Ian M. Lamont

Software Engineer

Philadelphia, PA | +1 (267) 586-1407 | imlamont@comcast.net | imlamont.com | LinkedIn: imlamont | GitHub: imlamont

Computer science engineer with hands-on systems experience: network protocol engineering at Connectify, plus independent projects spanning distributed storage, RISC-V processor design, GPU kernel benchmarking, and computer vision.

Education

University of Pennsylvania, School of Engineering and Applied Science — *Philadelphia, PA*

M.S.E. in Computer and Information Science; Concentration in Artificial Intelligence (Candidate) — Expected May 2027
B.S.E. in Computer and Information Science; Minor in Mechanical Engineering and Applied Mechanics — May 2026 (Cum Laude)
Coursework: Probability, Algorithms, Computability & Complexity, Applied Optimization (SAT, LP/MIP, constraint programming), Computer Architecture (including GPU architecture), Operating Systems, Distributed Systems, Databases, Linear Algebra, Machine Learning, Computer Graphics

Projects — Systems & Performance

PennCloud — Distributed Webmail & Cloud Storage — *C++ · Distributed Systems · Networking*

- Built the HTTP server (request routing, cookies, chunked transfer encoding) and the load balancer (round-robin distribution across frontend replicas) for a 4-person distributed webmail and cloud-storage system; contributed to the design of, and paired on, its fault-tolerant replicated key-value store modeled on Google's BigTable

ICPU-v0 — Pipelined RISC-V Processor — *SystemVerilog · RV32IM · FPGA · AXI4-Lite*

- Built a 5-stage pipelined RV32IM processor in ∼1.8k lines of SystemVerilog — full bypass network (MX/WX/WD), load-use and divide-use interlocks, branch flush, and a pipelined signed divider — driving instruction and data memory over an AXI4-Lite interface
- Ran Dhrystone in 255,086 cycles at 1.32 CPI, matching the reference trace cycle-for-cycle across the RV32IM test suite; profiling the stall mix showed taken-branch flushes cost 22.9% of cycles against 1.2% for load-use. Timing closure targeted at 20MHz on a Zedboard

PennOS — Unix-like Operating System — *C · Linux · Operating Systems*

- Built with a 4-person team; owned the kernel-level file I/O layer (open/read/write/close/lseek over a custom file descriptor table) and the user-space shell, including job control (fg/bg/jobs) with signal-based foreground-process management

qq — Command-Line AI Client — *C17 · POSIX · libcurl*

- Wrote a C17 command-line client for OpenAI-compatible APIs (∼2.8k lines; libcurl, cJSON, poll, fork/exec): 39 KB stripped binary that starts in 5–10 ms, with unit and mock-server integration tests passing under AddressSanitizer/UBSan and Valgrind

Triton vs. CUDA — Kernel Benchmarking & PTX Analysis — *CUDA · Triton · PTX · cuDNN*

- Implemented 2D convolution five ways — coalesced global-memory CUDA, two shared-memory tiling strategies, cuDNN, and Triton — and benchmarked all five over 32 input sizes from 128^2 to 4096^2, isolating kernel time from launch and transfer overhead
- At 4096^2 the best hand-written CUDA tiling ran 20.5ms; the Triton kernel ran 13ms (1.6x faster), within 1.5x of cuDNN and 40x ahead of the CPU baseline. Traced this gap to the emitted PTX

MiniMinecraft — Multithreaded Voxel Engine — *C++ · OpenGL · Multithreading*

- Real-time voxel engine awarded the semester Hall of Fame prize: overlapped procedural terrain generation with rendering across worker threads and optimized collision and physics hot paths

Experience

Connectify, Inc. — *Philadelphia, PA*

Software Engineering Intern — *June 2026 – August 2026*

- Researched and stress-tested the SRT streaming protocol for speed and reliability on proprietary OpenWRT hardware ahead of a customer-facing release, under a compressed deadline
- Developed and released an internal fork of ModemManager to improve IPC between the modem stack and VPN daemon

Systems Engineering Intern — *June 2025 – August 2025*

- Modified OpenWRT firmware for proprietary router hardware: integrated a LAN-wide VPN and built a hardware-monitoring and anomaly-detection daemon in C, with a web interface added for OpenWRT LuCI
- Prototyped a lightweight autoencoder for network-attack detection, reaching 98% accuracy within the compute and memory budget for real-time CPU inference on embedded hardware
- Assessed and implemented FinOps infrastructure improvements (server rightsizing), generating $12,000 in realized annual savings

University of Pennsylvania, Data Center Research Group — *Philadelphia, PA*

Research Assistant — *January 2025 — December 2025*

- Modeled data-center cooling requirements against power and water consumption across climates and resource constraints

Technical Skills

Languages: C, C++, Python, Assembly (x86, RISC-V, PTX), SystemVerilog, CUDA, Bash, SQL
Systems: Linux, operating systems, distributed systems, networking, multithreading, embedded firmware/OpenWRT
Profiling & Debugging: Linux perf, GDB, Nsight Compute/Systems, PTX inspection, benchmark harness design, FPGA timing analysis
Tools: Git, CMake, Docker, PyTorch, cuDNN