

# Junhui Zhu

daniel.zhujunhui@gmail.com | [github.com/daniel-junhui](https://github.com/daniel-junhui) | [jzhu.xyz](https://jzhu.xyz) | [linkedin.com/in/junhui-zhu-93059a181/](https://linkedin.com/in/junhui-zhu-93059a181/)

## Work Experience

---

### Antelope Technology

2023/03 – Now

Full-time Software Engineer

Singapore

### Antelope Technology

2023/03 – 2023/09

Software Engineer Intern

Singapore

- Developed market data feeder, distributing market data to the corresponding IPC sockets using ZeroMQ for consumption by trading programs. Developed trading gateway that received trading decisions through sockets and directly interacted with the exchange via WebSocket or REST protocols.
- Attempted to do microsecond optimization of the internal system, including the introduction of a new json parsing library and using ringbuffer for market data feeding.
- Daily O&M on-call and monitoring scripting.

### Undisclosed hedge fund

2022/07 – 2023/01

Software Engineer Intern

Singapore

- Developed a service based on Elasticsearch, React and Django for internal Q&A to facilitate internal communication among company.
- Built a market data parser & feeder for TMX exchange, which is a stream processing class for market data, distributing data for different events (trades or quotes) to various receivers through existing C++ interfaces. Enabled traders to directly read internal data formats from selected captured data packets and gained familiarity with the TMX eXtreme Message Transfer (XMT) protocol.

### BrightRidge Investments

2021/09 – 2022/03

Software Engineer Intern

Shanghai, China

- Built trading gateway, market data feeder and docked with strategy programs with Node and Cpp addons to trade on a new exchange.
- Implemented serialization and unserialization of trading signal forecaster with Metaclass in Python to persist the signal library at the Python class level.
- Implemented alerting mechanism detecting the improper resource usage of clusters with InfluxDB and Grafana as well as catching error logs with Loki and Prometheus, making the trading system more robust and respond faster to unexpected situations.

### Meituan, Ltd.

2021/06 – 2021/09

Software Engineer Intern

Beijing, China

- Traced call stack of slow RPCs and explored speed bottlenecks. After refactoring or rewriting, average response time sped from 500ms to 50ms.
- Cooperated with PMs and frontends, responded to user requirements, read PRDs and wrote backend logics.

## Education

---

### National University of Singapore

2022/08 – 2024/05 (Expected)

Master of Computing in Computer Science, GPA 4.86/5

Singapore

- Courses: Distributed Systems, Advanced Operating Systems, Systems Security

### Shanghai Jiao Tong University

2018/09 – 2022/06

B. Eng in Industrial Engineering

Shanghai, China

- Courses: Stochastic Process, Introduction to Computer Systems, Operations Research, Real Analysis

## Publications

---

### DeltaBoost: Gradient Boosting Decision Trees with Efficient Machine Unlearning

ACM SIGMOD 2023

Zhaomin Wu, Junhui Zhu, Qinbin Li, Bingsheng He

## Projects

---

Toy FAT Command-line Tool [https://github.com/daniel-junhui/fat\\_tool](https://github.com/daniel-junhui/fat_tool)

- Command binary that can manipulate FAT image files.
- Ability to review the format of FAT files, list all file contents in the file system, copy files from other file systems into an image file, or copy files from an image file to other file systems.

#### **Multi-Paxos KV Store** Closed source as requested

- Mapping keys to different partitions (mapping is defined during compiling time), with each partition being managed by a Paxos cluster.
- Allocation of partitions to different clusters is managed by a Paxos cluster, which handles the joining of new clusters and the departure of old clusters.

#### **Raft Library in Go**

- Lab project of MIT 6.824 Distributed Systems.
- Used Golang RPC lib and Goroutines to implement a lightweight Raft library and passed the tests provided by MIT for over 1000 times;

#### **TCP Library in C++11** <https://github.com/daniel-junhui/sponge>

- Lab project of Stanford CS144 Computer Networks.
- Decoupled the state of the entire TCP connection into a tuple of TCP sender and TCP receiver states, resulting in a more elegant implementation of state transfer

#### **Compiler Frontend for Simplified C Language** <https://github.com/daniel-junhui/Frontend-for-weakened-C>

- Simplified C language by fixing the size of variables.
- Directly translated the AST into x86\_64 assembly code, which could be assembled to ELF by gcc or clang.

### **Skills**

---

**Programming Languages:** C++, Python, Node.js, Java, Go

**Tech Skills:** Distributed Systems, Docker, P4 (programmable switch language)

**Languages:** English, Mandarin Chinese, Shanghainese