

# 姜帅

手机号: 18243623585 | 邮箱: 1624252184@qq.com

年龄: 20岁 | 博客: <https://zihenzzz.github.io/>



## 个人荣誉

- 全国大学生数字媒体科技作品及创意竞赛 - 全国三等奖 (2023)
- 创新创业训练计划 (已立项) - 国家级 (2024)
- 中国大学生计算机设计大赛 - 省级二等奖 (2024)
- 第四届吉林省大学生人工智能创新大赛 - 省级二等奖 (2023)
- 第五届吉林省大学生人工智能创新大赛 - 省级三等奖 (2024)

## 开源贡献代码经历

### Tencent/spring-cloud-tencent

#### issue: polaris.discovery.heartbeat.enabled is not effective

排查:我们发现,即使设置heartbeat.enabled=false,日志中仍会出现[AsyncHeartbeat]相关日志。

PolarisServiceRegistry 的错误逻辑导致心跳机制 (Heartbeat) 错误地依赖 healthCheckUrl,而非正确的配置项 heartbeatEnabled。healthCheckUrl 的本意是提供健康检查的 HTTP 接口 (如 /health),用于外部系统 (如 Polaris 控制台) 探测服务状态,而非控制心跳。

修改部分:

- [此PR的信息添加到项目根目录下的CHANGELOG.md](#)中。
- 添加属性heartbeatEnabled于实体类和yaml文件,
- 原代码中通过 heartbeatExecutor.scheduleWithFixedDelay() 强制启用的心跳线程被完全移除,改为依赖 Polaris SDK 的 registerInstance() 方法内置的心跳机制。
- 原代码用StringUtils.isBlank(healthCheckUrl) 作为判断条件 → 错误逻辑。新代码直接使用 heartbeatEnabled 配置 → 正确逻辑。

## 实习经历

### 卓然科技 - 后端开发实习生 - (2025.03 - 2025.05)

- 监控CPU使用率, GC次数以及每次GC耗时,参与大对象GC问题排查与优化,排查出堆内存问题,使CPU占用率从25%降低到5%,GC次数明显减少
- 使用流式Flux会话相应机制并针对大模型的”幻觉“、知识时效性和专业领域知识缺失问题,采用RAG检索增强生成技术,构建知识库等
- 实现了线程池的优雅关闭和任务持久化,即使在宕机情况下,重新上线时重新恢复任务,增强了程序的健壮性和可靠性。

### 上海墨加科技有限公司 - 后端开发实习生 - (2024.06 - 2024.09)

- 支持业务方订阅周期扣款需求,在业务方发起首次扣款请求后,通过定时任务周期维护扣款任务。通过扣款失败主动回调,定时任务补偿来解决业务方-数据一致性问题。
- 针对其他服务高频率调用查询用户是否在黑名单,在读多写少且数据基本不修改的业务背景下,优化本地缓存+定时任务方案导致的一致性问题。

- 统一中台对**业务方错误码规范**，上线过程中,为避免造成全局影响，采用**反射+自定义注解+灰度控制接口影响范围**，事故排查与性能优化。
- 优化**本地事务和分布式锁使用不当导致潜在可能发生的数据库连接池打满,事务可见性问题**，提升接口性能，防止接口超时风险。
- 通过**责任链模式**实现订单挽留功能，**依次校验红包、库存正向触点**，基于Spring框架的Bean实现高度可扩展的设计，**提升了用户的订单留存率**

---

## 项目经验

### 智慧刷题系统

**项目描述：**这是一款专门为程序员打造的沟通交流社区，采用主流的微服务框架技术栈来做为技术架构。旨在统一程序员信息差，进行平台统一化，程序员可以在平台，完善自身知识，刷自身薄弱点面试题。

- 采用**DDD领域拆分思想**，对项目模块进行领域设计，如值对象、充血模型，专注自身职责
- 采用Gateway实现统一的鉴权及分布式会话共享功能，在网关层实现统一的全局异常处理；
- 测试进行了**线程池数据压测**，确定出合理线程池数量，探索**线程池与cpu关系公式**。
- **封装自定义的esclient，支持索引切换**，封装了常用的业务函数，实现网站的高亮搜索功能；
- 采取**工厂+策略模式**实现不同题目类型的操作，采取适配器模式实现oss对接；
- 基于**futuretask及completablefuture优化了分类标签的并发查询**，提升性能80%
- 选取**xxl-job配合redis的hash接口实现点赞收藏功能的开发及数据持久化**，减轻数据库交互压力；

### MyRaft

- **基于MIT6.824课程的实现的Raft程序**，基于GO语言实现方式，用Java进行重构
- **实现了Leader、follower、Candidate的基本逻辑**，如领导选举、日志复制、持久化
- 发生日志冲突使用**二分法快速查找冲突日志**，**减少日志同步时间**，从n降到logn

---

## 相关技能

- 熟练掌握**Java核心知识**，熟悉常用集合及数据结构、面向对象、类加载等；
- 有大模型相关知识，**coze、workflow、rag向量数据库、SpringAI、Langchain4j**等
- 熟练使用**MySQL数据库**，对MySQL底层和优化有一定了解，如**事务、MVCC、索引、锁机制**等
- 熟练使用**Redis中间件**，了解缓存、持久化、双写一致性、主从复制、数据过期策略、哨兵机制等
- 熟练掌握**JVM内存结构、双亲委派机制、垃圾回收算法、垃圾回收器等**
- 熟悉**Java并发编程**，各种锁机制、CAS、AQS、线程池等
- 熟悉**SSM、SpringBoot、SpringCloud**，深刻理解IOC、循环依赖等
- 熟悉**RabbitMQ、RocketMQ**消息队列中间件，了解死信队列、死信交换机等
- 了解**分布式原理**，如CAP理论，Base理论，分布式锁，Raft算法，了解限流算法等相关知识
- 熟悉**HTML、CSS、VUE3、uniapp、Element Plus**等前端技术

---

## 教育背景

长春建筑学院 - 本科 - 数据科学与大数据技术（2022.09-2026.07）（绩点前5%）

- **主修课程：**算法设计与分析、Hadoop、计算机网络、Spark、面向对象程序设计、数据挖掘与分析等