👨‍💻Yongzhen Zhao

(+86) 181-6810-0075 | zhaoyzzz@outlook.com

Date of Birth: March 22, 2002 | Place of Origin: Qingyang, Gansu, China

Education

Nanjing University – Master of Engineering – Software Engineering (2024.09–2026.07) (Recommended Admission)

Nanjing University – Bachelor of Engineering – Software Engineering (2020.09–2024.07)

Skills

- **Fundamentals**: Proficient in foundational knowledge of computer networks, operating systems, and compiler principles.
- **Programming Languages**: Proficient in **Java**, **Python**, and **C++**.
- **Backend Frameworks**: Familiar with **SpringBoot** and capable of quickly learning other Web frameworks.
- **Technical Sharing**: Shares technical insights on my blog at https://rzyn2020.github.io/.
- **Language Proficiency**: English (CET-6), capable of proficiently reading English documentation and books, and watching English technical videos.

Internship Experience

Tencent – CSIG – TRTC – Audio/Video SDK (2025.06 – Present)

Tencent Real-Time Communication (TRTC) is a core real-time audio/video communication PaaS product provided by Tencent Cloud. I am primarily responsible for audio/video SDK development.

NetEase – Fuxi – Backend System Development (2023.06 – 2023.09)

AOP (Agent-Oriented Programming) is a new programming paradigm designed by NetEase Fuxi, allowing users to invoke agent services through this framework. I was primarily responsible for developing the DDL module of the project's internal serialization framework. I adopted **Test-Driven Development (TDD)** and **Type-Driven Development** to ensure high code quality and maintainability. My main responsibilities included:

- **Feature Development**: Extended the DDL module's functionality to support various complex data types. Additionally, introduced type hints to the project and integrated **mypy** for static type checking, effectively improving code readability, maintainability, and type safety, while reducing runtime errors.

- **Performance Analysis**: Developed performance analysis scripts and utilized **flame graphs** to pinpoint key performance bottlenecks.
- **Test Enhancement**: Improved testing for the DDL module based on **Property-Based Random Testing** principles, increasing test coverage to **90%**.
- **Performance Optimization**: Employed various optimization methods to enhance DDL's performance. Prior to optimization, DDL serialization averaged **50 times slower** than protobuf; after optimization, performance was within **2 times** of native protobuf in most typical scenarios, and less than **one-quarter** in some specific scenarios.

Projects

Qiushi Community (2025.04 – 2025.07)

Developed a full-stack, decoupled community platform for philosophy and sociology enthusiasts, integrating **AI virtual philosophers** for in-depth discussions. I was responsible for technology selection and overall framework design, applying **ADD (Architecture Driven Design)** and **DDD (Domain-Driven Design)** methodologies for architectural design and feature implementation.

Tech Stack: Spring Boot, MyBatis-Plus, MySQL, Redis, MongoDB, Elasticsearch, RabbitMQ, Docker

Core Technologies:

- Utilized **RabbitMQ** to asynchronously decouple user comments, likes, collections, and system messages, significantly enhancing system **concurrent processing capability**.
- Implemented **real-time user activity ranking** using **Redis ZSET**, employing a "write to MySQL first, then delete Redis cache" strategy to effectively ensure **cache consistency** in high-concurrency scenarios.
- Designed and implemented a **distributed ID generation scheme** based on the **Snowflake algorithm** to ensure business ID uniqueness and traceability in high-concurrency scenarios, significantly **reducing ID generation latency**.
- Integrated **RAG** and **Agent** technologies to introduce **AI virtual philosophers**, enabling them to understand and participate in user philosophical discussions, greatly enriching the community interaction experience.

Transformer-LLM (2025.03 – 2025.04)

- Implemented a **Large Language Model** based on the **Transformer architecture**, followed by performance optimization, fine-tuning, and alignment.
- Conducted in-depth research and practiced LLM **profiling** and **performance optimization techniques**.
- Through this project, gained a comprehensive understanding of the **end-to-end process and key technologies** for LLM from model design, training, optimization, to deployment.

SysY-RISCV Compiler (2024.07 – 2024.09)

- Developed a **high-performance compiler** using **C++17**, with generated code performance reaching **GCC O2** levels.
- Implemented various **compiler optimization techniques** based on **SSA IR**, including **dead code elimination, constant folding, loop optimization, and register allocation**, significantly improving target code execution efficiency.
- Responsible for backend compiler development and served as **team leader**, enhancing teamwork and leadership skills, and deepening understanding of programming languages.

miniOS Operating System (2022.03 – 2022.06)

- Implemented a **multi-processor supported operating system** using **C language**.
- Developed **memory allocators** based on **linked lists, red-black trees, and slab allocation**, gaining a profound understanding of the performance trade-offs of different memory management strategies and integrating them into the "**Fast Path, Slow Path**" system design principle.
- Gained a deep understanding of **fundamental concurrency theories** and recognized the importance of **"defensive programming"** in concurrent programming during kernel multithreading implementation.