

一面:

- HashMap 实现原理, ConcurrentHashMap 实现原理
- 红黑树, 为什么允许局部不平衡
- TCP, UDP 区别, 为什么可靠和不可靠
- 一次 HTTP 请求的全过程, 包括域名解析、定位主机等
- TCP 三次握手
- MySQL 事务是什么? 四大特性, 四大隔离级别
- ConcurrentHashMap 和 Hashtable 区别
- spring IOC 和 AOP, 以及各有什么优点
- 有哪几种常用的线程池
- 什么情况下使用 Runnable 和 Thread 创建线程, Runnable 和 Callable 的区别
- 线程方法中的异常如何处理, 副线程可以捕获到吗
- synchronized 和锁的区别, 什么情况下使用 synchronized 和 ReentrantLock
- JVM 的对象分配在哪个区, Class 对象分配在哪个区

二面:

- 常用的设计模式介绍: 单例模式、装饰者模式等
- Java 会出现内存溢出吗? 什么情况下会出现?
- 双亲委派模型, 为什么这样做?
- 对象什么情况下进入老年代?
- 快速排序说一下过程
- AOP 实现原理: 动态代理
- BIO、NIO (如何实现的)、AIO
- 消息中间件有哪些? 他们之间的优劣势?

- Redis, 持久化框架
- 栈和队列
- 垃圾回收算法
- MySQL 的索引
- Tomcat 类加载器
- OOM 内存泄漏, 什么情况下会出现, 如何排查

三面:

- 介绍你实践的性能优化案例, 以及你的优化思路
- 微服务和 SOA 的区别, 优劣势
- SQL 慢查询的优化方案, 索引和表的优化方案。
- MySQL 与 MongoDB 的区别, 海量数据的存储
- 缓存框架, 例如 Redis、Memcached 之间的区别, 优劣势比较
- 请描述一致 hash 算法
- 分布式 session 的共享方案有哪些, 有什么优劣势
- 高并发情况, 系统的优化方案有哪些, 以及优先级排序。

面试总结

阿里的面试特别喜欢面试技术原理, 特别是:

- 多线程
- NIO
- 异步消息框架
- 分布式相关的缓存算法等
- JVM 的加载过程和原理
- 回收算法

- 以及具体使用过的框架，会问部分参数检验你是否熟用
- 第一面被通过，后续被录用的可能性就会很大了，第一轮非常重要，建议系统的学习面试题