

张嘉

18292060445 · plus_zzz@163.com · 2002-09

个人主页: pluszzz.top

Github: <https://github.com/Pluszzz>



教育经历

2024-至今

西北大学

软件工程/硕士

211 & 双一流

- “华为杯”中国研究生数学建模竞赛国家二等奖

2020-2024

西安邮电大学

信息与计算科学/本科

- 全国大学生数学建模竞赛省级二等奖
- CET-4

专业技能

- AI Agent 与大模型应用:** 熟悉基于大语言模型的智能应用开发流程, 理解 Agent 工作流设计、工具调用、任务规划、上下文管理等核心机制; 具备复杂业务场景下 LLM 应用架构设计经验
- LLM 应用开发:** 具备企业级 RAG 系统开发经验, 熟悉文档解析、Embedding、Milvus/Elasticsearch 检索、混合召回、知识增强和生成优化等完整链路
- Prompt Engineering 与上下文优化:** 理解大模型 Prompt 设计方法, 熟悉上下文窗口管理、Token 优化、结构化输出约束和推理效果优化方法, 具备长文本场景优化经验
- LLM 服务工程化:** 熟悉主流大模型接口协议和服务集成方式, 理解流式输出、Function Calling、多模型适配和模型服务调用链路设计
- Java 后端工程:** 熟悉 Java 语言及 Spring Boot 开发体系, 掌握 WebFlux/Reactor 异步编程模型; 具备企业级服务开发、接口设计和系统性能优化经验
- 数据库与数据处理:** 熟悉 MySQL 数据建模、索引优化、执行计划分析、事务机制; 具备复杂业务场景下的数据存储设计和 SQL 优化经验
- 工程化能力:** 熟悉 Git、Maven、Docker 等工程工具, 理解 Spec-Driven Development (SDD) 开发流程, 具备需求拆解、模块设计、服务部署、问题定位和工程交付能力

实习经历

西安恒品电子科技

AI Agent 工程师

2026-03 ~ 2026-08

PlusAgent(超级智能体)

项目介绍: 企业级 Multi-Agent Runtime 平台, 基于 Orchestrator-Workers 架构实现 Agent 生命周期管理、任务编排、工具生态接入、上下文管理和结构化产物生成, 支持复杂业务场景下多 Agent 协同执行

技术架构: SpringBoot 3 + WebFlux(Reactor) + MCP + Groovy + MyBatis Plus + MySQL + Vue3

职责描述:

- 多 Agent 编排引擎:** 自研状态机驱动的 Orchestrator-Workers 编排引擎, 实现任务拆解、子 Agent 调度、结果审查和终态收敛; 通过工具白名单、执行轮次限制和状态护栏控制 Agent 行为
- Agent Runtime 与事件系统:** 基于 WebFlux/Reactor 构建多 Agent 异步事件流架构, 设计统一 Event Protocol, 通过 SSE 实现 Agent 切换、工具调用和流式输出实时交互
- MCP Tool Runtime:** 实现 MCP JSON-RPC Client, 支持 stdio/HTTP 双传输协议, 实现工具发现、Schema 转换和动态注册, 将外部 MCP Server 能力接入 Agent Tool Runtime
- 上下文工程与记忆系统:** 构建 Agent Context Pipeline, 实现角色隔离、上下文裁剪、Token Budget 控制和 Prompt 组装; 结合增量摘要、质量评分和长期记忆沉淀实现长会话管理
- 大模型网关与产物生成 Pipeline:** 实现兼容 Anthropic/OpenAI 协议的模型接入层, 解决流式响应协议差异; 基于 Markdown AST 和 JSON Repair 机制提升结构化产物生成稳定性
- Skill Runtime:** 基于 GroovyClassLoader 实现动态技能加载框架, 支持工具脚本运行时编译、增量扫描和不停机更新, 实现 Agent 能力模块热扩展

PlusNexus(智能检索系统)

项目介绍: 企业级智能搜索与知识检索平台, 基于 LLM Query Understanding 构建自然语言查询解析、搜索计划生成、多源检索融合和智能结果生成能力, 实现用户通过自然语言查询企业结构化与非结构化数据。

技术架构： Spring Boot + Elasticsearch + Milvus + FastAPI + gRPC + MySQL + SSE

职责描述：

- **并发逻辑规划引擎：** 研发 Query Plan Tree 框架，利用 LLM 将复杂中文查询拆解为 AND/OR/NOT/SEARCH/COUNT 逻辑树。基于线程池并发执行子查询并引入短路优化，支持全集差集计算与结果分支溯源
- **元数据驱动 LLM 路由：** 通过 Java 端动态维护数据库字段 Schema，Python 端自动提取实体与规则注入，实现新增搜索维度仅需数据库增加一行配置，代码与 Prompt 零修改，解决了 Prompt 强耦合业务知识的痛点
- **多级降级与动态 DSL：** 设计“LLM 结构化分析 → 规则引擎 → 关键词字典匹配”的三级容错链，确保模型异常时系统依然具备稳定的检索能力；开发 DynamicQueryBuilder，实现 ES bool query 条件的自动路由与动态拼接
- **双协议与流式响应优化：** 采用 HTTP + gRPC 双协议架构，分别面向前端访问与 Java 服务间高性能调用；基于 SSE 实现阶段化流式输出，将结果生成拆分为“确定性数据构建 → LLM 摘要 → 引文关联”，避免 LLM 直接生成结构化事实数据，并提升长耗时任务的响应感知
- **微服务解耦与双级缓存：** 独立封装中文相对时间解析微服务（Time-MCP-server），统一处理 模糊时间表达；针对高频相同查询设计 Plan/Parse 双级缓存，大幅降低 LLM 调用成本并提升响应速度

项目经历

Ragent
https://github.com/Pluszzz/RagentAI
2025-09 ~ 2026-02

项目介绍: 企业级 RAG 智能体平台，核心实现了从文档入库到智能问答的全链路 RAG 系统。系统完整覆盖了意图识别、问题重写、多路检索、会话记忆、模型路由容错等企业生产环节。采用分布式队列限流控制并发，通过熔断器和优先级降级保障多模型场景下的高可用性

技术架构: SpringBoot + MyBatis Plus + Milvus + Redis + Redisson + Apache Tika + Sa-Token

职责描述:

- **多路检索架构设计：** 设计多路检索引擎（意图定向 + 全局向量检索 + 后处理流水线），在保证召回率的同时提升检索精准度
- **问题理解优化：** 基于 LLM 实现问题重写与拆分机制，自动补全上下文并拆解复杂问题
- **会话记忆管理：** 设计会话记忆压缩策略（滑动窗口 + 自动摘要），降低 token 消耗，解决长上下文 Token 超限问题
- **模型调度与容错：** 实现模型路由与三态熔断机制，支持多模型优先级调度与故障自动切换
- **意图识别体系：** 构建树形多级意图识别结构，置信度不足时系统主动引导用户澄清需求
- **工具调用框架：** 实现 MCP 工具集成框架，通过注册表自动发现，实现知识检索与工具调用融合
- **分布式限流控制：** 实现 Redis ZSET + Lua 的滑动窗口限流方案，支持全局与用户级并发控制

短链接平台
https://github.com/Pluszzz/ShortLinkPlatform
2025-05 ~ 2025-08

项目介绍: SaaS 短链接系统，为企业和个人用户提供了一个高效、安全和可靠的短链接管理平台。提供了深入的分析和跟踪功能，用户可以灵活地管理和优化其链接

技术架构: Spring Boot + Spring Cloud Alibaba + MySQL + Redis + RocketMQ + ShardingSphere + Sentinel

职责描述:

- **消息幂等控制：** 基于 Redis 实现 MQ 消费幂等性，确保消息在指定时间窗口内仅消费一次
- **缓存穿透防护：** 封装缓存穿透防护方案，通过双重判定锁与空值缓存优化大量不存在数据的查询问题
- **布隆过滤优化：** 使用 Redisson 布隆过滤器判断短链接存在性，提高判断效率，有效减少无效查询请求
- **流量治理与限流保护：** 整合 Sentinel 实现 QPS 级别限流，触发限流时进行降级处理保障系统稳定性
- **异步削峰架构：** 采用 RocketMQ 消息队列异步消费模式，削峰处理高并发访问下的监控数据写入需求
- **缓存一致性设计：** 设计“更新 DB 删除缓存”策略，保障短链接缓存与数据库之间的数据一致性
- **分布式并发控制：** 使用 Redisson 分布式读写锁，支撑链接短高并发场景下的数据修改操作
- **分库分表扩展：** 基于 ShardingSphere 实现分片路由，支撑跳转与分页场景下的定向查询