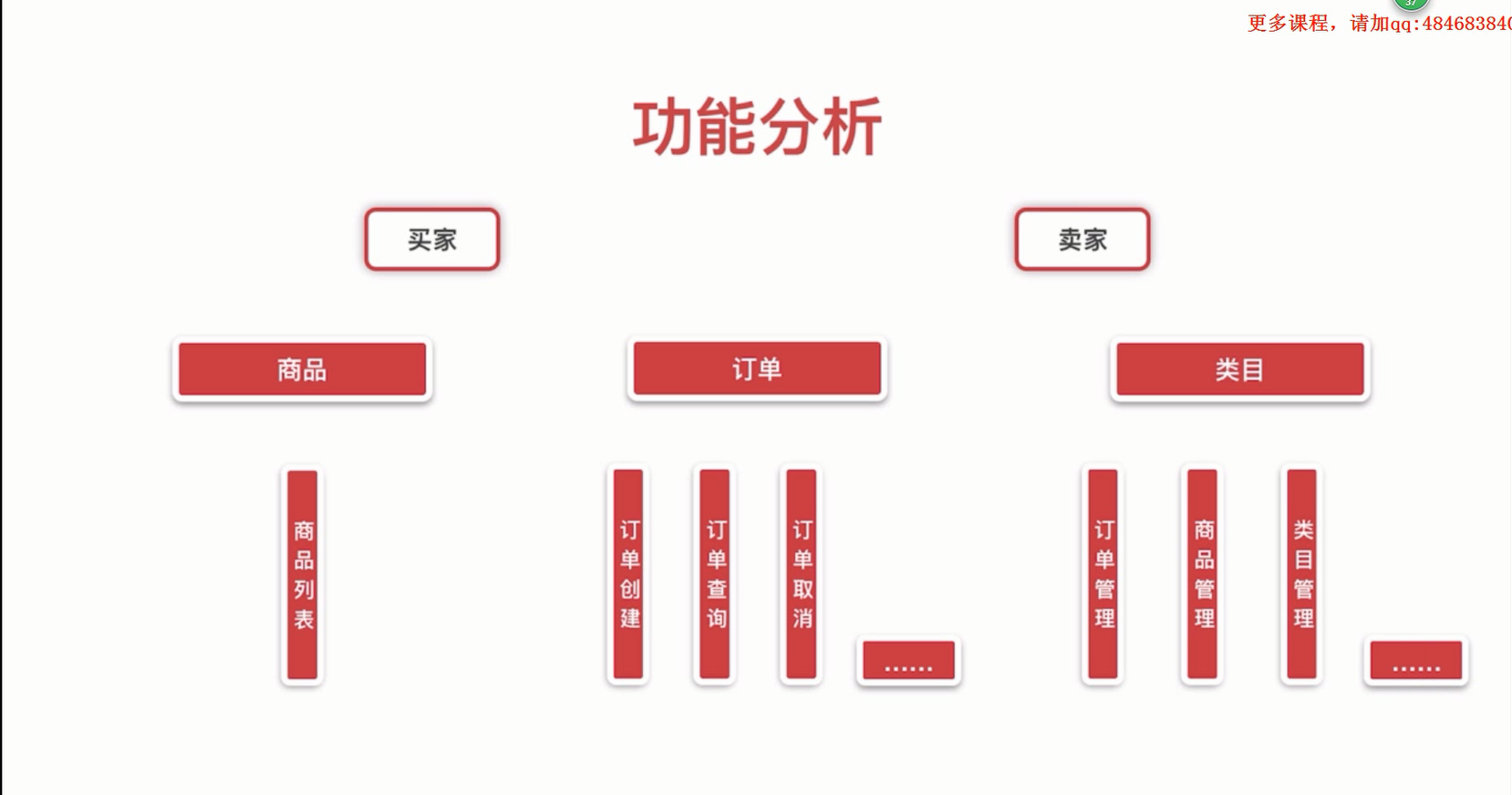
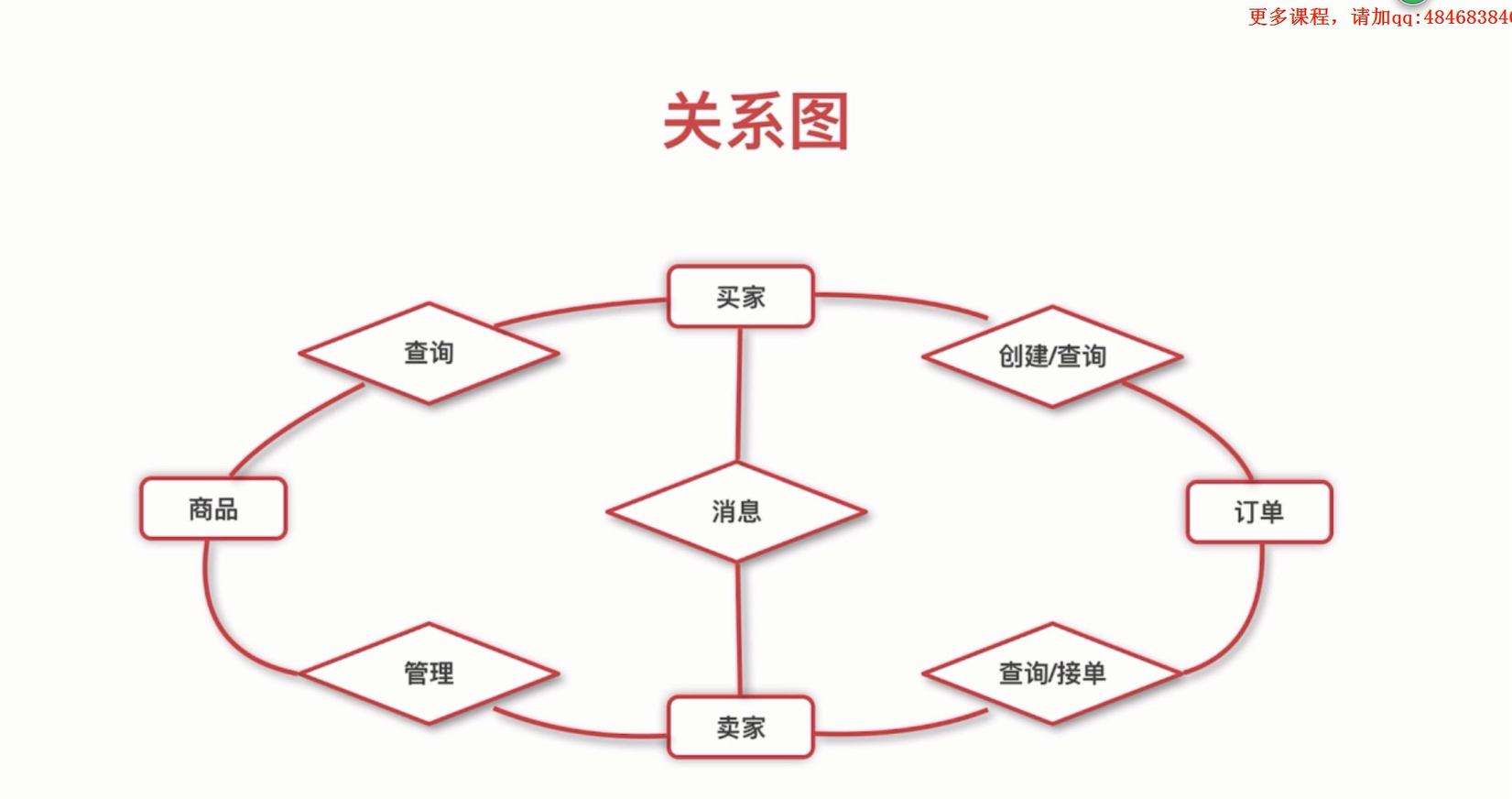


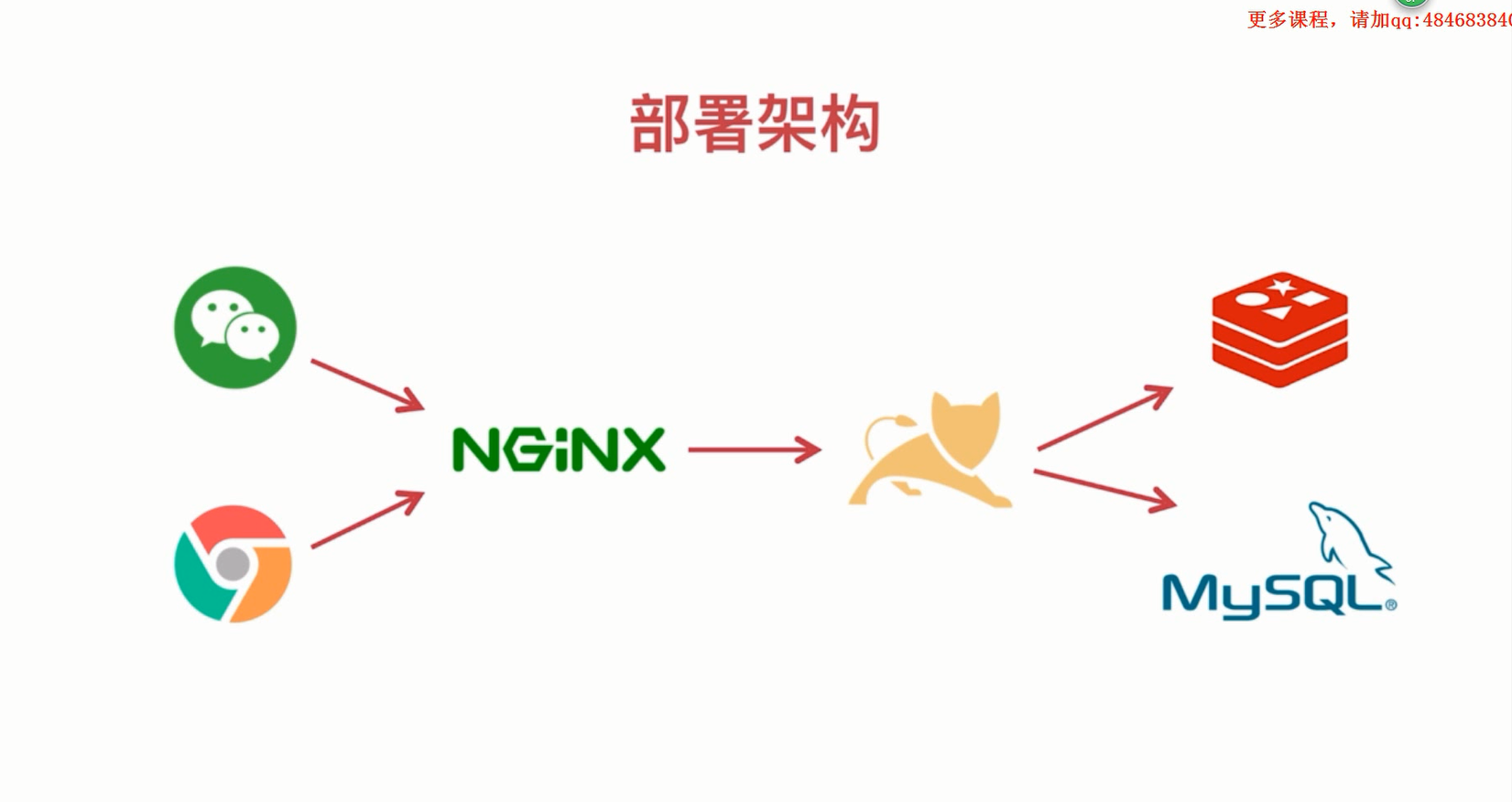


**角色划分：买家（手机端）- 微信公众号提供的服务 卖家（PC端）**

**功能模块：商品、订单、类目**







**nginx实现动态资源与静态资源的分离与负载均衡，应用支持分布式部署，图上的tomcat表示的是多个Tomcat上的多个应用。**



**Dubbo使用的服务治理服务之间的互相调用是如何实现的呢？**

**Spring Cloud的服务之间的相互调用可能是基于Eureka+SpringCloudRibbon或者SpringCloudFeign来实现的。**

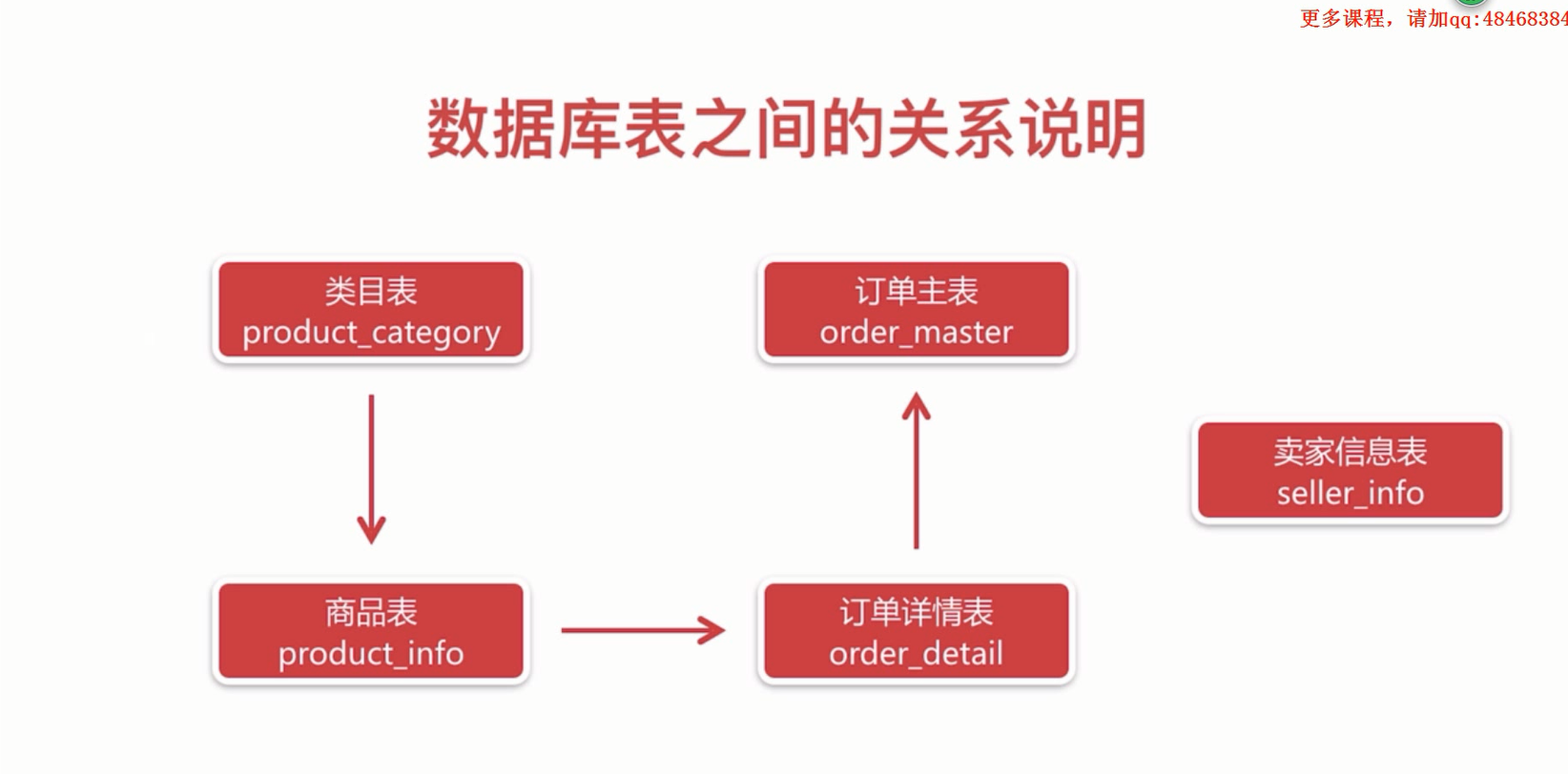
**SpringCloud之中的zuul组件在使用Dubbo做服务治理的时候是不是只能使用nginx来替代zuul的功能？？？**

**项目难点：微信扫码登录、微信模板消息推送、webSocket消息推送、微信支付与退款**

**千人千面推荐，千人十面余弦过滤，**人工×智能应该是现阶段更适合行业发展的内容分发方式。例子：新浪新闻的要闻模块：千人十面，新浪新闻的推荐模块：千人千面

要闻频道是基于新浪新闻在大事件报道、人工编辑把控以及智能推荐等方面的优势，以千人十面的算法向用户推荐的普适性重大新闻聚合区。进入要闻频道，排在最前面的版块是当天最重要的新闻以及专题报道，其他热点新闻则隐藏在“今日热点”下的滚动栏中，用户可以自主选择查看。排在热点新闻之后的则是由专业编辑从各个频道精选出来的各领域重要内容，算法根据用户兴趣将不同内容版块依次进行推荐。比如关注时政新闻的用户能优先看到时事板块，资深车迷则优先看到汽车版块，喜欢娱乐资讯的用户则优先看到娱乐版块。

我理解的千人十面：先由编辑精选出各个领域的重要新闻之后进入一个池子，用户可以分别查看每个领域内的重要新闻。每个模块内有根据用户兴趣的推荐操作，比如体育模块可能发现用户比较喜欢篮球性质的新闻，那体育模块里面给用户推送的篮球类新闻就多一些。当然每个模块内都包含一些普适性新闻，防止仅有算法推荐的信息推荐越来越闭塞，比如用户喜欢NBA，体育模块里面推送的篮球新闻会多一些，但不是只有篮球新闻。 简言之就是采用算法+人工干预的模式。



**一个类目下包含多个商品；一个订单主表中的记录对应多个订单详情表中的记录**



为什么选择SLF4j和Logback呢？SLF4j和Logback是同一个作者写的，这个作者还写了Log4j，作者说Log4j太烂了以至于他不想改了，所以写了Logback.

Jboss-logging与jul不通用，Log4j2有过度设计的嫌疑。

Commons-logging.jar为JCL接口，slf4j.jar为SLF4j接口，springboot对于这两种接口都提供了默认的支持；springboot使用的默认日志接口的实现是logback。

拥抱开源，使用别人做好的一些优秀的组件，不要重复造轮子！！！

Windows抓包工具：fiddler，调试时配置手机代理之后使用使用fiddler抓包

**分布式系统下使用Redis模拟实现的Session**

**分布式系统定义：**旨在支持应用程序和服务的开发，可以利用物理架构由多个自治的处理元素，不共享主内存，但通过网络发送消息合作。

**分布式系统三个特点：**多节点、消息通信、不共享内存

**容易混淆的三个概念：**分布式系统、集群、分布式计算

**分布式系统与集群的异同：**

**分布式系统：**厨房里有两个人，一个炒菜的，一个洗菜的，这就是分布式系统，强调的是系统中含有多个不同功能的模块，模块之间有消息通信；

**集群：**厨房里四个人，2个人炒菜，2个人洗菜，这就是一个分布式系统，但里面有两个集群，一个炒菜的集群，一个洗菜的集群，强调的是同一个集群中的节点业务功能都相同。

**分布式计算：Hadoop中的Map-Reduce；**

**水平扩展：采用集群的形式，集群中每个节点提供的都是应用的全部功能。**

**垂直扩展：拆分服务，每个节点负责一部分功能。**

**Redis Desktop Manager：redis管理工具，www.redis.cn**

**SessionID在分布式系统下的唯一性采用什么算法解决呢？**

**高并发压测模拟：Apache ab，LoadRunner等**

**Redis分布式锁**

**分布式锁设计目标：**

**1、可以保证在分布式部署的应用集群中，同一个方法在同一操作只能被一台机器上的一个线程执行。**

**2、这把锁要是一把可重入锁（避免死锁）**

**3、这把锁最好是一把阻塞锁（根据业务需求考虑要不要这条）**

**4、这把锁有高可用的获取锁和释放锁功能**

**5、这把锁获取锁和释放锁的性能要好…**

**Redis号称可以支持1s十几万的并发**

**秒杀活动 redis setnx, getset命令，因为redis是单线程的，所以可以利用这两个命令配合实现分布式锁(主要是setnx命令)-自己造轮子**

**还可以使用Redis的官方推荐java客户端redisson，里面有原生的分布式锁的实现，非常流弊。**

**分布式锁可以概括为多台机器上多个进程对一个数据进行操作的互斥，比如商品秒杀时的库存数据**

**参考：**

分布式锁的实现与应用场景对比**https://blog.csdn.net/lemon89/article/details/52796775**

**Redis缓存**

**缓存命中：@Cacheable()注解**

**缓存失效：@CacheEvict**

**缓存更新：@CachePut**

**使用注解使用Redis的方式是AOP原理实现的**

**打jar包部署应用的方式比较优秀，比较流行的是使用容器编排的方式。**