

徐傲轩

hi@xuaoxuan0623.ccwu.cc | 个人网站: xuaoxuan0623.ccwu.cc

GitHub: github.com/roseri66 | 求职意向: AI 应用开发实习生

教育背景

江汉大学 · 计算机科学与技术 · 本科 · 2024.09 - 2028.06 · 专业排名前 30% · 校级奖学金

核心课程: 数据结构与算法, 操作系统, 计算机网络, 数据库原理, 计算机组成原理

项目经历

山海镜 ShanHaiMirror | 受约束的 LLM 决策系统 (Java 网关 + Vue 回放器 + C++ 客户端) · 2026.07 - 至

今 · github.com/roseri66/ShanHaiMirror

技术栈: Java 17 · Spring Boot 3 · WebClient · Resilience4j · Bucket4j · Micrometer · Vue 3 + TypeScript · UE5/C++

项目内容: 让 LLM 在实时系统里参与决策, 难点不在调用, 而在它会超时、会返回非法 JSON、会给出破坏平衡的组合。需要一套架构保证「LLM 可以整体失败, 而系统不失败」。

我的贡献:

- 类型系统固化安全边界:** LLM 输出的类型里装不下数值, 数值只在四道护栏 (结构/预算/互斥/公平性) 之后由查表产生——越权决策在编译期就构造不出来
- Java 决策网关:** Spring Boot 3 + WebClient 非阻塞承接 4-5s 慢上游, Resilience4j 熔断 (刻意不加重试) + Bucket4j 限流 (429 是设计内的降级路径) + 语义指纹分桶缓存; API key 移出客户端
- 拒绝把护栏搬上服务端:** 会破坏「断网可用」不变式、C++/Java 双写必漂移、单机客户端本就是权威——信任边界不在那里, 技术上可行不等于应该做
- 决策数据回流:** 有界队列 + 单消费线程异步批量落库, 满则丢弃且可观测 (旁路不阻塞主链路)、库挂了照常返回决策; 用历史流水重放不同指纹方案, 据此发现并修掉一处分桶边界缺陷

项目成果: 三端 231 个单测全绿 (C++ 63 / Vue 84 / Java 84); 断网、后端停机、返回非法 body 三条降级路径均实测可用; 缓存命中将决策耗时从 3.6s 降至 2ms, P50 2.15s / P99 4.83s (Micrometer → Prometheus)。

MetaPool | Java 并发资源治理控制面 (已发布 Maven Central) · 2026.04 - 至今 · github.com/roseri66/MetaPool

技术栈: Java 17 · Spring Boot 3.4 · Micrometer · HikariCP · Bucket4j · Redisson · Netty · Maven 多模块/BOM · JMH · JUnit 5 · Testcontainers

项目内容: 一个应用里连接池、限流器、线程池、分布式锁各是一座孤岛, API、配置与监控互不相通。做一层治理控制面纳管 HikariCP/Bucket4j/Redisson 等成熟库, 统一生命周期与可观测性。

我的贡献:

- 能力接口双向隔离:** borrow/tryAcquire 等功能性 API 下沉为可选接口, 编译期根除「线程池 acquire() 抛 UnsupportedOperationException」式的里氏替换破坏; 反向亦然——Redis 不实现 Pool (多路复用无借还语义) 而 Netty 实现 (release 即引用计数): 语义存在才可映射
- SPI 扩展 + 统一可观测:** JDK ServiceLoader 做扩展点, 五次接入新资源 (线程池/锁/对象池/Redis/堆外内存) 核心零改动、commit 可核验; Micrometer 统一 tag 使七类资源指标共存于一块看板
- Actuator 自定义端点热调参:** 将 HikariConfigMXBean 与 Bucket4j replaceConfiguration 两套异构 API 统一为 apply(patch) 门面, 白名单 + 启动期 fail-fast, 连接池上限可不重启热调

项目成果: 12 个构件 (含 BOM) 已发布 Maven Central (2.0.0 → 2.4.0, 6 个版本); 193 个测试 0 跳过 + 从 Central 拉正式件的使用方冒烟; JMH 实测治理开销: 限流 +1.75%、连接借还 ~20ns 常量、真实 PostgreSQL < 0.1%; 曾据 JMH (自研线程池吞吐仅 JDK 的 48%) 主动推翻 1.0 自研路线。

技术栈

- 编程语言:** Java (主力, JDK 17)、Python、TypeScript / JavaScript、C++、SQL
- LLM 应用:** Prompt 工程与模板化、输出 Schema 校验与安全护栏、多 Provider 抽象与逐级降级、语义指纹缓存、跨语言 JSON 契约设计
- 后端框架:** Spring Boot 3 (自动装配、Actuator 端点扩展、starter 封装)、WebClient、Resilience4j、Bucket4j
- 前端:** Vue 3 (组合式 API、单文件组件) + TypeScript、Vite、Vitest、手写 SVG 可视化
- 可观测与工程:** Micrometer / Prometheus / Grafana、JUnit 5、JMH、Docker + Testcontainers、Maven 多模块与 Central 发布、GitHub Actions CI
- 计算机基础:** 数据结构与算法、计算机网络 (HTTP/TCP)、JVM 内存与 GC、JUC 多线程并发 (线程池/锁/CAS/内存可见性)、MySQL / Redis