

ARYAN SINGH

Chicago, IL | s.aryan0328@gmail.com | linkedin.aryansingh.cc
github.aryansingh.cc | aryansingh.cc | U.S. Permanent Resident

SUMMARY

Computer Science and Mathematics student at UIUC with experience in software development, Linux infrastructure, and applied AI. Work includes a centralized LLM gateway, multi-agent AI systems, GPU-optimized AI inference, personal cloud infrastructure, and FPGA accelerator research.

EDUCATION

University of Illinois Urbana-Champaign

Urbana, IL

B.S., Computer Science and Mathematics | GPA: 3.78 / 4.00

Expected Graduation May 2028

- Coursework: Machine Learning, Artificial Intelligence, Parallel Programming (CUDA), Algorithms, Distributed Systems, Computer Architecture, Systems Programming, Compilers, Database Systems
- Honors: 1st of 400, CS 341 Malloc Contest (top-performing memory allocator); 4th place, Advanced Track, UIUC CTF 2026; 3rd place, UIUC CTF 2025; 5th of 50 schools, Robot Tour, UIUC State Science Olympiad 2024

EXPERIENCE

University of Illinois Chicago

Chicago, IL

Undergraduate Researcher | Prof. Debjit Pal's research group

Aug 2025 - Present

- Co-authored [BoostedSOSA](#), a dual-FPGA HPC scheduler replacing unreliable user-provided runtime estimates with ML predictions; paper reports **63.85%** lower error and **17x** speedup.
- Currently implementing two zero-copy, lock-free host-FPGA ring buffers to replace flag polling and reduce PCIe handoff overhead in the continuous scheduling pipeline.

Software Engineering Intern

May 2025 - Aug 2025

- Converted an FPGA scheduler from batch-oriented I/O into a free-running streaming pipeline so HPC jobs could be scheduled as they arrive instead of waiting for batches to fill.
- Built nonblocking Vitis HLS Reader/Writer kernels and multithreaded C++/XRT host I/O; paper measured **<4 us/job** for streaming vs. **245 ms** batching at 100 jobs/s.
- Designed and verified the systolic scheduling datapath, virtual FIFO, and cost calculator with ~1,100 lines of Monte Carlo-driven Vitis HLS tests on Alveo U55C.

SYSTEMS AND SOFTWARE PROJECTS

Advanced Systems Programming (CS 341) | C, Linux/UNIX, POSIX Threads, TCP/IP

- Memory Allocator: Built a C allocator for efficient memory reuse, winning **1st place out of 400 students**; used address-sorted free lists, coalescing, fragmentation-aware splitting, and in-place realloc growth.
- Network Server: Enabled **thousands of concurrent TCP clients** on one thread with a C epoll reactor and length-prefixed binary protocol; per-connection state machines resume partial I/O after EAGAIN without busy-waiting.
- UNIX Shell: Built a POSIX shell in C to run commands and manage UNIX processes, implementing fork/exec, signal handling, I/O redirection, and advanced shell commands.
- Parallel Execution: Parallelized builds and data processing with a multithreaded make clone and local MapReduce; used mutexes and condition variables to coordinate work and prevent deadlocks.

Freeway | TypeScript, Node.js, SQLite

[GitHub link](#)

- Built a self-hosted gateway so applications can use **12 LLM providers** through one OpenAI-compatible API, with no external runtime dependencies and providers added through JSON configuration.
- Implemented **two-tier caching** with exact-request matches and optional semantic search, scoped by model and system prompt; replayed cached responses as SSE without consuming upstream quota.
- Kept conversations continuous during provider failures with four-step context refitting; migrated a **41k-token thread to an 8k-context model** and routed requests by EWMA latency.
- Classified failures before retries and reserved quota at dispatch to prevent concurrent requests from exceeding rate limits; validated failover and metering against mock upstreams.

Vega - Multi-Agent AI Trading System | Python, MLX, Alpaca

[GitHub link](#)

- Orchestrated four research agents on one local LLM, with tool calling, time budgets, and checkpointed JSON outputs for Alpaca paper trading.
- Added an approve/reduce/veto review agent and deterministic risk checks for position sizing, trade limits, and buying power before execution.

Homelab: Self-Hosted Production Infrastructure | Proxmox, K3s, OPNsense, Cloudflare

[Project link](#)

- Operate public websites and projects serving **30K+** monthly unique visitors and **1M+** requests per month on a 3-node HA K3s/Proxmox cluster; automate rebuilds with Ansible.
- Secure public services with an OPNsense firewall, Suricata and CrowdSec intrusion detection, Cloudflare mTLS, and Authelia single sign-on; harden Linux hosts with nftables and fail2ban.
- Monitor hosts, containers, Proxmox, and the firewall with **16 Prometheus scrape jobs and 26 alerting rules**, plus Grafana/Loki and GeoIP-enriched dashboards that surface malicious traffic in real time.
- Automate builds, tests, and K3s deployments on push with Jenkins CI/CD and the GitHub Checks API.
- Built **TLS edge failover** with an Oracle HAProxy relay and Python watchdog; validated primary, local, and backup HTTPS routes before DNS failover, with automatic failback after sustained recovery.

GPT-2 Inference Engine from Scratch | CUDA C++, Nsight, CUTLASS | 4-person team

[Project link](#)

- Accelerated GPT-2 text generation on an NVIDIA A40 GPU, reducing per-token latency from **311 ms to 12.6 ms** through KV caching over a preallocated device cache.
- Implemented the FlashAttention-2 forward kernel, tiling Q/K/V through shared memory and registers so the full attention probability matrix never reaches GPU DRAM.
- Removed a 32-way shared-memory bank conflict with an XOR swizzle (**1.37x kernel speedup**) and profiled tile shapes; the team's stack reached **24.2x end-to-end speedup** (249.8 ms to 10.3 ms).
- Used an Agile workflow to track tasks and development progress, keeping the 4-person team on track.

TECHNICAL SKILLS

Languages: C, C++, Python, Rust, Java, TypeScript, SQL, Bash

Systems and Networking: Linux/UNIX, POSIX, concurrency, TCP/IP, memory management

Backend and Security: Node.js, Spring Boot, REST APIs, PostgreSQL, SQLite, OIDC/OAuth2, mTLS

Cloud and DevOps: Docker, Kubernetes (K3s), Ansible, Jenkins CI/CD, Prometheus, Grafana

Performance and Accelerators: CUDA, Vitis HLS, XRT, Nsight, GDB, profiling

Development Practices: Git, Agile workflow

COURSE CERTIFICATES

Anthropic Academy: AI Fluency (Framework and Foundations); Introduction to MCP

Hugging Face: AI Agents Course; LLM Course

Cloud Fundamentals: AWS; Microsoft Azure