些内容直接看源代码就OK了

2.多线程并发编程，现在并发几乎是写服务端程序必须的技术，那对Java中的多线程就要有足够的熟悉，包括对象锁机制、synchronized关键字，concurrent包都要非常熟悉，这部分推荐你看看[《Java并发编程实践》](https://www.baidu.com/s?wd=%E3%80%8AJava%E5%B9%B6%E5%8F%91%E7%BC%96%E7%A8%8B%E5%AE%9E%E8%B7%B5%E3%80%8B&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YduHTduHPhmHRYuHbzuH9W0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnWfsPWb1nHDsnjnYnHnknj6Y" \t "_blank)这本书，讲解的很详细

3.I/O，Socket编程，首先要熟悉Java中Socket编程，以及I/O包，再深入下去就是Java NIO,再深入下去是[操作系统](https://www.baidu.com/s?wd=%E6%93%8D%E4%BD%9C%E7%B3%BB%E7%BB%9F&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YduHTduHPhmHRYuHbzuH9W0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnWfsPWb1nHDsnjnYnHnknj6Y" \t "_blank)底层的Socket实现,了解Windows和Linux中是怎么实现socket的

4.JVM的一些知识，不需要熟悉，但是需要了解，这是Java的本质，可以说是Java的母体, 了解之后眼界会更宽阔，比如Java内存模型（会对理解Java锁、多线程有帮助）、字节码、JVM的模型、各种垃圾收集器以及选择、JVM的执行参数（优化JVM）等等,这些知识在[《深入Java虚拟机》](https://www.baidu.com/s?wd=%E3%80%8A%E6%B7%B1%E5%85%A5Java%E8%99%9A%E6%8B%9F%E6%9C%BA%E3%80%8B&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YduHTduHPhmHRYuHbzuH9W0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnWfsPWb1nHDsnjnYnHnknj6Y" \t "_blank)这本书中都有详尽的解释,或者去oracle网站上查看具体版本的JVM规范.

5.一些常用的设计模式，比如单例、模板方法、代理、适配器等等，以及在Core Java和一些[Java框架](https://www.baidu.com/s?wd=Java%E6%A1%86%E6%9E%B6&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YduHTduHPhmHRYuHbzuH9W0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnWfsPWb1nHDsnjnYnHnknj6Y" \t "_blank)里的具体场景的实现，这个可能需要慢慢积累，先了解有哪些使用场景，见得多了，自己就自然而然会去用。

6.常用数据库（Oracle、MySQL等）、[SQL语句](https://www.baidu.com/s?wd=SQL%E8%AF%AD%E5%8F%A5&tn=44039180_cpr&fenlei=mv6quAkxTZn0IZRqIHckPjm4nH00T1YduHTduHPhmHRYuHbzuH9W0ZwV5Hcvrjm3rH6sPfKWUMw85HfYnjn4nH6sgvPsT6KdThsqpZwYTjCEQLGCpyw9Uz4Bmy-bIi4WUvYETgN-TLwGUv3EnWfsPWb1nHDsnjnYnHnknj6Y" \t "_blank)以及一般的优化

7.JavaWeb开发的框架，比如Spring、iBatis等框架，同样他们的原理才是最重要的，至少要知道他们的大致原理。

8.其他一些有名的用的比较多的开源框架和包，Netty网络框架，Apache common的N多包，Google的Guava等等，也可以经常去Github上找一些代码看看。

1、Java基础技术体系、JVM内存分配、垃圾回收、类装载机制、性能优化、反射机制、多线程、网络编程、常用数据结构和相关算法。

2、对面向对象的软件开发思想有清晰的认识、熟悉掌握常用的设计模式。

3、熟练掌握目前流行开源框架(spring/springmvc/ibatis)，并且对其核心思想、实现原理有一定认知。

4、熟悉Oracle、MySQL等数据库开发与设计以及缓存系统Redis或 Memcached的设计和研发。

5、熟悉底层中间件、分布式技术（包括缓存、消息系统、热部署）、消息中间件：ActiveMQ、RabbitMQ、工作流中间件:JBPM。

6、精通shell编程，熟练应用awk、sed、grep、strace、tcudump、gdb等常用命令。

7、有大型分布式、高并发、高负载（大数据量）、高可用性系统设计开发经验。

8、对配置管理和敏捷研发模式有所了解（svn,git）。

9、熟悉常见的一些解决方案及其原理：单点登录、分布式缓存、SOA、全文检索、消息中间件，负载均衡、连接池、流计算等。

10、能大概知道市面上主流技术的特点及业务瓶颈。

1

**常用设计模式**

* 单例模式：懒汉式、饿汉式、双重校验锁、静态加载，内部类加载、枚举类加载。保证一个类仅有一个实例，并提供一个访问它的全局访问点。
* 代理模式：动态代理和静态代理，什么时候使用动态代理。
* 适配器模式：将一个类的接口转换成客户希望的另外一个接口。适配器模式使得原本由于接口不兼容而不能一起工作的那些类可以一起工作。
* 装饰者模式：动态给类加功能。
* 观察者模式：有时被称作发布/订阅模式，观察者模式定义了一种一对多的依赖关系，让多个观察者对象同时监听某一个主题对象。这个主题对象在状态发生变化时，会通知所有观察者对象，使它们能够自动更新自己。
* 策略模式：定义一系列的算法,把它们一个个封装起来, 并且使它们可相互替换。
* 外观模式：为子系统中的一组接口提供一个一致的界面，外观模式定义了一个高层接口，这个接口使得这一子系统更加容易使用。
* 命令模式：将一个请求封装成一个对象，从而使您可以用不同的请求对客户进行参数化。
* 创建者模式：将一个复杂的构建与其表示相分离，使得同样的构建过程可以创建不同的表示。
* 抽象工厂模式：提供一个创建一系列相关或相互依赖对象的接口，而无需指定它们具体的类。

2

**基础知识**

* ava基本类型哪些，所占字节和范围
* Set、List、Map的区别和联系
* 什么时候使用Hashmap
* 什么时候使用Linkedhashmap、Concurrenthashmap、Weakhashmap
* 哪些集合类是线程安全的
* 为什么Set、List、map不实现Cloneable和Serializable接口
* Concurrenthashmap的实现，1.7和1.8的实现
* Arrays.sort的实现
* 什么时候使用CopyOnArrayList
* volatile的使用
* synchronied的使用
* reentrantlock的实现和Synchronied的区别
* CAS的实现原理以及问题
* AQS的实现原理
* 接口和抽象类的区别，什么时候使用
* 类加载机制的步骤，每一步做了什么，static和final修改的成员变量的加载时机
* 双亲委派模型
* 反射机制：反射动态擦除泛型、反射动态调用方法等
* 动态绑定：父类引用指向子类对象
* JVM内存管理机制：有哪些区域，每个区域做了什么
* JVM垃圾回收机制：垃圾回收算法 垃圾回收器 垃圾回收策略
* jvm参数的设置和jvm调优
* 什么情况产生年轻代内存溢出、什么情况产生年老代内存溢出
* 内部类：静态内部类和匿名内部类的使用和区别
* Redis和memcached：什么时候选择redis，什么时候选择memcached，内存模型和存储策略是什么样的
* MySQL的基本操作 主从数据库一致性维护
* mysql的优化策略有哪些
* mysql索引的实现 B+树的实现原理
* 什么情况索引不会命中，会造成全表扫描
* java中bio nio aio的区别和联系
* 为什么bio是阻塞的 nio是非阻塞的 nio是模型是什么样的
* Java io的整体架构和使用的设计模式
* Reactor模型和Proactor模型
* http请求报文结构和内容
* http三次握手和四次挥手
* rpc相关：如何设计一个rpc框架，从io模型 传输协议 序列化方式综合考虑
* Linux命令 统计，排序，前几问题等
* StringBuff 和StringBuilder的实现，底层实现是通过byte数据，外加数组的拷贝来实现的
* cas操作的使用
* 内存缓存和数据库的一致性同步实现
* 微服务的优缺点
* 线程池的参数问题
* ip问题 如何判断ip是否在多个ip段中
* 判断数组两个中任意两个数之和是否为给定的值
* 乐观锁和悲观锁的实现
* synchronized实现原理
* 你在项目中遇到的困难和怎么解决的
* 你在项目中完成的比较出色的亮点
* 消息队列广播模式和发布/订阅模式的区别
* 生产者消费者代码实现
* 死锁代码实现
* 线程池：参数，每个参数的作用，几种不同线程池的比较，阻塞队列的使用，拒绝策略
* Future和ListenableFuture 异步回调相关
* 算法相关：判断能否从数组中找出两个数字和为给定值，随机生成1~10000不重复并放入数组，求数组的子数组的最大和，二分查找算法的实现及其时间复杂计算

3

**其他**

* 算法：常用排序算法，二分查找，链表相关，数组相关，字符串相关，树相关等
* 常见序列化协议及其优缺点
* memcached内存原理，为什么是基于块的存储
* 搭建一个rpc需要准备什么
* 如果线上服务器频繁地出现full gc ，如何去排查
* 如果某一时刻线上机器突然量变得很大，服务扛不住了，怎么解决
* LUR算法的实现
* LinkedHashMap实现LRU
* 定义栈的数据结构，请在该类型中实现一个能够找到栈最小元素的min函数
* 海量数据处理的解决思路
* reactor模型的演变
* 阻塞、非阻塞、同步、异步区别
* Collection的子接口
* jvm调优相关
* zookeeper相关，节点类型，如何实现服务发现和服务注册
* nginx负载均衡相关，让你去实现负载均衡，该怎么实现
* linux命令，awk、cat、sort、cut、grep、uniq、wc、top等
* 压力测试相关，怎么分析，单接口压测和多情况下的压测
* 你觉得你的有点是什么，你的缺点是什么
* spring mvc的实现原理
* netty底层实现，IO模型，ChannelPipeline的实现和原理
* 缓存的设计和优化
* 缓存和数据库一致性同步解决方案
* 你所在项目的系统架构，谈谈整体实现
* 消息队列的使用场景
* ActiveMQ、RabbitMQ、Kafka的区别

常用框架;

Spring 、mybatis、dubbo等

常用功能

Java基础、消息队列、docker、分布式、Linux、并发、大数据、数据库优化、缓存、jvm、常用工具[性能工具、测试工具]、设计模式