

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **余胜军** | | | |  |
| 20岁 | | 男 | | 本科 | | 一年经验 |
| | | | [644064779@qq.com](mailto:644064779@qq.com) | |

|  |  |
| --- | --- |
|  | **个人优势** |

1. 熟练掌握JavaSE基础知识，熟悉多线程与并发编程，拥有丰富的系统设计分析能力及面向对象分析和设能力，熟悉常用的设计模式（代理、单例、工厂、建造者、观察者、策略、模版、装饰、门面等）及反射、自定义注解、泛型等；
2. 熟练掌握Eclipse/IDEA、Svn、Git、Maven/构建企业级私服仓库等开发及版本控制工具，熟练掌握Java基础技术Jsp、Servlet、Jdbc，熟练掌握Web应用系统开发，具备的独立开发能力；
3. 熟悉java并发编程 Java内存模型、volatile关键字的原理、synchronized锁原理、锁的升级偏向锁、轻量锁、

重量锁、

1. 熟悉JVM参数调优、垃圾回收机制原理、性能检测工具、生产环境服务器jvm调优配置、压力测试工具JMeter、阿里代码扫描工具；
2. 熟练使用Tomcat/Jetty服务器，发布部署web项目及Docker+K8s+Jenkins实现自动化部署及灰度发布，熟练使用Nginx+Keepalived实现服务高可用、OpenRest集合Lua代码实现接口的Api限流、黑名单与白名单、实现亿级别商品详情页面；
3. 熟练使用Oracle、MySQ等关系型数据库，熟练使用Redis、MongoDb等非关系型数据库及搜索服务框架Elasticsearch和Lucene；
4. 熟练运用Spring5、Mybatis3、SpringBoot2.0等优秀的开源框架，熟悉Spring的IOC、DI、AOP设计思想；
5. 有阅读过Mybatis3.x （Mapper接口绑定、二级缓存设计、插件开发）、和增强版本MyBatisPlus、Spring5.x（IOC容器、AOP切面编程、事务底层实现、webflux响应式编程）、SpringMVC（执行流程）、SpringBoot 2.0框架源码，能够完全完全独立的纯手写MyBatis、SpringMVC、SpringBoot框架；
6. 有深入研究过Java8以上List、Set、Map源码分析，能够完全纯手写Jdk1.8HashMap（红黑树与数组）、Arraylist、Conurrent HashMap、LinkeList集合；
7. 数据结构（ 链表、数组、红黑树、 B+树、 平衡二叉树）、算法分析：冒泡、快速、折半
8. 熟悉SOA、分布式、微服务、大型网站架构，熟悉RPC远程调用、Dubbo、Dubbox、SpringCloud第一代和第二代、SpringCloudAliBaba等微服务框架；
9. 熟悉SpringCloud体系下常用组件、注册中心Eureka、Consul载均衡器Ribbon、客户端调用工具Rest和Feign，分布式配置中心Config ，服务保护Hystrix网关Zuul ，Api管理 Swagger ，服务链路Zipkin，消息总线Bus等；
10. 熟悉SpringCludAlibaba核心组件Nacos实现注册中心、分布式配置中心、高可用集群模式的、新一代服务网关Gateway、动态微服务网关路由的模式及高可用、使用sentinel可以对热词、系统自适应的限流模式、整合seata解决分布式的事务，深入了解过seata底层实现原理、mysql与分布式缓存数据一致性框架canal
11. 熟练掌握分布式框架集群选举算法实现原理：Zookeeper的ZAB协议、Nacos集群的Raft算法的原理、Eureka的集群相互注册实现原理
12. 熟练使用网络通讯协议，熟悉nio、netty网络编程、能够基于Netty手写一些分布式解决方案的框架，比如（Dubbo、分布式配置中心、注册中心、事务框架、消息中间件）、熟练使用json、xml通讯格式，熟悉常见消息中间RocketMQ、Kafka 使用及分布式消息中间件在微服务情况下应用场景；
13. 熟悉Restfull API开发规范，熟悉三方授权OAuth协议、JWT协议、对接第三方支付接口(银联、支付宝、微信支付)及微信公众号开发和微信小程序开发；
14. 熟悉分布式协调工具Zookeeper、深入研究过Zookeeper一致性原理、系统的集群、高可用、高并发解决方案 服务降级、限流、熔断、隔离等，熟悉分布式和微服务常见解决方案（如：分布式任务调度平台、分布式服务追踪与调用链、分布式Session一致性问题、分布式锁解决方案、分布式事务解决方案LCN和Seata、分布式日志收集ELK+Kafa、分布式生成全局ID、微服务开放平台设计）；
15. 熟悉junit单元测试、jmeter压力测试工具、testNG自动化测试等主流测试框架；
16. 熟悉互联网安全架构平台设计例如XSS、SQL注入问题、防盗链等，互联网API接口幂等设计、互联网API接口安全设计等；
17. 有丰富微服务电商项目开发实战经验，熟悉整个微服务电商项目的架构流程，并实际参与过微服务电商项目，从项目需求讨论、研发阶段、分布式环境部署、技术难题解决、上线CDN部署等。

|  |  |
| --- | --- |
|  | **期望职位** |

Java 上海 30k-50k 电子商务·互联网·移动互联网

|  |  |
| --- | --- |
|  | **工作经历** |

|  |  |
| --- | --- |
| 上海每特教育科技有限公司 | |
| Java高级开发工程师 | 研发平台 | 2018.01—至今 |

1.负责后台接口开发与设计

2. 项目需求及上线评审，系统架构方案讨论

3. 负责指导其他同事，项目需求子任务拆分。

4. 处理线上相关问题。

5. 系统重构

**工作经历说明**

在每特科技公司，从事每特微服务电商项目开发，主要参与项目需求讨论、研发阶段、分布式环境部署、技术难题解决、上线CDN部署等

**A. 项目采用SpringBoot2.0+SpringCloudAlibaba构建微服务电商项目**

**1.使用Nacos作为注册中心，实现服务治理**

**2.使用新一代Gateway网关框架管理服务请求入口**

**3.使用Ribbon实现本地负载均衡器和openFegin客户端调用工具**

**4.使用Sentinel服务保护框架(系统自适应限流、降级、热词限流等)**

**5.使用消息总线Stream RocketMQ和Kafka**

**6.微服务API接口安全控制与单点登陆系统CAS+JWT+Oauth2.0**

**B. 分布式基础设施环境构建**

**1.分布式任务调度平台XXL-Job**

**2.分布式日志采集系统ELK +kafka**

**3.分布式事务解决方案Seata和RocketMQ**

**4.分布式锁解决方案Zookeeper、Redis**

**5.分布式配置中心Nacos**

**6.高并发分布式全局ID生成雪花算法**

**7.分布式Session框架Spring-Session 禁止使用Session 使用token**

**8.分布式服务追踪与调用链ZipKin**

**9.基于canal解决MySQL与Redis一致性问题**

**10.基于网关统一解决微服务接口跨域问题**

**11.基于openresty+lua+Redis实现亿级商品详情页面**

**C.项目运营与部署环境**

**1.分布式设施环境，统一采用docker安装**

**2.使用jenkins+docker+k8s实现自动部署**

**3.微服务API管理ApiSwagger**

**4.使用GitLab代码管理**

**5.统一采用第三方云数据库**

**6.使用七牛云服务器对静态资源实现加速**

**7.构建企业级Maven私服**

**D.前端技术架构**

**1.全完采用Vue前后端分离架构模式**

**项目一**：**每特微服务电商项目微信公众号  
项目名称**：**每特微服务电商项目微信公众号**

**开发时间/开发周期**：2013.06-至今

责任描述：

1. 构建微信公众号回调接口通知平台服务
2. 构建自定义实现关键字回复
3. 定时调用微信消息模版的接口实现消息推送
4. 实现用户扫码关注公众号登陆

技术的描述：

微信公众号环境准备

构建微信开发项目，整合weixin-java-mp框架实现我们的微信公众号快速开发

构建自定义实现关键字回复

1.使用策略模式实现对微信公众号的不同的行为减少if判断，提高代码可用性

2.数据库设计关键字对应的回复信息的内容，微信客户端输入关键字查询我们数据库关键字回复表，

如果该关键字在我们数据库中不存在情况下，调用第三方接口查询回复

3.根据微信消息的全局id解决消息响应的幂等问题

调用微信消息模版的接口实现消息推送

1. 整合分布式任务调度XXL-Job实现定时群发消息推送
2. 根据用户openid调用微信消息模版接口实现推送
3. 对我们XXL-Job的执行器实现集群，分片定时模式群发
4. 对调用接口失败信息采用日志记录，后期可以人工或者定时任务补偿

扫码关注公众号的实现

后端实现：

1. 调用微信接口生成一个带参数的二维码链接 中传递了userid
2. 当用户扫二维码的时候，在将该参数的userid传递到我们的微信开发回调开发接口

查询是否有关联过openid。

前端实现：

Vue前端采用ajax每个2s定时轮训，查询该用户是否有关联过openid，如果没有关联openid情况下展示

带参数的二维码链接。

**项目一**：**每特微服务电商平台**

**项目名称**：**每特微服务电商平台-会员服务/SSO服务**

**开发时间/开发周期**：2013.06-至今

**使用技术:SpringBoot2.0+SpringCloud2.0+jenkins+docker+k8s+ApiSwagger+Maven**

**责任描述**：

1. 负责创建会员服务和SSO单点登陆系统项目
2. 负责会员服务核心模块开发(令牌Token登陆和多端唯一登陆接口的开发)
3. 负责会员服务QQ和微信联合登陆功能开发
4. 负责SSO单点登陆系统的开发

**技术描述**：

**令牌登陆实现:**

客户端（移动端、PC端、微信端）调用会员服务登陆接口成功后，返回对应的Token令牌，有效期为90天，客户端保存该Token信息，客户端使用令牌关联用户会话信息。

使用UUID随机生成Token令牌、用户登陆成功后会生成对应的Token令牌作为key，存入在Redis中，Redis中令牌对应的Value为UserId、有效期为90天。

**唯一登陆实现:**

后端实现：

1. 整合Spring 异步注解@Async和线程池的配置，采用多线程的形式，异步的处理唯一登陆操作，提高我们的登陆接口的响应接口速度。
2. 多线程操作如果产生异常情况下，采用日志的形式记录下来，后期人工或者是定时补偿；
3. 因为使用了多线程处理异步操作，会导致CPU调度的非常频繁从而会降低我们的服务器的性能，所以后期我们改用为使用mq异步的处理我们的唯一登陆操作，提高我们的登陆接口的响应
4. 使用数据库表记录每次客户端登陆的Token令牌，每次产生新的用户Token都会删除上一次登陆的Token，客户端采用定时器检查当前客户端保存的Token是否过期或者是否重复登陆。

前端实现：采用定时器形式，使用Token调用验证接口是否已经过期。

**联合登陆实现：**

调用腾讯/微信开放平台，使用OAuth2.0授权获取用户相关openid，使用openid关联会员服务账号信息

**SSO登陆实现:**

使用XXL-SSO框架实现微服务平台单点登陆开发

**项目一**：**每特微服务电商平台分片定时消息推送平台**

**项目名称**：**特微服务电商平台分片定时消息推送平台**

**开发时间/开发周期**：2013.06-至今

**使用技术:SpringBoot2.0+SpringCloud2.0+jenkins+docker+k8s+ApiSwagger+Maven**

**责任描述**：

1. 负责开发微信活动消息模版的接口
2. 整合分布式任务调度平台XXL-JOB
3. 负责百万级定时消息模版跑批群发消息

**技术描述**：

**微信消息模版接口**

在微信公众号创建微信消息模版，采用模版id+openid对关注的用户发消息模版

**整合分布式任务调度**

搭建分布式任务调度平台xxl-job环境搭建 xxl\_admin

**负责百万级定时消息模版跑批群发消息**

1. 将定时任务逻辑与我们逻辑代码完全分开
2. 单独构建定时任务模块执行器
3. 对我们执行器(定时任务项目)，采用集群分片的模式部署 根据不同的index 查询不同分内容实现跑批
4. 执行采用分片集群模式部署可以实现动态与缩容
5. 中途如果定时任务出现异常，采用日志形式后期人工补偿

**项目二**：**每特微服务电商平台**

**项目名称**：**每特微服务电商平台-商品搜索服务**

**开发时间/开发周期**：2013.06-至今

**使用技术:SpringBoot2.0+SpringCloud2.0+jenkins+docker+k8s+Elasticsearch+Maven**

**责任描述**：

1. 负责整个商品数据库的设计
2. 负责商品搜索服务接口的开发
3. 负责实现Elasticsearch与MySQL数据库同步问题
4. 负责商品详情页面静态化开发

**技术描述**：

**商品数据库设计:**

分类表采用无限级分类策略，采用**SPU(产品单元)、SKU(库存量单元)具体商品规格。**

**商品搜索服务接口:**

为了提高用户的搜索效率，采用Elasticsearch存放商品信息数据，集成IK分词+PingYin插件，实现拼音和中文分词高效查询。

**MySQL与ES数据同步:**

1. 采用MQ实时性和Logstash定时实现MySQL与ES同步，可能会存在短暂的延迟

MQ同步方案:发布/修改/删除/商品的时候，以MQ异步形式同步到**Elasticsearch**中，可以实时保证双方数据一致性问题。

Logstash同步方案:使用Logstash定时方式读取数据库最新的数据>=update\_time时间 ，让后在格式化成json格式输出到

Elasticsearch中。

2.基于canal订阅mysql的binlog文件，canal将binlog文件转换成json发送给MQ服务器端，MQ消费者异步将数据写入到**Elasticsearch**

**商品详情页面静态化:**

1. **后台每次发布商品的时候，在nginx结合lua代码生成模板页面，在通过lua动态的获取Elasticsearch数据**

**项目三**：**每特微服务电商平台**

**项目名称**：**每特微服务电商平台-聚合支付平台**

**开发时间/开发周期**：2013.06-至今

**使用技术:SpringBoot2.0+SpringCloud2.0+jenkins+docker+k8s+ApiSwagger+Maven**

**项目描述**：构建微服务聚合支付平台，整合第三方支付接口（支付宝、银联支付、微信支付等），封装成接口提供给服务接口使用。

**责任描述**：

1. 与第三方支付接口平台技术人员对接
2. 聚合支付平台数据库表结构设计
3. 基于设计模式构建聚合支付平台
4. 基于MQ实现分布式事务解决方案
5. 使用xxl-job实现聚合支付对账
6. 常用遇到支付难题问题解决

**技术描述**：

**与第三方支付接口平台技术人员对接**

负责支付宝、微信支付、银联支付等支付接口技术人员沟通对接

**聚合支付平台数据库表结构设计**

负责聚合支付平台数据库表结构设计，payment\_channel渠道表、payment\_transaction流水表、payment\_transaction\_log日志表等实现完全可视化界面支付开关控制支付渠道。

**基于设计模式构建聚合支付平台**

使用策略+工厂设计模式实现支付接口提交html表单元素重构、使用模版方法+工厂设计模式实现共同异步回调代码重构，采用多线程+MQ实现支付回调日志收集

基于预提交参数的形式，保证接口的传递的安全性

**使用xxl-job实现聚合支付对账**

使用分布式任务调度平台XXL-JOB实现支付接口产生延迟导致订单状态不同步实现自动化补偿，每天晚上两点钟时间定时触发任务，商户端主动使用支付id调用第三方支付渠道接口查询支付状态是否已经支付过，如果第三方支付渠道接口已经支付过，同步商户端订单状态。

**基于MQ实现分布式事务解决方案**

在微服务系统中，支付服务调用积分服务接口增加对应积分，该场景存在分布式事务问题，我们采用rabbitMQ实现+补单形式保证最终一致性实现双方数据同步

**基于seata解决分布式事务的问题**

**常用遇到支付难题问题解决**

问题1：支付回调接口中，产生延迟通知？导致支付状态不一致问题

解决方案：商户端使用任务调度平台主动调用第三方接口查询，实现数据同步

问题2：重试支付回调接口时，如何保证回调接口通知幂等性问题

解决方案：因为第三方支付接口采用间隔性重试机制，使用支付全局id查询支付状态已经支付，及时响应成功状态通知给第三方支付平台不在继续重试。

问题3：支付金额与商品金额如果不一致时，如何处理

解决方案：在支付回调中查询待支付表记录金额与实际回调金额是否一致，如果不一致该笔订单纳为异常订单

问题4：第三方支付平台如何与对接系统保证分布式事务问题

解决方案：使用重试+补偿+日志记录形式保证最终一致性实现数据的统一

问题5：支付服务如何与其他系统保证分布式事务问题

解决方案：MQ/TCC/LCN分布式事务解决框架解决分布式事务问题

**项目四**：**每特微服务电商平台**

**项目名称**：**每特微服务电商平台-分布式日志采集平台ELK+Kafka**

**开发时间/开发周期**：2013.06-至今

**使用技术:SpringBoot2.0+SpringCloud2.0+jenkins+docker+k8s+ApiSwagger+Maven**

**项目描述**：搭建分布式日志采集平台统一管理微服务日志文件，方便开发者生产环境查询日志

**责任描述**：

1. 搭建ELK+Kafka收集日志平台
2. 使用Aop技术将日志信息投递到Kafka中
3. Logstash订阅Kafka将日志信息输出到ES中

**技术描述**：

**ELK+Kafka收集日志平台原理**

使用AOP技术环绕和异常通知拦截日志文件信息，转化成json格式投递到Kafka主题中，Logstash订阅Kafka主题实时将日志信息输出到ES中，最后使用kibana调用ES接口以图形报表信息查询日志。

**项目五**：**每特微服务电商平台**

**项目名称**：**每特微服务电商平台-微服务智能报警系统**

**开发时间/开发周期**：2013.06-至今

**使用技术:SpringBoot2.0+SpringCloud2.0+jenkins+docker+k8s+ApiSwagger+Maven**

**项目描述**：构建微服务智能报警系统，监听服务出现代码异常、cpu、内存、硬盘飚升 会主动的将该异常上报给开发者人员，减少开发者搜索日志的成本。

**责任描述**：

基于Aop切面或者全局捕获异常拦截系统错误日志结合Kafka将日志信息发送给微信开发者。

**技术描述**：

1. 基于Aop的形式拦截系统错误错误
2. 调用Linux接口查询cpu、内存、硬盘信息
3. 将异常信息转换为json格式投递到MQ中
4. 整合kafka异步的将错误日志上报给开发人员

**项目六**：**每特微服务电商平台**

**项目名称**：**每特微服务电商平台-微服务秒杀抢购商品**

**开发时间/开发周期**：2013.06-至今

**使用技术:SpringBoot2.0+SpringCloud2.0+jenkins+docker+k8s+ApiSwagger+Maven**

**项目描述**：

基于SpringCloud构建秒杀抢购活动，能够支持百万级级别以上访问

**责任描述**:

1. 秒杀抢购实现方案技术选型
2. 负责前端静态资源带宽优化
3. 负责核心秒杀接口代码编写

**技术描述**：

**秒杀抢购实现方案技术选型**

前端：

七牛云静态资源服务器+CDN内容分发实现减少服务器带宽资源、复杂验证码

后端：

网关实现安全过滤、幂等、高并发、限流、用户频率设计 、基于MQ+库存令牌桶实现防止超卖和减少数据库IO的操作

**负责前端静态资源优化**

1. 使用maven插件自动对静态资源实现压缩、每次发布的时候使用版本号码更新静态资源
2. 将页面静态资源(css、js、img)等，存放到七牛云静态资源服务器，七牛云静态资源能够实现CDN内容分发，将静态资源缓存到全国各地节点，能够减少服务器的带宽传输，从而提高带宽传输速度。

**负责核心秒杀接口代码编写**

**网关:**

1. 基于责任链设计模式实现对网关权限控制实现动态开关
2. 使用guava RateLimiter 令牌桶算法实现对秒杀接口限流
3. 使用redis的setnx命令实现对用户频率限制

**后台运营:**

1.每次发布商品的时候，提前在redis中生成好对应相等的库存令牌

**秒杀接口：**

1. 使用Sentinel对秒杀接口实现服务隔离防止雪崩效应
2. 从redis中获取对应商品的令牌，只要能够抢到令牌就能够秒杀成功
3. 使用异步形式RabbitMQ实现对商品的库存修改
4. 使用乐观锁机制(版本号)+库存>0判断防止库存超卖问题

|  |  |
| --- | --- |
|  | **教育经历** |

|  |  |
| --- | --- |
| 武汉华中科技大学 | |
| 计算机 | 本科 | 2010—2014 |

SpringIOC原理 HashMap 自主研发 模拟面试系统 预计年底