



(+86) 181-6810-0075 | [zhaoyzzz@outlook.com](mailto:zhaoyzzz@outlook.com)

生年月日: 2002年3月22日 | 出身地: 中国甘肅省慶陽市



## 学歴

南京大学 - 修士 - ソフトウェア工学専攻 (2024年9月 - 2026年7月) (推薦入学)

南京大学 - 学士 - ソフトウェア工学専攻 (2020年9月 - 2024年7月)

## 専門スキル

- **基礎知識**: コンピューターネットワーク、オペレーティングシステム、コンパイラ原理などの基礎知識に精通。
- **プログラミング言語**: Java、Python、C++に精通。
- **バックエンドフレームワーク**: SpringBootに精通。その他のWebフレームワークも迅速に習得可能。
- **技術共有**: ブログ <https://rzyn2020.github.io/> にて技術情報を共有。
- **語学力**: 英語 (CET6)。英語のドキュメントや書籍を熟読し、英語の技術動画を視聴可能。

## インターンシップ経験

テンセント (Tencent) - CSIG - TRTC - 音声・動画SDK (2025年6月 - 現在)

テンセントリアルタイムコミュニケーション (Tencent Real-Time Communication、略称TRTC) は、テンセントクラウドが提供する中核的なリアルタイム音声・動画通信PaaS製品です。私は主に音声・動画SDKの開発を担当しています。

ネットイース (NetEase) - 伏羲 (Fuxi) - バックエンドシステム開発 (2023年6月 - 2023年9月)

AOP (Agent-Oriented Programming、エージェント指向プログラミング) は、ネットイース伏羲が設計した全く新しいプログラミングパラダイムであり、ユーザーはこのフレームワークを通じてエージェントサービスを呼び出すことができます。私は主にプロジェクト内で使用されるシリアライゼーションフレームワークDDLモジュールの開発を担当し、**テスト駆動開発 (TDD)** や**型駆動開発**などの技術を採用して、コードの高品質と保守性を確保しました。主な業務内容は以下の通りです。

- **機能開発**: DDLモジュールの機能を拡張し、様々な複雑なデータ型への対応を追加しました。同時に、プロジェクトに型ヒントを導入し、**mypy**と連携して静的型チェックを行うことで、コードの可読性、保守性、型安全性を効果的に向上させ、実行時エラーの発生確率を低減しました。

- **性能分析**: 性能分析スクリプトを開発し、**フレームグラフ**を利用して性能分析を行い、主要な性能ボトルネックを特定しました。
- **テスト改善**: **Property-Based Random Testing**の思想に基づき、DDLモジュールのテストを改善し、カバレッジを**90%**に向上させました。
- **性能最適化**: 様々な最適化手法を採用してDDLの性能を向上させました。最適化前はDDLのシリアライゼーションの平均処理時間はprotobufの約50倍でしたが、最適化後はほとんどの典型的なシナリオでネイティブprotobufと2倍以内の差となり、一部のシナリオではprotobufの4分の1以下となりました。

## プロジェクト経験

### 求是コミュニティ (2025年4月 – 2025年7月)

哲学・社会学愛好家向けのフロントエンド・バックエンド分離型コミュニティプラットフォームを実装し、深層議論に参加する**AIバーチャル哲学者**を導入しました。私はプロジェクトの技術選定と全体フレームワークの構築を担当し、**ADD (Architecture Driven Design)** と **DDD (Domain-Driven Design)** 手法を用いてアーキテクチャ設計と機能実装を行いました。

**技術スタック**: Spring Boot、MyBatis-Plus、MySQL、Redis、MongoDB、Elasticsearch、RabbitMQ、Docker

#### コア技術:

- **RabbitMQ**を活用してユーザーコメント、いいね、お気に入り、システムメッセージを非同期的に分離し、システムの**並行処理能力**を効果的に向上させました。
- **Redis ZSET**を通じて**ユーザーアクティビティのリアルタイムランキング**を実装し、「MySQLにまず書き込み、その後Redisキャッシュを削除する」戦略を採用することで、高並行シナリオにおける**キャッシュの一貫性**を効果的に保証しました。
- コミュニティの高並行シナリオにおけるビジネスIDの一貫性と追跡可能性を満たすため、**Snowflakeアルゴリズム**に基づいた**分散ID生成スキーム**を設計・実装し、ID生成の遅延を大幅に削減しました。
- **RAG**と**Agent**技術を組み合わせ、**AIバーチャル哲学者**を導入しました。これにより、AIがユーザーの哲学的な議論を理解し参加できるようになり、コミュニティのインタラクション体験が大幅に豊かになりました。

### Transformer-LLM (2024年3月 – 2025年4月)

- **Transformerアーキテクチャ**に基づいた**大規模言語モデル**を実装し、その上で性能最適化、ファインチューニング、アライメント作業を行いました。
- LLMの**プロファイリング**と**性能最適化技術**について深く研究し、実践しました。
- このプロジェクトを通じて、モデル設計、トレーニング、最適化からデプロイメントまでのLLMの**エンドツーエンドのプロセスと主要技術**を習得しました。

### SysY-RISCV コンパイラ (2024年7月 – 2024年9月)

- **C++17**を使用して**高性能コンパイラ**を完成させ、生成されたコードの性能は**GCC O2レベル**に達しました。
- **SSA IR**に基づき、**デッドコード削除**、**定数畳み込み**、**ループ最適化**、**レジスタ割り当て**など、様々な**コンパイラ最適化技術**を実装し、ターゲットコードの実行効率を大幅に向上させました。
- コンパイラのバックエンド開発を担当し、**チームリーダー**を務め、チームワークとリーダーシップ能力を養い、プログラミング言語への理解を深めました。

### miniOS オペレーティングシステム (2022年3月 – 2022年6月)

- **C言語**を使用して**マルチプロセッサ対応のオペレーティングシステム**を実装しました。

- メモリ管理の実装過程で、**リンクリスト**、**赤黒木**、**スラブ**に基づいたメモリ管理をそれぞれ実装し、異なるメモリ管理戦略の性能トレードオフを深く理解し、「Fast Path, Slow Path」を組み合わせたシステム設計原則に組み込みました。
- カーネルマルチスレッドの実装過程で、**並行プログラミングの基本的な理論**を深く理解し、並行プログラミングにおける「**防御的プログラミング**」の重要性を認識しました。