Wang, Fei

Tel: (+36)203130453

Email: job@wangfei.dev

Github: github.com/bytemaker-io

Linkedin: linkedin.com/in/fei-wang-113592270/

Location: Budapest, Hungary

Education Background

- **University**: Budapest University of Technology and Economics
- **Degree**: MSc in Computer Engineering
- **GPA**: 4.56/5

2022-09-01 - 2024-07-01

- **University**: Shangluo University
- **Degree**: BSc in Computer Science and Technology
- **GPA**: 82/100

2018-09-01 - 2022-07-01

Professional Skills

- **System Design**: Capable of grasping the overall architecture design of systems, with the ability to design highly available and scalable systems.
- **Performance Optimization**: Able to perform performance analysis and optimization of systems, solving system bottlenecks.
- **Distributed Systems**: Mastery of fundamental principles and common technologies of distributed systems, capable of designing and implementing various components of distributed systems.

Technical Skills

- **Programming Languages & Frameworks**
  - **Core Languages**: Java (Spring Boot/Cloud), Rust, Golang
  - **Frontend**: React, Next.js, TailwindCSS
- **Cloud Native & DevOps**
  - **Containerization**: Docker, Kubernetes
  - **CI/CD**: Jenkins, GitLab CI
  - **Service Mesh**: Istio
  - **Observability**:
    - Distributed Tracing: OpenTelemetry, Jaeger
    - Monitoring: OpenTelemetry, Prometheus, Grafana
    - Logging: OpenTelemetry, Elasticsearch, Kibana
- **Database & Storage**
  - **Relational**: MySQL, TiDB
  - **NoSQL**: Redis, Elasticsearch
- **Message Queue**: RocketMQ, Kafka

- **Core Competencies**
  - System Design & Architecture
  - Performance Optimization
  - High Availability Design
  - Microservices Architecture
  - API Design & Development

Project Experience

Open Source Project: Snowflake ID Rust Implementation

Duration: January 2024 - February 2024

Project Description: Implemented a high-performance distributed unique identifier generator based on Twitter's Snowflake algorithm, designed for large-scale distributed systems requiring globally unique IDs.

Project Link: github.com/bytemaker-io/snowflake-rs

Technical Details:

1. **High Performance Architecture**:
   - Achieved 410K+ IDs/second generation rate through lock-free design (CAS)
   - Implemented atomic operations for thread-safe timestamp and sequence management
   - Optimized memory usage with efficient bit manipulation techniques
2. **Reliability & Safety Features**:
   - Implemented comprehensive error handling for system clock drift
   - Built-in protection against sequence overflow and node ID conflicts
   - Designed fail-fast mechanism for system anomalies
3. **Testing & Quality Assurance**:
   - Developed extensive test suite covering unit tests and concurrent scenarios
   - Implemented performance benchmarks using Criterion framework
   - Achieved 100% thread-safety verification through stress testing
4. **Technical Highlights**:
   - Leveraged Rust's ownership system for memory safety
   - Utilized atomic primitives for lock-free concurrent operations
   - Implemented efficient bit-wise operations for ID composition
   - Designed modular architecture for easy integration and maintenance

E-Food Full Stack Developer

Duration: September 2023 - December 2023

Project Description: An in-restaurant self-service ordering system developed using microservices architecture. The front-end uses React, TailwindCSS, and Next.js, while the back-end is built with Java.

Key Technical Achievements:

1. **Intelligent Food Recommendation**:
   - Integrated Elasticsearch with ChatGPT for intelligent food search and recommendations
   - Implemented speech-to-text functionality for natural language ordering

2. **Authentication & Authorization**:
    - Implemented SSO using OAuth2 with social login support
    - Utilized Casbin for fine-grained RBAC access control
    - Istio external authentication for authentication and authorization Service
3. **High-Performance Architecture**:
    - Leveraged Next.js SSR for optimal page load performance
    - Implemented distributed WebSocket service with consistent hashing for real-time order notifications
    - Used Redis for caching and distributed locking
4. **Reliable Transaction Processing**:
    - Implemented distributed transactions using RocketMQ and Seata
    - Ensured data consistency across microservices
5. **Cloud-Native Deployment**:
    - Containerized with Docker and orchestrated using Kubernetes
    - Implemented service mesh with Istio for service governance
    - Built automated CI/CD pipeline using Jenkins and GitLab

Publications

Title: An Approach to Formal Verification of Atomic Swap Protocols

Conference Title: 2nd Workshop on Intelligent Infocommunication Networks, Systems and Services, 2024

Date: February 5, 2024

Author: Wang Fei

DOI: https://doi.org/10.3311/WINS2024-010