1、面试题

说一下的dubbo的工作原理？注册中心挂了可以继续通信吗？说说一次rpc请求的流程？

2、面试官心里分析

MQ、ES、Redis、Dubbo，上来先问你一些思考的问题，原理（kafka高可用架构原理、es分布式架构原理、redis线程模型原理、Dubbo工作原理），生产环境里可能会碰到的一些问题（每种技术引入之后生产环境都可能会碰到一些问题），系统设计（设计MQ，设计搜索引擎，设计一个缓存，设计rpc框架）

当然比如说，hard面试官，死扣，结合项目死扣细节，百度（深入底层，基础性），阿里（结合项目死扣细节，扣很深的技术底层），小米（数据结构和算法）。

那既然开始聊分布式系统了，自然重点先聊聊dubbo了，毕竟dubbo是目前事实上大部分公司的分布式系统的rpc框架标准，基于dubbo也可以构建一整套的微服务架构。但是需要自己大量开发。

当然去年开始spring cloud非常火，现在大量的公司开始转向spring cloud了，spring cloud人家毕竟是微服务架构的全家桶式的这么一个东西。但是因为很多公司还在用dubbo，所以dubbo肯定会是目前面试的重点，何况人家dubbo现在重启开源社区维护了，未来应该也还是有一定市场和地位的。

既然聊dubbo，那肯定是先从dubbo原理开始聊了，你先说说dubbo支撑rpc分布式调用的架构师啥，然后说说一次rpc请求dubbo是怎么给你完成的，对吧。

3、面试题剖析

（1）dubbo工作原理

第一层：service层，接口层，给服务提供者和消费者来实现的

第二层：config层，配置层，主要是对dubbo进行各种配置的

第三层：proxy层，服务代理层，透明生成客户端的stub和服务单的skeleton

第四层：registry层，服务注册层，负责服务的注册与发现

第五层：cluster层，集群层，封装多个服务提供者的路由以及负载均衡，将多个实例组合成一个服务

第六层：monitor层，监控层，对rpc接口的调用次数和调用时间进行监控

第七层：protocol层，远程调用层，封装rpc调用

第八层：exchange层，信息交换层，封装请求响应模式，同步转异步

第九层：transport层，网络传输层，抽象mina和netty为统一接口

第十层：serialize层，数据序列化层

工作流程：

1）第一步，provider向注册中心去注册

2）第二步，consumer从注册中心订阅服务，注册中心会通知consumer注册好的服务

3）第三步，consumer调用provider

4）第四步，consumer和provider都异步的通知监控中心

（2）注册中心挂了可以继续通信吗？

可以，因为刚开始初始化的时候，消费者会将提供者的地址等信息拉取到本地缓存，所以注册中心挂了可以继续通信