Haoran Zhang

zhhaoran@umich.edu | zhhr0321

Education

University of Michigan, Ann Arbor August 2024 – May 2026
B.Eng in Computer Science Michigan, United States

- major GPA 3.98/4.0
- A+/A Courses: Intro to Operating System, Compiler, Computer Networks, Cryptography, etc.

Shanghai Jiao Tong University September 2022 – August 2026
B.Eng in Mech Engineering Shanghai, China

- A+/A Courses: Probabilistic Methods in Engineering, Honors Calculus IV, Honors Calculus III, etc.

Research Experience

Agentic-ds-ops MAY. 2025 – PRESENT
Distributed Systems Project Order Lab, Ann Arbor, MI

- Built an **agent-based autonomous mitigation system** for **distributed failures** like overload, network faults on **ZooKeeper** clusters. Experimented with a custom overload benchmark in Go.
- Integrated **Prometheus** metrics with sliding window to dynamically detect metrics trend and **JMX-exporter** to collect logs. Combined for failure detection.
- Designed **risk-aware mitigation framework** where agent selects from pre-defined actions via **Haproxy**, **Resilience4j** based on symptom severity.
- Applied pre- and post-evaluator to quantitatively **predict** and **verify** outcomes against **SLOs/throughput** before concluding mitigation success.

CUDA Graphs for Reducing Kernel Launch Overhead SEP. 2025 – PRESENT
MLSys Research Project Ann Arbor, MI

- Built a hybrid **runtime proxy** reducing **kernel launch overhead** and cut down on **tail latency** in Large recommendation system, MoE inference.
- Applied **CUDA Graphs** for stable, high-arithmetic compute (MLP blocks/attention) and a **persistent kernel** consuming device-queue tasks for irregular micro-ops (pack/scatter/routing).
- Used **bucketing** and **static pools** to capture several robust graphs and achieved higher hit rate of **graph replay**.

LLM-Powered Robotic Manipulation System MAR. 2024 – AUG. 2024
Robotics Research Shanghai Jiao Tong University

- Designed and implemented an end-to-end robotic control pipeline **integrating LLM** (for natural language task planning and code generation) with **SAM-6D**, enabling robotic arms to execute **manipulation tasks** from natural language commands.
- Developed vision-language-action loop: SAM-6D processes RGB-D input for real-time object localization; LLM parses user intent and generates motion primitives; motion planner translates high-level commands to joint trajectories.

Selected Project

Selective Memory Snapshotting for `ptrace` SEPT. 2025 - OCT. 2025

- Extended `ptrace` without changing its ABI by adding `SNAPSHOT`, `RESTORE` and `GETSNAPSHOT` to **capture/restore** a specified writable region of a tracee's address space.
- Validated regions via VMA permission/length checks; stored snapshots in kernel space keyed by (pid, start, len) with caps (`MAX_SNAPSHOT_LEN`, per-tracee `MAX_TOTAL_SNAPSHOT_SIZE`); auto-cleanup on restore and tracee exit.

Simulated Distributed System SEPT. 2025 - Present

- Primary-Backup One-Fault-Tolerance Storage System
  - Implemented with Lexical Confinement design for high-concurrency requests using Go.
- Paxos-based Fault-Tolerant Key/Value Storage System
  - Implemented a replicated key/value store using multi-instance Paxos to totally order Get/Put/Append operations without a central coordinator.
  - Ensured linearizable single-copy semantics and at-most-once execution while tolerating server and network failures, with lagging replicas catching up via the Paxos log.

Network File Server MAR. 2025 – APR. 2025

- Built a concurrent, crash-consistent file system with hierarchical directories, supporting FS_READBLOCK, FS_WRITEBLOCK, FS_CREATE, FS_DELETE over TCP.

- Ensured crash safety via ordered metadata writes; scaled concurrency with Boost threads and reader–writer locks; Built network communication using POSIX sockets for client-server interaction.

Memory Manager (Pager) FEB. 2025 – MAR. 2025

- Designed a multi-process virtual memory pager with swap-backed and file-backed mappings (akin to Unix mmap), per-process page tables, and MMU protection bits.
- Handled vm_map, page faults, fork with copy-on-write, eviction via clock (second chance), zero fill fast paths, and eager swap reservation; preserved sharing for file-backed aliases.

Thread Library JAN. 2025 – FEB. 2025

- Built a user-level thread library, managing CPU booting and thread life cycle; and monitor primitives supporting single and multiCPU execution, preemptive scheduling via timer interrupts.
- Implemented synchronization primitives like mutex, condition variable and spin-lock, using advanced Unix context management techniques.

Compiler Construction - Dynamic Typed Compiler Jan. 2025 - Apr. 2025

- Used Rust to develop a compiler for a simple language supporting dynamic typing and heap allocation on x86-64 architecture.
- Implemented front-end checking, middle-end Single Static Assignment (SSA) forming, and backend code generation with System V ABI.
- Implemented optimizations including register allocation and assertion removal.

Teaching Experience

University of Michigan Ann Arbor, MI

Foundations of Computer Science (EECS376) FA.2025

- Knowledge includes Algorithms; Turing Reduction; P-NP problems; Cryptography
- Held Office Hours weekly.

Shanghai Jiao Tong University Shanghai, China

Honors Calculus III (MATH2550J) SU.2024

- Knowledge includes Linear Algebra; Multiple, Line, Surface integrals; Complex Analysis
- Held Recitation Classes and Office Hours weekly.

Honors Calculus II (MATH1560J) FA. 2023

- Knowledge includes Limits and Continuity; Differentiation; Integration
- Held Recitation Classes and Office Hours weekly. (Materials Attached here)

Skills

Programming Languages: C/C++, Python, Go, Java, Rust,etc.
Distributed System: Docker, Kubernetes, ZooKeeper, Resilience4j, Prometheus, ChaosBlade/ChaosMesh, etc.
Machine Learning: PyTorch.