

应聘职位：Java 高级工程师

学历：统招本科/计算机科学与技术

邮箱：755399667@qq.com

张国才

工作年限：7 年

联系电话：18988790466

微信：18988790466

掌握技术能力

- 熟悉常用数据结构和排序、查找算法;
- 熟悉常用 IO 模型，了解内部实现机制;
- 熟练使用线程安全与非线程安全容器集合, 了解内部实现机制;
- 深入理解 Java 内存模型，对线程安全三要素: 可见性、有序性、原子性有深刻认识;
- 熟练使用多线程、线程池进行并发编程;
- 熟练使用 JUC 同步工具、Synchronized、Volatile 保证线程安全以及了解内部实现机制;
- 深入理解 JVM 底层原理，熟悉 JVM 常用垃圾收集器;
- 深入理解 Spring 框架，研究过 Bean 管理、IOC 容器管理部分核心功能源码;
- 深入理解 Mysql 索引原理、锁机制、事务实现机制;
- 熟练使用 Redis 缓存中间件，熟练使用 5 种数据类型使用场景以及了解底层数据结构实现; 熟悉缓存高并发的使用场景, 比如: 缓存雪崩、缓存穿透、缓存失效、热点缓存重建等;
- 熟练使用 RabbitMQ 消息中间件; 解决过常见消息通信场景的疑难问题，比如: 消息丢失、消息重复消费、大规模消息积压;
- 熟悉使用 Zookeeper，了解内部系统模型、会话机制、ZAB 协议、Leader 选举;

项目经验

华为应用市场内容管理中台

华为应用市场内容管理中台负责应用市场商店内容信息管理业务，包括开发者应用管理、开发者用户管理、内容运营管理、数据源管理等业务;本团队涉及服务组包括如下:

边缘微服务: 网关服务，统一管理对外接口暴露, 对接口进行鉴权、转发;面向应用市场运营平台门户提供服务;

应用采集微服务: 从 AGC，抓包引擎，快应用中心，鸿蒙/PC 类应用，运营上传等进行应用信息同步，并且转换成统一的格式交给后续去处理;

应用审核微服务: 应用分配，应用自动上架，应用先上后审等规则管理;审核任务的管理;审核中电子流的处理;审核任务的查询和导出;

应用文件处理微服务: 文件的下载，拷贝，CDN 预热，视频上传到华为腾讯云等;

操作日志微服务:应用市场管理服务统一的合规日志入口，提供操作日志的存储，合规审计查询等能力;

涉及技术栈: 数据库技术栈: Mybatis/Mysql/Rainbow 数据库中间件/Redis;
微服务技术栈:Java/Spring boot/华为开源微服务引擎 CSE/STS 安全票据服务;
配置中心化管理:WiseEye 动态配置中心/IAC 配置项代码化管理;
消息中间件: DMQ 分布式消息中间件;
分布式事务管理: ServiceComb Pack;

总体技术架构: 边缘微服务作为应用市场运营平台门户网关,对接口进行统一鉴权、路由转发;
STS 安全票据服务实现微服务之间认证以及加密传输过程中的敏感数据;
使用分布式框架 ServiceComb Pack 保证多个事务一致性;
采用 DMQ 分布式消息中间件,实现 bundle 应用编译 apk 完成、AGC 应用上下架、应用检测结果等业务异步消息通知,便于实现应用解耦、流量削峰、提供接口响应速度;
采用 Redis 实现查询内测游戏评级评分、全球管理微服务国家信息、应用场景数据隐私等业务数据缓存以提高请求响应速度与降低数据库压力;
采用分布式锁解决预审检测结果通知等入口同时新增应用属性导致属性重复;

参与功能设计: **应用上架审核业务应用属性重复插入异常处理**

产生的现象:

- 1、处理 apk 预审检测结果 DMQ 消息通知 与 应用上架审核流程 都会对 相同应用属性进行插入;
- 2、触发处理时间比较接近,但不至于频繁发生;
- 3、处理应用属性插入业务都有做重复性校验;

原因分析以及解决方法:

- 1、由于业务特性无法将两处应用属性插入归并到统一入口;
- 2、由于 应用上架审核流程复杂,处于一个长事务;因此插入应用属性还没有事务提交,导致预审检测结果 DMQ 消息通知里面重复性校验失效;
- 3、放弃使用数据表唯一索引解决重复插入问题,原因:
 - 由于应用属性表数据量比较大,涉及将应用 id、属性 id、属性值三个字段建立唯一组合索引,索引存储空间比较大、插入速度变慢;
 - 代码需要对重复键异常进行 try catch 捕获处理保证业务正常运行,很不优雅;
 - 后期涉及分库分表会导致唯一约束失效;
- 4、采用 Redis 分布式锁来解决,应用 id、属性 id、属性值作为 Key 后缀一部分,过期时间为 2 分

钟，因为应用上架审核流程事务处理最长时间也 10 多秒;主要保证其中一方事务提交之前分布式锁还存在就能解决插入冲突;

心愿先生(MR.WISH)网红爆款新零售福袋机

MR.WISH 心愿先生品牌体验礼盒福袋机，在线下娱乐商场中为每一位用户爆出高于预期的独特礼品。MR.WISH 福袋机入驻全国 50 多个核心城市, 活跃机量超过 3 万台。研发团队 100 多人，承担第三组开发小组长。MR.WISH 心愿先生福袋机业务由网关服务系统、认证鉴权服务系统、用户服务系统、商品服务系统、购物车服务系统、订单服务系统、优惠券服务系统、评论服务等系统支撑;

相关技术栈: 数据库相关技术栈: Mybatis/Seata/Mysql/Redis;

微服务技术栈: Java/Spring boot/Spring Cloud Eureka/Spring Cloud Ribbon/Spring Cloud OpenFeign/Spring Cloud Config/ Spring Cloud Gateway;

消息中间件: RabbitMQ;

总体技术架构: 服务组包括用户服务系统、商品服务系统、购物车服务系统、订单服务系统、优惠券服务系统、评论服务等微服务;

通过网关鉴权提供对外服务，采用分布式架构保证高可用、高并发;

订单服务系统、用户服务系统、商品服务系统等系统采用 Seata 框架保证用户下单等业务异库事务一致性;

采用 Redis 实现分布式缓存抗住高并发降低数据库压力,例如: 实时排行榜业务;现网采用集群模式保证 Redis 高并发、高可用;

采用 RabbitMQ 作为消息中间件，实现系统订单业务流量削峰、提高下单响应速度;

采用 Hystrix 实现订单、库存、用户等核心服务链路的实现限流熔断降级;

参与功能设计: **积分抽奖奖励拉活活动**

业务: 周末早上 10 点推送积分抽送活动，然后利用积分奖励与积分换购拉高用户活跃度，同时引导用户线下体验心愿先生盲盒购买;

涉及主要微服务: 营销活动服务、订单服务、商品服务、用户服务;

主流程:

1、用户进入积分抽奖转盘页面点击积分抽奖,向营销活动服务发起抽奖请求;订营销活动服务检验活动状态、生效时间、抽取次数效验;校验通过后,请求商品服务拉取所有积分商品库存并对有库存积分商品进行抽奖;

2、抽取出积分商品请求商品服务扣减该商品库存;

3、订营销活动服务调用订单服务免支付下单接口, 保存积分商品交易凭证;

4、请求用户服务增加用户积分

5、向用户展示抽取积分;

技术优化:

1、抽取到的积分商品进行库存扣减成功后,直接返回给用户抽取积分;订单服务免支付下单、用户服务增加用户积分均通过订阅 MQ 消息异步处理,缩短请求响应时间;

2、商品服务库存扣减接口保证幂等性;利用 Redis 缓存做库存扣减,降低数据库压力;在缓存中设置订单 id 扣减库存标识(订单 id、商品 id 组成 key),防止重复扣减,保证幂等性;

3、商品服务缓存库存扣减且异步消息同步数据库库存;数据库扣减库存利用死信队列做事务补偿,保证最终一致性;

4、周末早上由于抽奖流量很大,导致订单创建、积分扣减消息出现大量积压且用户延迟体验感差;增加订单服务、积分服务实例,增强消费能力;增大单个 Channel 预读取值;单个 Channel 使用线程池并行处理消息;

公司经历

中软国际科技服务有限公司	Java 高级开发工程师	2020 年 – 2021 年
· 外派入驻到深圳坂田华为应用市场业务部门参与开发工作;		
广州摘星者信息科技有限公司	Java 高级开发工程师	2018 年 – 2019 年
· 参与开发 MRWISH 心愿先生福袋机业务;		
惠州音平电声科技股份有限公司	Java 开发工程师	2015 年 – 2017 年
· 参与音平商城业务开发;		

开发语言与技术

微服务技术栈 : Java/Groovy/Grails/PlayFramework/Spring/Spring

Boot/Struts2/Hibernate/Mybatis/Dubbo/Spring Cloud 系列;

分布式事务框架: Seata/ServiceComb Pack;

数据库相关技术栈 : Mysql/Redis/Mongodb;

消息中间件: RabbitMQ;

搜索引擎: Solr/Easyrec;

管理、设计工具 : Trello/Microsoft Project/禅道/Confluence/Jira/UML/Draw.io/Proccesson;

相关开发工具 : Jenkins/Tomcat/WampServer/Nexus/Maven/Gradle/Git/Gitlab/Svn/IntelliJ IDEA /Eclipse;