淘淘商城项目总结

第一天：

1. 电商行业背景介绍：找工作要找互联网相关的工作。
2. 系统架构：
   1. 什么是集群
   2. 什么是分布式架构
   3. Soa架构，面向服务的架构。
3. 工程搭建使用maven管理工程
   1. 服务层工程
   2. 表现层工程
4. Svn的使用
   1. 把代码上传到svn
   2. 从svn下载代码

第二天：

1. 服务中间件dubbo
   1. 服务的生产者
   2. 服务的消费者
   3. 注册中心
   4. 监控中心。
2. 注册中心使用zookeeper实现。
3. 发布服务需要使用容器来发布。Spring容器。
4. 消费服务需要使用容器来引用。Spring容器。
5. 框架整合
   1. Ssm框架
   2. 发布服务：dao、service、interface
   3. 引用服务：controller、interface
6. 商品列表查询
   1. Pagehelper分页插件
   2. 商品列表查询实现
7. Dubbo的监控中心。

第三天：

商品添加实现

1. 商品类目选择
   1. easyUI的Tree控件。异步tree。
   2. State：如果有子节点就是closed，如果没有就是open
2. 图片上传
   1. 图片服务器。fastDFS。分布式文件系统。
   2. fastDFSJava客户端的使用方法。
   3. 使用KindEditor的多图片上传插件。
   4. 浏览器兼容性问题。
3. 富文本编辑器的使用。KindEditor。

第四天：

1. 前台系统搭建
2. 前天系统动态展示
   1. 后台cms系统的实现
      1. 内容分类管理
      2. 内容管理
   2. 从数据库中取内容信息。动态展示到页面。实现大广告位的动态展示。

第五天：

分布式缓存的实现

1. 问什么要有缓存？解决高并发、减轻数据库压力。
2. 使用redis做缓存。
3. 五种数据类型
   1. String
   2. Hash
   3. List
   4. Set
   5. SortedSet
4. Redis的持久化
   1. Rdb快照（默认持久化方案）
   2. Aof 把命令保存到文件中。
5. Redis集群
   1. Redis-cluster
   2. Hash槽
   3. 集群中至少有三个节点。
6. 在业务逻辑中添加缓存
   1. 添加缓存不能影响正常业务逻辑
   2. 查询数据之前查询缓存
   3. 查询数据之后添加到缓存中。
7. 缓存同步。

第六天：

1. solr服务器搭建。
2. 从数据库中导入数据到索引库
3. 搜索功能的实现。

第七天：

1. solrCloud集群的搭建。
   1. Zookeeper集群，管理solr集群。
   2. Solr集群。
2. 集群分片。
   1. Shar1+shar2+sharn=collection
   2. 每片都保存到不同的服务器。
3. 把搜索功能切换到solr集群。
   1. SolrJ客户端连接集群。
   2. 修改spring的配置文件切换到集群。

第八天：

添加商品后吧商品添加到索引库。

1. 使用ActiveMQ实现。
2. ActiveMQ
   1. Queue
   2. Topic
3. 生产者
4. 消费者
5. ActiveMQ整合spring。
   1. JmsTemplate对象发送消息
   2. 接收消息使用MessageListener的实现类。
   3. 配置监听容器，直接MessageListener。

第九天：

商品详情页面

1. 动态展示jsp+redis
   1. 使用redis缓存商品数据
   2. 设置缓存的过期时间。提高缓存的利用率
2. 网页静态化
   1. Freemarker
   2. Freemarker的语法
3. 添加商品动态生成静态页面。
   1. 监听商品添加的事件ActiveMQ的topic
   2. 访问页面。需要使用http服务器。使用nginx。

第十天：

Nginx

1. nginx实现虚拟主机。
   1. 基于端口的虚拟主机
   2. 基于域名
2. Nginx实现反向代理。
   1. 把请求转发给应用服务器。
   2. Proxy\_pass+upstream
3. 负载均衡
4. Nginx的高可用。
   1. Keepalived+nginx实现高可用。

第十一天

1. sso系统的实现
   1. Session共享
      1. 使用redis模拟session
      2. 设置过期时间。
   2. Token跨越访问
      1. 使用cookie的跨越访问方式。
2. 通过token查询用户信息。
   1. 从redis中取用户信息
   2. 重置过期时间。
3. Jsonp
   1. Callback参数回调方法名
   2. 服务端拼接js语句返回

第十二天

购物车

1. 使用cookie实现购物车
   1. 优点不占用服务端存储空间
   2. 实现简单
   3. 缺点：更换设备购物车不能同步。
2. 解决方案
   1. 把购物车消息保存到服务端。
      1. 可以是数据库
      2. 可以是redis（推荐）
3. 订单确认页面
   1. 强制用户登录
   2. 拦截器实现。

第十三天

1. 订单提交功能实现。
2. 系统部署
   1. Tomcat热部署。