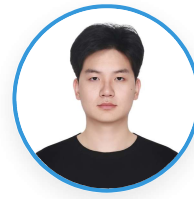




赵勇臻



联系方式: (+86) 181-6810-0075 | 邮箱: zhaoyzzz@outlook.com

技术博客: rzyn2020.github.io

出生年月: 2002.3.22 | 籍贯: 甘肃庆阳

教育背景

南京大学 - 硕士 - 软件工程专业 (2024.09-2026.06) (推免)

南京大学 - 本科 - 软件工程专业 (2020.09-2024.06)

专业技能

- 编程语言:** 熟练掌握 Java, Python, C++, 具备良好的编码规范。
- 后端技术:** 熟练掌握 SpringBoot 框架, 深入理解其核心原理; 具备快速学习并应用其他 Web 框架的能力。
- 基础知识:** 扎实掌握计算机网络、操作系统、编译原理、数据结构与算法等计算机核心基础。
- 开发工具:** 熟练使用 Git, Docker, RabbitMQ, Redis, MySQL 等常用开发与中间件工具。
- 语言能力:** 英语 CET-6, 能无障碍阅读英文技术文档、文献及观看技术视频。

实习经历

腾讯 - CSIG - TRTC - 音视频 sdk (2025.06 -)

- 核心职责:** 负责腾讯实时音视频 (TRTC) **回声检测 (Echo Detection)** 的算法优化与工程落地, 旨在提升复杂声学环境下的通话质量。
- 技术优化:** 主导了对 WebRTC 开源方案的改进, 并结合业务场景进行参数深度调优与工程化改造。
- 量化成果:** 成功实现回声精准识别, **精准率接近 100%, 召回率达 80%**, 显著改善了用户体验。
- 效率提升:** 为加速迭代, 独立设计并搭建了一套自动化测试评估框架, 支持测试数据自动生成、拼接及量化报告输出, 大幅度缩减了算法验证周期。

网易 - 伏羲 - 后端系统开发 - (2023.06 - 2023.09)

- 核心职责:** 负责面向智能体编程 (AOP) 框架的核心组件——**自研序列化框架 DDL** 的开发与性能优化。
- 功能开发:** 扩展 DDL 模块的功能, 新增对多种复杂数据类型的支持同时, 采用**类型驱动开发**的思想, 为项目引入 type hints, 结合 mypy 进行静态类型检查, 有效提升了代码的可读性、可维护性和类型安全性, 减少了运行时错误的发生概率。
- 性能分析:** 开发性能分析脚本, 利用火焰图进行性能分析, 定位了关键性能瓶颈。
- 完善测试:** 采用**测试驱动开发 (TDD)** 的思想, 基于 **Property-Based Random Testing** 技术, 完善了 DDL 模块的测试, 将覆盖率提升至 **90%**。
- 性能优化:** 采用了多种优化方法提升了 DDL 的性能。在优化之前 DDL 序列化平均耗时在 protobuf 的**50倍**左右, 优化后在大部分典型场景下性能与原生 protobuf 相差在**2倍**以内, 部分场景为 protobuf 的**四分之一**不到。

项目经历

求是社区 - (2025.04 - 2025.07)

主导设计并开发了一个面向哲学与社会学爱好者的前后端分离社区平台，创新性地引入 **AI 虚拟哲学家** 参与深度讨论。作为技术负责人，采用 **ADD (Architecture Driven Design)** 与 **DDD (Domain-Driven Design)** 的思想指导整体架构设计与实现。

核心技术:

- 利用 **RabbitMQ** 消息队列将用户评论、点赞等高频操作进行异步解耦，有效提升系统并发处理能力与响应速度。
- 搭建 **Redis 集群** 与 **数据库读写分离** 实现服务的高可用与高性能；并利用 **Redis ZSET** 高效实现了 **用户活跃度实时排行** 功能。
- 采用 **自适应锁策略** 优化缓存架构，针对热点 key 的并发访问进行串行同步，以防止其失效时导致的缓存击穿。
- 实施“先写 MySQL，再删除 Redis 缓存”的策略，在高并发场景下有效保证了 **缓存与数据库的一致性**。
- 结合 **RAG** 技术，引入 **AI 虚拟哲学家**，使其能理解并参与用户的哲学讨论，极大丰富了社区互动体验。

Transformer-LLM - (2025.03 - 2025.04)

- 实现了一个基于 **Transformer** 架构的 **大语言模型**，并在此基础上进行性能优化以及微调，对齐工作。
- 深入研究并实践了 LLM 的性能剖析 (**Profiling**) 以及 **性能优化** 技术。
- 通过该项目，掌握了 LLM 从模型设计、训练、优化到部署的 **端到端流程和关键技术**。

SysY-RISCV 编译器 - (2024.07 - 2024.09)

- 使用 **C++17** 完成了一个 **高性能编译器**，其生成的代码性能表现达到 **GCC O2 水准**。
- 基于 **SSA IR** 实现了多种 **编译器优化技术**，如 **死代码消除**、**常量折叠**、**循环优化**，**寄存器分配**，显著提升了目标代码执行效率。
- 负责编译器后端开发并担任 **组长**，锻炼了团队协作和领导能力，加深了对编程语言的认识。

miniOS 操作系统 - (2022.03 - 2022.06)

- 使用 **C语言** 实现了一个 **支持多处理器的操作系统**。
- 在实现 **内存分配器** 的过程中，分别实现了基于 **链表的**，基于 **红黑树的**，和基于 **slab** 的内存分配器，深刻理解了不同内存管理策略的性能权衡，并将其融入到“**Fast Path, Slow Path**”相结合的系统设计原则中。
- 在实现内核多线程的过程中，深入理解了 **并发编程的基本理论**，认识到在并发编程中“**防御式编程**”的重要性。