XUANWEI ZHANG

(+86)159-2275-8321 loloxwg@gmail.com loloxwg.top github.com/loloxwg

EDUCATION

Chongqing University of Posts and Telecommunications *September 2020 - June 2022*
Bachelor in Internet of Things Engineering(IoT), School of Automation *Chongqing, China*
- **Awards**: Mathematical Contest in Modeling(first prize)
- **Activities**: Participated in the testing and maintenance of the red-rock(programming org) school website with a DAU of 5000+ users.
- **Core Courses**: *Discrete Mathematics*, *Data Structures and Algorithms Design*, *Object Oriented Programming*, *Database Systems*, *Software engineering*, *Industrial Big Data Technologies*, *System and Embedded application*.

Nanchang Hangkong University(NCHU) *September 2016 - June 2020*
Bachelor in Flight Vehicle Propulsion Engineering, School of Aircraft Engineering *Jiangxi, China*
- **Awards**: College Merit Scholarship, Invention Patent.
- **Core Courses**: *Linear Algebra*, *C Language*, *Mathematical Culture*, *Network Engineering*, *Theoretical Mechanics*.

EXPERIENCE

SenseTime *August 2021 - Present*
Software Engineer Intern - Data Storage Team *Shanghai, China*
- Design and develop backend services of the storage management system for the internal datasets using Golang, Gin, and Gorm.
- Responsible for packaging FSclient, writing tool modules, and processing data gateways.Configure Ceph OSS lifecycle permissions.
- Create various file management modules for the FS data gateway to ensure the stability and reliability of file operations.
- Package Docker images and deploy on servers. Use Kubectl to manage containers. Write unit test and regression tests. Design microservices that are scalable and secure.

Industrial Internet Data Storage System, School of Automation *January 2022 - Present*
Research Assistant/Project Lead *Chongqing, China*
- Design an IoT MQTT message distribution and storage system based on micro-services for school research project. Used emqx rules engine for processing data into the Kafka message queue through kong gateway reverse proxy.
- Implement micro-services to expose external APIs interface for different storage engines. Use RPC for internal communication and ensure structured data and unstructured data persistence. Use Grafana for data visualization and monitoring.

PROJECTS

Percolator distributed transaction model in TinyKV *April 2022 - Present*
- Implement an improved version of 2PC transaction model to solve the synchronization blocking problem of 2PC based on Percolator.
- Develop transaction models based on Percolator atomic commit with Consensus and 2PC algorithms to ensure the availability and consistency of decision logs.
- Realize isolation-level snap shot, coordinator stateless node to solve synchronous blocking, lazy lock cleaning and TSO clock timestamp with core fucntions such as DB transaction commit, ResolveLock, and CheckTxnStatus.

C++ based concurrency control DB for SQL operations *December 2020 - February 2022*
- Implemented LRU-based buffer pool to reduce random disk IO. Maintained multiple instances to reduce global lock contention.
- Realized extendible hashing, a dynamically updateable disk-based index structure unlike the static hashing and SQL query operators like *aggregation*, *hash_join*, *delete*, *distinct*, *insert*, *limit*, *seq_scan* and *update*.
- Used *would_wait* scheme for deadlock prevention. Developed transaction isolation levels like *READ UNCOMMITTED*, *READ COMMITTED* and *REPEATABLE READ*.

Wi-Fi Positioning System(IPS) in Java *May - June 2021*
- Designed and developed a Wi-fi indoor positioning system (IPS) using Java, Spring Boot and MyBatis to obtain the size and strength of Wi-Fi signals through client-side acquisition.
- Developed signal/user/mapping and location modules. Wrote REST API design docs and visualized result with GUI.

MIT6.824 Fault-tolerant K-V storage using Raft consensus algorithm *2020 - 2021*
- Implemented abstractions and implementation techniques for engineering distributed systems such as fault tolerance, replication, and consistency. Built a K-V storage system with Raft using Golang.

TECHNICAL SKILLS

Programming Languages:	Golang, C++, Python, C, Java, JavaScript
Web & Frameworks:	Spring Boot, Gin, Node.js, HTML, CSS, Grafana, MyBatis
Data & Tools:	Git, Docker, ETCD, MySQL, InfluxDB, Redis, AWS S3, Kafka