

# SHUNING(SCOTT) LIN

Phone: 878-999-6296 | Email: [shuninglin5@gmail.com](mailto:shuninglin5@gmail.com) | GitHub: <https://github.com/ScottLinnn>

LinkedIn: <https://www.linkedin.com/in/shuning-lin-scott/>

## EDUCATION

### Carnegie Mellon University

Master of Science in Information Networking (MSIN)

Pittsburgh, PA

09/2023 – 12/2024

Courses: Storage Systems, Advanced OS and Distributed Systems, Mobile and Pervasive Computing

### Carnegie Mellon University

Master of Advanced Architectural Design

Pittsburgh, PA

09/2021 – 05/2023

Courses(elective): Database Systems, Computer Networks, Intro to ML, Distributed Systems, Software Engineering

### University of Nottingham

China

Bachelor of Engineering, Architecture

09/2017 – 06/2021

## SKILLS

Programming Languages: C/C++, Java/Kotlin, Go, Rust, Python, Shell, HTML, CSS, JavaScript

Others: Linux, Git, GitLab, CI/CD, Kafka, Docker, Kubernetes, Swagger, PostgreSQL, AWS, GCP, Azure, MySQL, Cassandra, GDB, RPC, DynamoDB, S3, Android Dev/Android Studio, OpenCV, TCP/IP/HTTP

## EXPERIENCE

### Software Engineer Intern

Beijing, China

MonographDB

06/2023 – 08/2023

- MonographDB builds a generalizable memory layer for compute-memory-storage disaggregated cloud databases.
- Built a distributed testing framework in **Python** to replace the MySQL test framework, enabled the company to **both write and run tests more effectively** and to access previously inaccessible states to uncover more bugs.
- Implemented cluster management mechanism, utility library and rich control (**8 command args**) to make the framework **highly usable and extensible**. It is now used by other developers who keep adding test cases upon it.
- Implemented test cases for partition and failover scenarios, **caught 4 bugs** which were not found by old tests.
- Deployed **5 ConcourseCI** pipelines with the new framework and new tests, providing stricter checks for every PR.

### Teaching Assistant

Pittsburgh, PA

CMU School of Computer Science

08/2022 – 12/2022

- TA'd **Distributed Systems** course, graded works/exams and answered questions on Piazza for **270 students**.
- Held Office Hours to assist students on projects in **Golang** (e.g. **Raft**, **MapReduce**, **Message Queue**).

### Software Engineer Intern

Pittsburgh, PA

Mohimani Lab, CMU School of Computer Science

04/2022 – 08/2022

- Developed new **RESTful** microservice endpoints in **Rust** and **Axum** web framework, enabled users to fragment molecules with customized algorithms, upload more types of data files and run expensive biology task remotely.
- Leveraged **Docker** and **PostgreSQL** to fully test the added services in local environment before submitting PR.
- Used **Swagger** to produce API doc and **automatic endpoint tests** for 100% of added services.

## PROJECTS

### DBMS Implementation (C++)

- Developed a buffer pool manager which enabled user to choose when to flush in-memory file pages to better provide ACID properties for transactions, with the **LRU-K** replacement policy which **addressed the ineffectiveness of LRU under sequential scan workload** which is common for database.
- Built a B+ Tree to offer  $O(\log n)$  search time +  $O(1)$  I/O number for index lookups, with latches to support **concurrent read/write/delete** and latch crabbing to **reduce lock contention by 42%**.
- Implemented **10+ query operators** to support common query patterns, with Hash Join and Top-K optimization.
- Designed and implemented a lock manager to support **4 isolation levels** to allow user to trade between consistency and efficiency, and **5 types of locks** to trade between concurrency and lock acquisition overhead.

### CloudFS (C++)

- Built a hybrid local-cloud file system, which combined the extensibility of cloud storage and low cost/fast performance of SSD by storing large files on cloud and small files on SSD. It implemented **22 file system calls**.
- Leveraged local proxy and **data deduplication** on large files to reduce **cloud cost by ~62%**.
- Supported **snapshot** that enabled user to restore file system to a previous state or browse an installed snapshot.