Shihang (Vic) Li

shli@cs.washington.edu | vic-li.me | github.com/vic-lsh

Research Summary

Ph.D. student in systems and networking, working on LLM agent systems, HW/SW co-design for memory management, and microservice systems. I build systems across the stack—from hardware abstractions to OS kernels and application runtimes.

Education

University of Washington, Seattle, WA Sept. 2023 – Present
Ph.D. in Computer Science. Advisors: Simon Peter, Tom Anderson. GPA: 3.8/4.0

Brown University, Providence, RI Sept. 2021 – May 2023
M.S. in Computer Science. Advisor: Malte Schwarzkopf. GPA: 4.0/4.0

New York University, New York, NY Sept. 2016 – May 2021
B.S. in Computer Science & Business (Double Major). GPA: 3.72/4.0

Research Experience

Microsoft Research Jun. 2025 – Sep. 2025
Research Intern. Mentors: Landon Cox, Pedro Henrique Penna, Rodrigo Fonseca Redmond, WA

FlowGuard: DIFC-based data-leak containment for LLM agents.

- Designed policies that present security-utility trade-offs for agents to reason about (e.g., no internet after seeing secrets).
- Leveraged decentralized information flow control (DIFC) to transparently contain agents and bound secret leakage.
- Built FlowGuard, a Rust DIFC reference monitor that transparently interposes on MCP traffic with no agent code changes.
- Prevented real prompt-injection attacks: demoed FlowGuard blocking OpenAI Codex from leaking GitHub secrets.

NEMO: high-fidelity memory observability via programmable memory controllers Mar. 2024 – Present
with Matthew Giordano, Daniel S. Berger, Tom Anderson, Simon Peter Under submission

A programmable memory controller (MC) that provides high-fidelity memory telemetry, flexibly programmed by the OS.

- Designed a programmable, match-action–style telemetry pipeline that inspects every memory request at the MC w/o sampling.
- Integrated NEMO into Linux and a userspace memory manager via Userfaultfd to manage memory across processes and VMs.
- Reduced detection time by up to 10x and overhead by up to 350x compared to Intel PEBS and MBM.
- Improved performance by up to 1.7× in throughput and 23% lower tail latency across tiering and noisy-neighbor scenarios.

Masa: rethinking microservice SLOs with end-to-end deadlines Mar. 2024 – Present
with Simon Peter, Danyang Zhuo, Arvind Krishnamurthy, Ratul Mahajan

A microservice runtime that enforces e2e RPC SLOs in microservice graphs, instead of per-service SLOs.

- Designed a distributed RPC scheduler that manages RPC slack in e2e SLO across service DAGs, improving goodput.
- Extended gRPC to transparently propagate per-RPC priority across microservice graphs with zero application change.
- Modified the Tokio async runtime (Rust) to support per-task priority and custom scheduling policies (e.g., least-slack-first).
- Improved goodput by 20% on DeathStarBench and diverse Alibaba call graphs via trace replay.

Published Research

Quicksand: Harnessing Stranded Datacenter Resources with Granular Computing NSDI '25 (paper, code)
Z. Ruan, **S. Li**, K. Fan, M. Aguilera, A. Belay, S. Park, M. Schwarzkopf

Minimize stranded datacenter resources with *resource proclets*—small, migratable units that primarily use one resource.

- Designed and implemented compute resource proclets, and evaluated them for GPU training data preprocessing.
- Implemented distributed sharded data structures (vector, queue, unordered/ordered map) for interactive and batch workloads.
- Presented the work at NSDI '25 (talk).

Loom: Efficient Capture and Querying of High-Frequency Telemetry SOSP '25 (paper, code)
F. Solleza, **S. Li**, W. Sun, R. Tang, M. Schwarzkopf, A. Crotty, D. Cohen, N. Tatbul, S. Zdonik

A telemetry collector of high-frequency metrics for observability workloads in production.

- Designed a writer-prioritized reader–writer lock with formal linearizability proof; implemented in Rust.
- Optimized ingestion path for high-frequency data collection (8M+ samples/s) with minimal resources (1 core, 512 MiB).

Skills

- **Systems**: Async & RPC frameworks (Rust, C++), OS kernel (C), HW design (SystemVerilog), Agents (Python, TypeScript).