个人介绍：

课题	讲师
电商项目整体架构详解	monkey老师
电商项目微服务架构拆分及授权中心实战	fox老师
从零到一实战高性能网关服务及授权中心对接	
秒杀系统-商品详情多级缓存优化实战（上）	monkey老师
秒杀系统-商品详情多级缓存优化实战（中）	
秒杀系统-商品详情多级缓存优化实战（下）	
商品搜索实战（上）	白起老师
商品搜索实战（下）	
秒杀系统-订单交易全链路优化实战（一）	monkey老师
秒杀系统-订单交易全链路优化实战(二)	
秒杀系统-订单交易全链路优化实战（三）	楼兰老师
秒杀系统-订单交易全链路限流实战（四）	fox老师
订单交易系统分布式事务实战	
Sharding分库分表&分布式事务&分布式主键(上)	楼兰老师
Sharding分库分表&分布式事务&分布式主键(下)	
电商项目并发编程实战与总结	monkey老师
电商项目微服务中台架构落地实战	楼兰老师
电商项目领域驱动开发DDD探索及实战	
容器部署Docker快速实战	诸葛老师
电商项目Docker Compose整体部署实战	
Kubernetes架构与核心组件详解以及安装部署实战	
Kubernetes核心对象与组件操作快速实战	
Kubernetes微服务电商后台部署实战	
自动化监控系统Prometheus&Grafana实战	
电商EFK日志系统实战	白起老师
全链路压测	神秘嘉宾
电商项目JVM&Mysql&Tomcat线上整体优化	诸葛老师

图灵电商项目白皮书.pdf
1.41MB

程序 = 算法 +数据结构的话

系统 =? + ?

电商项目业务整体概览：

电子商务类型

按交易主体

类型	电商形式	代表企业
B2C	直接面向消费者销售产品和服务商业零售模式	mi HUAWEI vivo
B2B	企业与企业之间通过互联网开展交易活动的商业模式	1688.com 全球最大的采购批发市场
B2B2C	B2B、B2C模式的演变和完善，把B2C和B2B完美地结合起来，	天猫 TMALL.com JD.COM 京东 GOME
C2C	个人与个人之间的电子商务	淘宝网 Taobao.com 闲鱼
O2O (OnlineToOffline)	一些线下的商家让互联网成为线下交易的平台	美团
B2G	企业与政府机构间的电子商务	中国政府采购网

按运营模式：

按运营方向:

传统电商　　淘宝天猫、京东、唯品会、聚美优品、当当...

社交电商　　拼多多、京东、国美、苏宁

网红电商　　抖音、快手

内容电商　　头条三农领域

常见术语：

PV：　Page view，即网站被浏览的总次数

UV：　Unique Vister的缩写，独立访客

CR：　ConversionRate的缩写，是指访问某一网站访客中，转化的访客占全访客的比例（订单转化率=有效订单数/访客数）

SPU： Standard Product Unit （标准化产品单元）,SPU是商品信息聚合的最小单位，是一组可复用、易检索的标准化信息的集合，该集合描述了一个产品的特性。
SKU： Stock keeping unit(库存量单位) SKU即库存进出计量的单位（买家购买、商家进货、供应商备货、工厂生产都是依据SKU进行的），在服装、鞋类商品中使用最多最普遍。例如纺织品中一个SKU通常表示：规格、颜色、款式。SKU是物理上不可分割的最小存货单元。

关于咱们这个电商项目我们的思考是

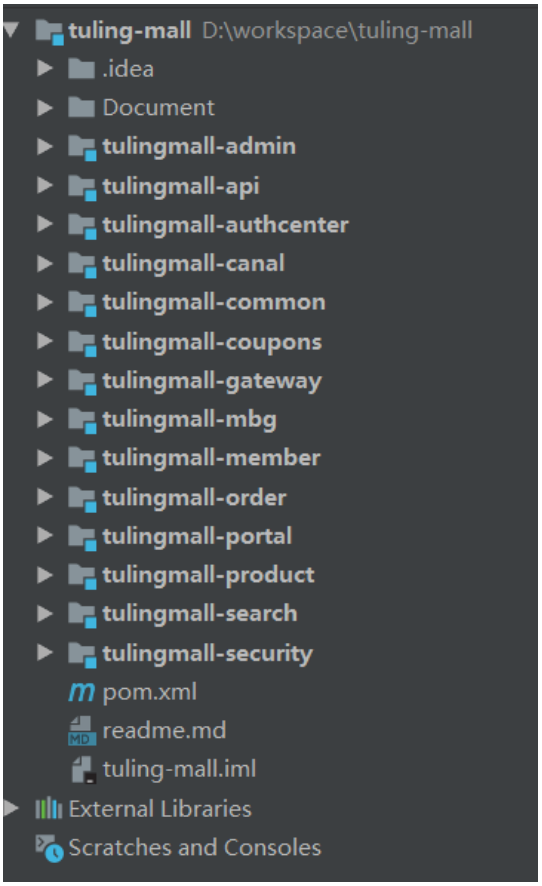
我们的思考：

容量规划、架构设计、数据库设计、缓存设计、框架选型、数据迁移方案、性能压测、监控报警、领域模型、回滚方案、高并发、分库分表

图灵电商介绍：

前端架构：
前台:tulingmall-front 后台：mall-amdin-web

后端架构：



项目架构图、技术选型、数据库介绍

数据库开头对应：

cms 后台模块

oms 订单模块

pms 商品模块

sms 营销模块

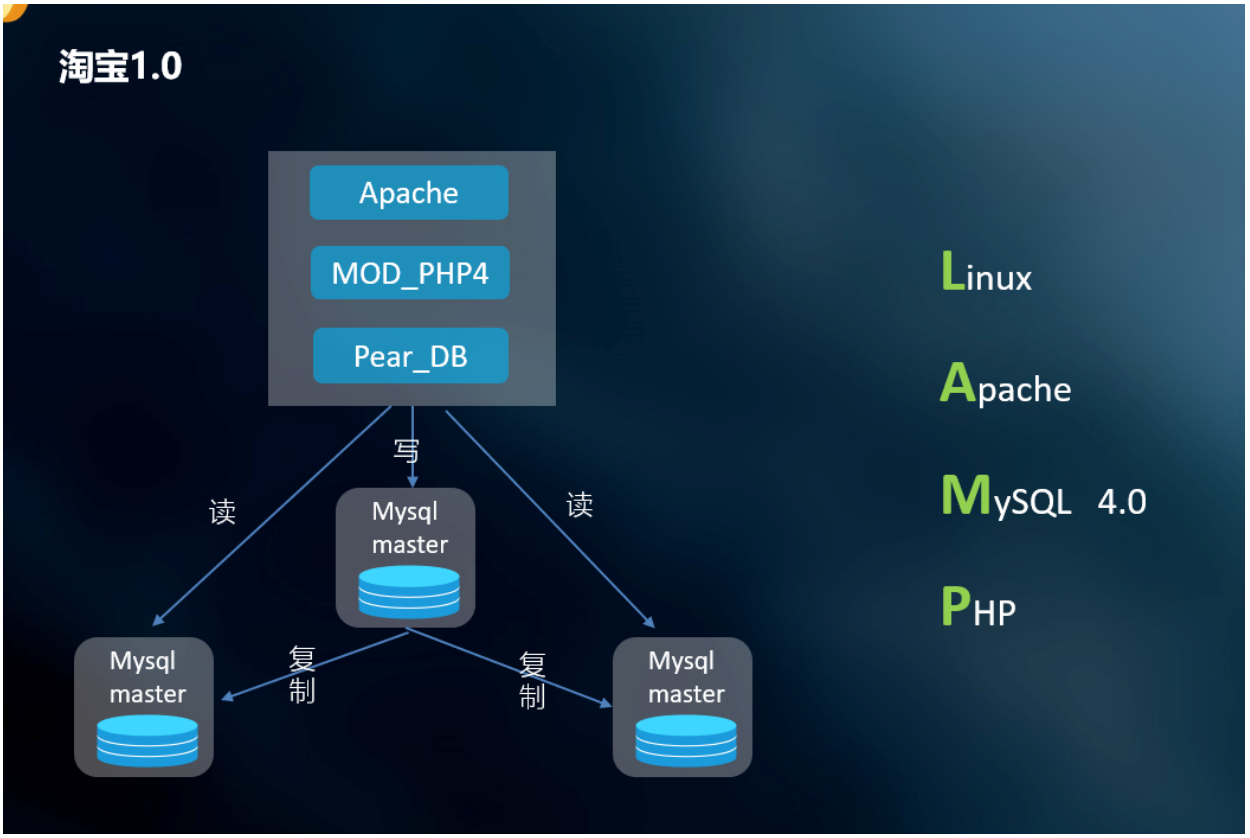
ums 用户模块

那大家看到的是一个已经相对来说比较完整的一个分布式电商网站。那他从最初的时候是怎么样的?

接下来我和大家来讲讲电商发展过程中技术的演变这块。

电商发展与技术演变:

淘宝技术这十年.pdf
14.73MB



个人 网站：内部调用（JVM）完成 数据库项目都同一台服务

下单》登录》session 》http

应用程序优化：

数据库优化：

1、换数据库 mysql》redis 》oracle》tidb 关系型性能

读多写少（访问商品）》redis 搜索 es内存搜索（全量、增量）

2、分库分表（读写分离）

jdbc应用层：shardingsphere（shardingjdbc）、tddl

proxy代理层：mycat、mysql-proxy

跨部门的问题：

垂直：　会员部门、商品部门 周四（开发》测试》预演》生产）

会员部门：成功（kpi）　失败（回滚）

性能解耦、开发的问题

带来很多问题：1个N个服务，服务之间调用依赖也会增多。

rpc调用：dubbo、fegin、http

1、异步调用 2、分布式事务（数据库事务A B 两个数据库、A服务、B、C）

rocketmq、shardingsphere（mq）、异步下单 （异步、削峰、解耦）

带来一些问题：消息重复、消息丢失、消息积压、幂等性问题

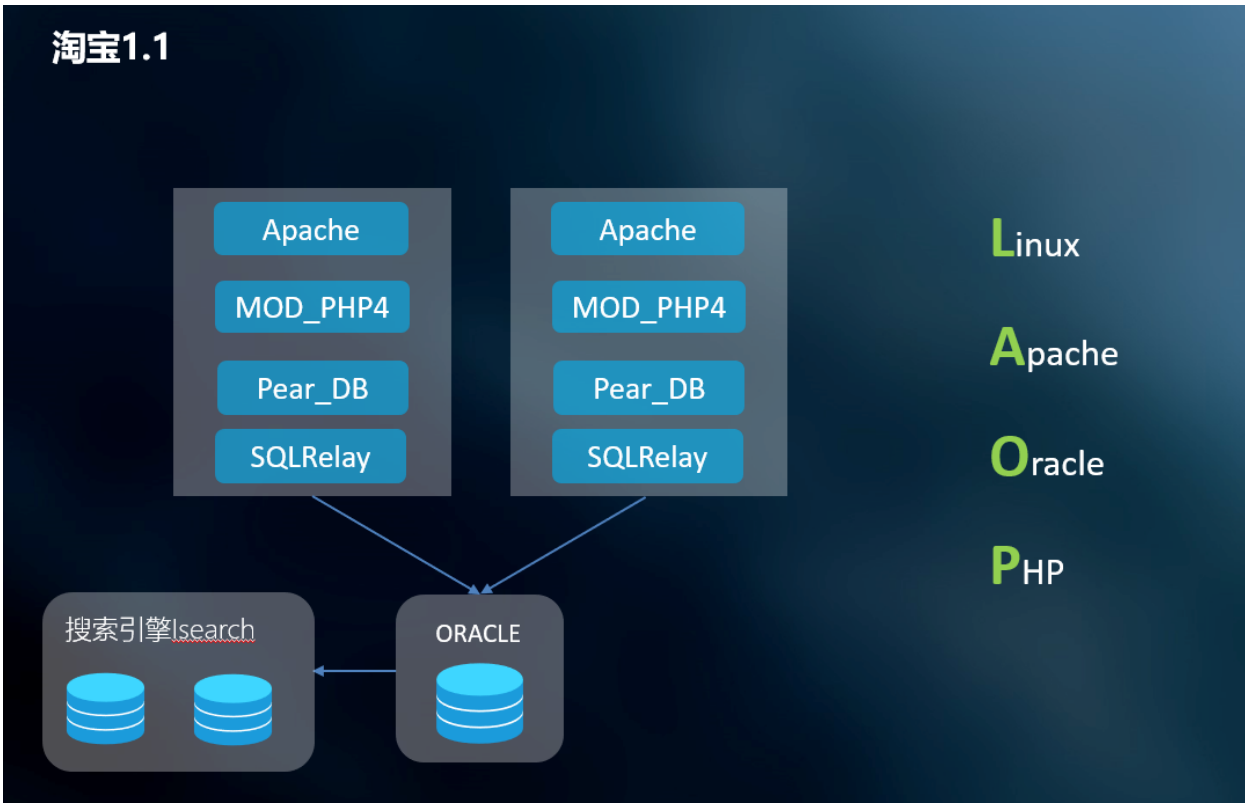
Monkey 100

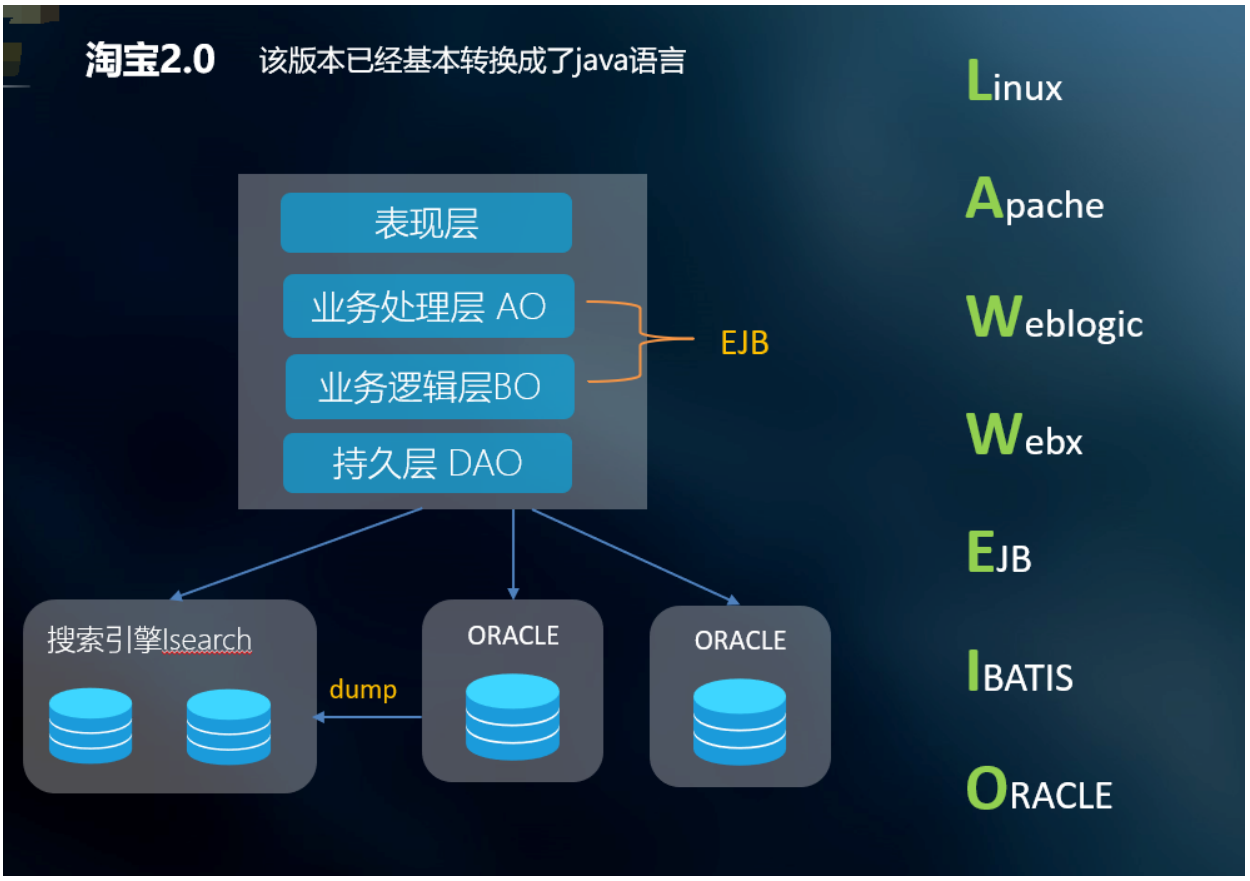
+100

+100

200

淘宝1.0升级版：





什么是4.0 时代?

云原生：

微服务、容器化部署、DevOps

微服务：

为什么这两个系统会相互影响?

微服务的本质是把一块大饼分成若干块低耦合的小饼，比如一块小饼专门负责接收外部的数据，一块小饼专门负责响应前台的操作，小饼可以进一步拆分，比如负责接收外部数据的小饼可以继续分成多块负责接收不同类型数据的小饼，这样每个小饼出问题了，其它小饼还能正常对外提供服务。

微服务解决的是我们软件开发中一直追求的低耦合+高内聚、单一原则

DevOps：

DevOps的意思就是开发和运维不再是分开的两个团队，而是你中有我，我中有你的一个团队。我们现在开发和运维已经是一个团队了，但是运维方面的知识和经验还需要持续提高。

持续交付：

持续交付的意思就是在不影响用户使用服务的前提下频繁把新功能发布给用户使用，要做到这点非常非常难。我们现在两周一个版本，每次上线之后都会给不同的用户造成不同程度的影响。

容器化：

容器化的好处在于运维的时候不需要再关心每个服务所使用的技术栈了，每个服务都被无差别地封装在容器里，可以被无差别地管理和维护，现在比较流行的工具是docker和k8s。

所以你也可以简单地把云原生理解为：**云原生 = 微服务 + DevOps + 持续交付 + 容器化**

项目演示:

1、前端环境：

tulingmall-front 前端部署与运行README.md

mall-admin-web前端部署与运行README.md

2、后台环境：（镜像）

host配置

nacos

redis

zk

rocketmq

mongodb

seata

项目中我们解决了哪些技术问题?

1、用户模块 （网关、授权）

2、商品新增 (数据库设计)

3、商品访问（多级缓存)

4、购物车（ 在线购物车、离线购物车）

5、下单（秒杀、防刷、分布式事务、取消订单）

6、订单 （分库分表、异步）

7、营销系统（秒杀业务设计）

本堂课笔记：

https://note.youdao.com/ynoteshare1/index.html?id=303b553ebbb1632d4f7b41f13a29eefd&type=note