

# Lilian Lin

672-855-0803 | [lin.yins@northeastern.edu](mailto:lin.yins@northeastern.edu) | [www.linkedin.com/in/lilian-lin003](https://www.linkedin.com/in/lilian-lin003) | <https://github.com/ln541758>

## EDUCATION

### Northeastern University

*Master of Science in Computer Science, GPA: 3.95*

Vancouver, Canada

September 2023 - December 2025

### University of Hawaii at Manoa

*Doctor of Philosophy in Architecture (Computer-Aided Design, HCI, VR, Wayfinding)*

Honolulu, USA

September 2019 - May 2023

## EXPERIENCE

### Software Development Engineer Intern

May 2024 – August 2024

*Jiachen Machinery Technology Co. | React, JPA, Redis, MySQL, AWS, Jenkins, CI/CD*

*Shanghai, China*

- Developed an internal forum system for a mechanical enterprise, with the frontend built using the React framework. Integrated the Element UI component library to create an aesthetically pleasing and user-friendly interface. The system supports user management, posting and commenting, and article search functionalities.
- Utilized Spring Data JPA to simplify database operation, and based on Redis cache user information, product information and other hotspot data to reduce the pressure on MySQL query, the average query time is reduced from 500 milliseconds to 40 milliseconds.
- Implemented a data storage solution to migrate image and video data to AWS S3, significantly reducing storage costs by 70%, and adopted data lifecycle policy and version control to automate data management.
- Automated testing and deployment tasks in real-time using Jenkins based on CI/CD and achieved 95% unit test coverage.

## PROJECTS

### GameHub | SpringBoot, React, Node.js, AWS, Redis, Kafka

Jan 2024 – April 2024

- Built a high-performance gaming web platform with a backend built on SpringBoot, MyBatis, Redis, AWS RDS, and Kafka, and a frontend built with React and Node.js.
- Supports up to 5,000 concurrent users, handles over 3,000 transactions per second, and maintains a P99 latency of less than 1 second.
- Implemented database sharing and table partitioning using Amazon RDS to effectively model complex user data, game resources, and transaction records, and enhanced query performance by using indexing and partitioning techniques.
- Cache data in Redis, reducing query latency from an average of 200ms to an average of 30ms. Configure TTL to manage data expiration.
- Optimize Kafka partition and replica configuration, optimize message asynchronous processing speed and traffic peak elimination, and increase throughput from 300 messages per second to 1,100 messages.

### Distributed KV Storage System | Raft, Redis, RocksDB, ReadIndex

September 2024 – Present

- Built a high-performance system handling 8,000+ ops/sec with sub-millisecond latency and 99.99% uptime using Raft consensus.
- Implemented the Raft consensus algorithm, with core functions such as Leader election, log replication, and snapshot update.
- Based on the consistent hashing architecture, the data is partitioned into Shards and can be migrated in multiple Raft Groups.
- Implement support for storage engines such as RocksDB, B-tree, and hash tables to adapt to scenarios with different IO models.
- Enhanced system performance by 40% and efficiency by 35% using optimizations including Asynchronous Apply, ReadIndex, FollowerRead, and Prevote techniques.

## TECHNICAL SKILLS

**Programming Languages:** Python, Java, JavaScript, TypeScript, C, C#, Shell

**Frontend Technologies:** HTML, CSS, React, Node.js, React Native, DOM

**Backend Technologies:** Flask, Spring Boot, REST APIs

**Databases:** MySQL, MongoDB, Firebase, Redis

**Tools:** Git, AWS, Linux, Maven, JVM