

# Peter (Shengzhou) Qiang

Buffalo, NY · sqiang@buffalo.edu · github.com/Alephant6 · sqiang.us · linkedin.com/in/sqiang

CS PhD student in distributed transactions & databases, seeking Summer 2027 software-engineering / research internships in database internals and infrastructure.

## EDUCATION

**University at Buffalo, SUNY** — Ph.D. in Computer Science (advisor: Prof. Haonan Lu) Aug 2023 – present (exp. 2028)

Research: distributed transactions, consistency protocols, and evaluation of distributed datastores.

**Nanjing University of Information Science & Technology** — B.S., Information and Computing Science 2019 – 2023

GPA 3.8/5.0 (top 5%); industry-embedded program with Neusoft Group.

## RESEARCH PROJECTS

**SONIC** — rethinking replication-critical read-only transactions (**lead author** — latest SOSP submission advanced through two review rounds to PC-meeting discussion) 2025 – present

- Leading a first-author project showing “replication-critical” read-only transaction designs are unnecessary: formalized the SONIC and Remote-Read theorems and built two systems demonstrating them.
- Built **Starry** by deep-modifying UW’s TAPIR codebase (**300+ commits, C++**) and an extended COPS-SNOW with chain replication; validated consistency with self-built **Jepsen falsifier suites** (Clojure); contributed upstream to **Jepsen** (Debian support, CockroachDB test hardening).
- Ran experiments on CloudLab via a self-built Docker-Compose cluster testbed (SSH-CA, scalable controller + N-node topology).

**Juicer** — transparent tail-latency layer for strictly serializable concurrency control (in submission) 2025 – present

- Co-building a proactive message-ordering layer in gRPC-Go interceptors that reorders transaction messages by timestamp before they reach data shards — **cutting P99 latency by up to 75% and aborts by up to 85% with zero database code changes**.
- Implemented concurrency-control protocols (2PL wound-wait/wait-die, MVTO) in a **14K-LoC Go transactional KV testbed**; built its workload generator (Zipfian, dynamic skew, read-modify-write).
- Evaluated on CloudLab (9 × 16-core / 25 Gbps) against NCC (SOSP’23 state of the art): **54% tail-latency reduction**.

**EaaS** — Evaluation-as-a-Service for distributed storage systems 2023 – 2025

- Co-built the first end-to-end evaluation framework for distributed storage (C++, gRPC, Raft-replicated control plane); **top contributor with ≈47% of 1,800+ commits** in a 6-person team.
- Designed the unified workload engine (YCSB, transactional YCSB-T, interactive TPC-C) and C++/Go/Java adapters integrating **CockroachDB, Cassandra, and EPaxos-family protocols** at ≈20 LoC per system.
- Re-evaluated replication protocols across AWS regions, **refuting EPaxos’s published 3× throughput claim (measured 1–1.5×)**; uncovered an EPaxos livelock at 9 replicas.

**PRC** — developer-intended consistency testing for distributed datastores (in submission) 2025 – present

- Directed implementation testing that turns community-known correctness pitfalls (e.g., timestamp inversion) into targeted tests, complementing random-exploration tools like Jepsen; evaluated the P formal-modeling framework.

## EXPERIENCE

**Neusoft Group** — Software Technology Trainee (industry-embedded B.S. program) 2020 – 2023

Built Hadoop-ecosystem pipelines: Hive warehouse, Storm streaming analytics, Spark batch jobs (Java).

## PUBLICATIONS

**S.-Z. Qiang** et al., “Instantaneous cross-correlation function type of WD based LFM signals analysis via output SNR inequality modeling,” *EURASIP J. Adv. Signal Process.*, 2021 (**first author**); 6 further co-authored papers (*Optik* ×3, *Physics Letters A*, *Math. Problems in Eng.*, *Chinese J. Electronics*), 2021–2022.

## SKILLS

**Languages** C++, Go, Python, Java, Clojure, Shell, LaTeX  
**Systems** distributed transactions & consistency (2PL/OCC/MVTO/NCC, Paxos/Raft/EPaxos, TAPIR), CockroachDB internals, gRPC, YCSB/TPC-C benchmarking, Jepsen, P formal modeling  
**Infra** Docker-Compose cluster orchestration (3 self-built SSH-CA testbeds), CloudLab, AWS, Linux, GitHub Actions  
**Other** Kaggle Foursquare Location Matching — top 10% (103/1079)