Yuqing Qiu

+17346043813 :: yuqingqi@andrew.cmu.edu :: linkedin.com/in/yuqing-q/ :: elena-qiu.github.io/

EDUCATION

Carnegie Mellon University - School of Computer Science Pittsburgh, Pennsylvania
M.S. in Computational Data Science, System Track | *GPA:* 4.00/4.00 *Dec. 2023 (expected)*
Selected Courses: Database Systems, Storage Systems, Advanced Cloud Computing, Parallel Computing, Search Engine

University of Michigan Ann Arbor, Michigan
B.S.E in Computer Science, Minor in Math | *GPA:* 3.95/4.00 | *Honors:* Dean's List, University Honors *May 2022*
Selected Courses: Distributed Systems, Operating Systems, Computer Architecture, Compiler Construction, Web Systems, Database Management System, Computer Networks, Computer Vision, Data Structures and Algorithms

Shanghai Jiao Tong University Shanghai, China
B.S.E in Electrical and Computer Engineering | *GPA:* 3.84/4.00 | *Honors:* Academic Excellence Scholarship *Aug. 2022*

SKILLS

Programming languages: C/C++, Python, Java, Golang, Verilog, CUDA, Rust, SQL, MATLAB, JavaScript, HTML/CSS
Frameworks/Tools: Spark, Hadoop, Airflow, Docker, Kubernetes, AWS (EC2, S3, EMR, Athena), PyTorch, Flask, React.js, SQLite3

WORK EXPERIENCE

Apple Inc. New York City, New York
Software Engineering Intern *May 2023 – Aug. 2023*

- Developed a **monitoring and alerting system** for ads platform to enable feature observability of downstream logs. Leveraged **AWS Athena** to query hive tables, extract valuable insights, and seamlessly integrate aggregated values into **Datadog**.
- Engineered a scalable infrastructure using **AWS EMR** and **Docker** for efficient data processing to enhance monitoring cadence.
- Streamlined end-to-end process with a **CI/CD pipeline** and **Airflow orchestration** to optimize workflow management.
- Implemented automated **data quality checks** with prompt anomaly alerts to enable model calibration for better performance.

Intel Corporation Shanghai, China
Software Engineering Intern *May 2021 – Aug. 2021*

- Wrote APIs utilizing **PySpark framework** to support the **distributed cluster servin**g for large-scale recommender systems.
- Optimized DLRM data preprocessing on Twitter dataset by tailoring **Apache Spark** join strategies and cut time from 13h to 1.2h.
- Trained a **recommendation system** from a WeChat dataset of over 10 million video feeds to predict user actions (e.g., likes and comments), achieving 72% accuracy. Applied Wide & Deep, DeepFM, XGBoost model, and AutoML frameworks.
- Deployed sentimental analysis example on **distributed training pipeline** and scaled out from single node to **big data clusters**.

Carnegie Mellon University Parallel Data Lab Pittsburgh, Pennsylvania
Software Engineering Intern (Part-time) *Jan. 2023 – May. 2023*

- Engineered a cost-effective **cloud-edge emulating framework** for testing cloud resources locally to reduce the operational costs.
- Built a streamlined pipeline for launching pods in **Kubernetes** with customized network topology and application deployment.
- Deployed a video surveillance application onto emulator with seamless cloud-edge collaboration for iterative model refining.

PROJECT EXPERIENCE

Advanced Cloud Computing | Carnegie Mellon University *Jan. 2023 – May 2023*

- Accelerated ETL processing and iterative model training using **Spark**, minimizing time by shuffling reduction and caching.
- Designed job scheduling policy in **Kubernetes** to maximize total utility for diverse job traces, utilizing FIFO and SJF strategies.

Parallel Computing | Carnegie Mellon University *Jan. 2023 – May 2023*

- Created a **CUDA** parallel circle renderer, optimizing memory access latency and ensuring atomicity and thread synchronization.
- Developed a parallel n-body simulator with **OpenMP** and **MPI** respectively, minimizing false sharing and workload imbalance.

Distributed and Fault-Tolerant Key/Value Services | University of Michigan *Jan. 2022 – May 2022*

- Built a distributed key/value database in **Golang** with **primary/backup replication** to serve Get, Put, and Append RPC requests.
- Implemented the **Leader-based Paxos Protocol** to solve consensus with tolerance of network unreliability and server failures.
- Sharded the storage system using **consistent hashing** to balance workload across servers and enhance availability and scalability.

Operating Systems Implementation | University of Michigan *Jan. 2021 – May. 2021*

- Implemented a **thread library** for mutex and condition variables in **C++** with FIFO scheduling and interrupt handling.
- Devised an **external pager** for address space and virtual memory management with system call and exception mechanism.
- Engineered a **network file system** for read/write/create/delete requests with multithreading, socket protocols and request caching.