

Zhipeng (Philip) Zhang

Richmond, TX (Greater Houston) | 346-587-8364 | philipzhang7425@gmail.com | philipzhang.dev

Software engineer (MSCS @ Georgia Tech in progress) with production backend experience and hands-on experience in **systems programming, distributed systems, and applied AI**. Built a **concurrent C file server** and a **distributed file system in C++14/gRPC**; developed a **computer vision inference pipeline** with PyTorch; and shipped Java microservices in production, cutting p95 latency from 3s to 0.3s via **Redis caching and Kafka** decoupling. Comfortable working across low-level systems, production backend services, and ML pipelines. US citizen, open to **Systems Infrastructure, and AI Engineering roles**.

EDUCATION:

Georgia Institute of Technology

M.S. in Computer Science

Atlanta, GA

08/2025 – Present

GPA: 4.0

The Pennsylvania State University

B.A. in Economics

University Park, PA

01/2012-08/2014

WORKING EXPERIENCE:

SOFTWARE ENGINEER

NOV. 2023 – DEC. 2024

HAO BRANDS INC., SOMERSET, NJ

- Cut peak page-load 3,000→300 ms by adding Redis caching and decoupling with Kafka.
- Owned ~20% of Ammall e-commerce backend; delivered core services end-to-end.
- Iterated APIs with Agile/Scrum to meet changing UI needs; shortened dev/deploy time by 20%.
- Modeled domains with OOAD/DDD; improved scalability and cut maintenance effort by 30%.
- Built Java microservices (Spring Boot/Cloud, Hibernate, MariaDB) with ZooKeeper for coordination.
- Prioritized requirements with AI-assisted analysis; shipped features aligned to the SDLC roadmap.
- Containerized and deployed services to private cloud with Docker; managed source/builds with Git + Maven, and automated testing/CI/CD via shell scripts and GitHub Actions.

PROJECT EXPERIENCE:

Multithreaded File Server & Client (C/POSIX Threads)

Graduate Project, CS 6200 | Fall 2025

- Built C concurrent file server + load-generator, implementing an HTTP-like GetFile protocol for reliable static-file serving
- Designed thread-pool + bounded queue; enforced back-pressure with pthread mutex/condvar to avoid races under load.
- Hardened socket I/O and defensive parsing to withstand malformed traffic.
- Benchmarked throughput/latency across worker counts; analyzed CPU-core scaling; documented worker-per-conn vs. thread-pool trade-offs.
- Debugged memory/concurrency with GDB/Valgrind; added unit tests for queue semantics and parser.
- Dockerized dev env; reproducible Makefile builds; lint/static-analysis as pre-commit hooks.

Distributed File System (C++14/gRPC)

Graduate Project, CS 6200 | Fall 2025

- Built a distributed file system in C++14 using gRPC and Protocol Buffers, supporting concurrent store, fetch, delete, and list operations across multiple clients and a single server.
- Implemented write-lock management with server-side coordination to prevent conflicting updates from concurrent clients.
- Designed a weakly consistent cache layer with inotify-based file watching, enabling automatic client-server synchronization without manual polling.
- Handled gRPC deadline timeouts and error propagation to ensure graceful degradation under network delays and partial failures.
- Managed thread-safe synchronization between async inotify events and periodic sync threads using mutex/condition-variable patterns.
- Dockerized development environment; structured builds with CMake and automated testing through CI workflows.

Computer Vision Insect Detection Pipeline (Python/ML)

Graduate Project | Fall 2025

- Integrated and fine-tuned pre-trained deep learning models for multi-class insect detection; adapted model outputs to meet domain-specific accuracy and latency requirements.
- Built end-to-end inference pipeline in Python: image preprocessing, model inference, and structured result output; designed for reproducibility and downstream integration.
- Evaluated model performance across detection thresholds; iterated on confidence scoring and post-processing logic to improve precision/recall trade-offs under real-world image conditions.
- Collaborated with domain stakeholders to align model behavior with operational requirements; documented pipeline design and model assumptions for cross-functional handoff.
- Managed experiment tracking and versioning to ensure reproducible results across model iterations.

TECHNICAL SKILLS:

Languages: C, C++, Java, Python, SQL

Systems Programming: Linux, POSIX threads, Sockets, TCP/IP, GDB, Valgrind

AI / ML: PyTorch, Deep Learning, Computer Vision, Inference Pipeline Design, Model Fine-tuning

Backend / Data: Spring Boot/Cloud, REST, Hibernate, Redis, Kafka, Docker, PostgreSQL, MariaDB

Infrastructure & Tooling: Docker, Git, GitHub Actions (CI/CD), Maven, Make, CMake