# 1、如何获得高并发的经验

## 结论

1、首先得在自己电脑上鼓捣出 Redis，kafka，dubbo，mycat 等高并发分布式组件的环境，通过这些环境了解分布式组件的基本 api。

2、为这些分布式组件找个项目场景，证明自己用过，并且解决过实际问题。这里需要指出的是单纯背高并发的面试题没用。

3、修改简历，找到高并发面试的机会，然后通过多场面试，不断调整自己讲述高并发项目经验的水平。这时如果能直接用高并发的说辞进大厂，那是最好的，如果不能，找个过渡性的能提供高并发实践的岗位。

4、如果有高并发项目的实践经验，那么后面就不用说了，只要多解决实际问题，高并发技能一定能很快掌握。这时掌握的高并发技能，就不是单纯的理论说辞和简单的 API 技能了，而是正宗的高并发项目实践技能。

5、要得到高并发经验，一般得分两步走，第一通过面试得到实践机会，第二在项目中提升，其实难点在第一点，在零基础前提下，准备高并发方面的面试，确实难。而通过面试找到项目实践机会后，可能会被人鄙视，也会被领导认为很水，但好歹得到了能进一步提升的机会。

那在零基础前提下，准备高并发方面的面试的难点在哪里呢？

1、高并发的组件大多是运行在 linux 上，windows 很难搭建环境，没有环境其它就不用说了。

2、零基础的球友，不知道高并发场景下的分布式组件怎么用，或者只知道如何用简单的 api。由于缺乏项目经验，所以也不知道分布式组件在项目里是怎么用的。

3、在学分布式组件解决高并发问题方面，往往只会孤立地使用某个组件，比如孤立使用 redis 做缓存，孤立使用 rabbitmq 做消息中间件，但现实场景下，往往是综合使用多个分布式组件，协同解决高并发问题。

4、由于只会理论，没有在项目里接触过高并发开发，所以普遍缺乏分析排查解决高并发方面问题的经验。在高并发场景下，往往会用到集群，熔断，限流等。在使用高并发组件的时候，往往也会踩到不少坑，比如 mycat 方言不对，netty 半包，消息中间件重发等。

举例：

1、小二具备 Spring Boot+JPA 开发经验，这其实也是高并发经验的基础。

2、小二掌握了如何在 windows 环境下搭建 Redis,MyCAT,RabbitMQ，nginx，dubbo+zookeeper，MongoDB 等组件运行环境的步骤，请注意是 windows 环境，而不是 linux 环境。虽然生产环境中，这些组件都在 linux 环境下，因为大家平时都用 windows，而且这些组件在 windows 和 linux 环境下的开发差异不大。

3、环境搭建好以后，首先运行一通 spring boot 整合诸多分布式组件的项目，这样他就知道怎么用，而不会仅仅知道怎么说。**然后我就让他为每个组件，找个业务背景。**

**4、**会用+结合业务背景使用高并发分布式组件以后，**我再让小二背些分布式组件方面的常用坑，以及常规的使用经验**，无非是 dubbo 调用超时，redis 超时，redis 内存溢出，dubbo 整合 zookeeper 的要点等。这方面不要太深，不用涉及到算法，也不用涉及到底层技能。而且这方面内容我书里大多也提到。**最后我让小二通过秒杀案例，串起来讲多个分布式组件的使用要点。**

**5、**这个是要点，我让小二准备了不少解决高并发问题的说辞。高并发问题其实也很普遍，无非是 redis 超时，kakfa 由于重发而导致的不幂等，dubbo 注册中心失效等。这些方面，我让我小弟准备哪些点呢？问题是如何发现（无非是通过日志告警或系统卡顿发现），如何排查（到 linux 看日志，日志中看到哪些具体的关键字），如何解决（改文件或改配置）。