Brandon Wang

wangb@berkeley.edu LinkedIn
512-940-6636 GitHub Portfolio

Education

University of California, Berkeley | GPA: 3.9/4.0 **May 2024**
B.A. Computer Science, B.A. Statistics *Berkeley, CA*

Relevant Coursework: Data Structures and Algorithms, Computer Architecture, Operating Systems and Systems Programming, Discrete Mathematics and Probability Theory, Efficient Algorithms and Intractable Problems, Circuits and Differential Equations, Signal Processing, Linear Algebra, Multivariable Calculus, Probability and Mathematical Statistics

Technical Skills

Languages: C, C++, Python, Golang, Rust, RISC-V, x86, Java, JavaScript, TypeScript, SQL, Bash
Technologies: React.js, Node.js, Docker, Kubernetes, PostgreSQL, MongoDB, AWS, GKE, Git, CI/CD

Work Experience

Citadel **Sept 2022 – Dec 2022**
Software Engineering Intern *New York, NY*

- Interning September '22 - December '22 for 12 weeks on the Equities Engineering Team in New York City.

Apple **May 2022 – Aug 2022**
Software Engineering Intern *Cupertino, CA*

- Proposed and architected a distributed cloud services cost optimization engine for user/organizational resource provisioning in GKE to reduce annual cloud-related costs in Apple FM Studio by up to 45% amortized at scale.
- Developed a high-frequency parallel processing system in Kubernetes to deliver throughput loads of up to 100k+ RPS, while synchronizing CPU, memory, and storage metrics collection across pods and services via cluster networking with TCP/UDP.
- Saved 100s of engineering-hours by automating compatibility builds for GCR Docker images to conform to AMD architecture.

NVIDIA **Feb 2022 – Apr 2022**
Software Engineering Intern *Seattle, WA*

- Implemented POSIX inode metadata support within a distributed file system mount of an OpenStack Swift object store.
- Reduced latency of lock acquisition by 10% on average through establishing an executive lock release algorithm which recalculates thread priority to avoid priority inversion in the underlying OS scheduler.
- Increased test coverage by 70% by porting file system tests from LXD to Docker and developed comprehensive logging tool.

Segmed **May 2021 – Aug 2021**
Software Engineering Intern *Palo Alto, CA*

- Built core REST API and caching systems of RESTful microservices in Go using mux router, Redis, and structured unit tests.
- Delivered production features for datasets valued at $200k+ in React, Go, and automated PostgreSQL queries with cron jobs.
- Improved data pipeline efficiency by up to 35% by introducing an OCR model utilizing an OpenCV scene detection scheme.

Projects

Low-latency Matrix Computation in C (link) | *Performance library in C for logical matrix operations* **Stack:** C, SSE, SIMD

- Accelerated by multiple-instruction, multiple-data parallelism to achieve >700x time optimization for matrix powering.

80x86 PintOS (link) | *Lightweight OS with kernel threads, user programs, and a file system* **Stack:** C, C++, x86 Assembly

- Implements creating process control syscalls, thread/context switching, priority scheduler, file system buffer cache, etc.

High-frequency Rust HTTP Server (link) | *Parallelized HTTP Server written in Rust* **Stack:** Rust, Tokio Sockets

- Utilizes a listening socket which spawns off an asynchronous thread which obtains a new socket address per HTTP request.

Multithreaded Server in C (link) | *Multithreaded server written in C* **Stack:** C, POSIX Threads

- Includes thread pools, socket specifications, mutex locks, thread safety, slow disk access simulation, and data structures in C.

Voice-controlled Rover (link) | *Self-stabilizing mobile robot guided by speech input signals* **Stack:** C++, Python, Circuitry

- Driven by an Arduino, mic-board, regulator circuits, controlled via feedback controls, classifies speech signals via SVD/PCA.

RISC-V CPU (link) | *A 32-bit RISC-V instruction pipeline built from logic gates, multiplexers, and registers* **Stack:** RISC-V, Logisim

- Supports all RISC-V ISA instructions through datapath and control logic, including the ALU, RegFiles, and 2-stage pipelining.

Securities Trading Bot (link) | *Bot that trades using ETF arbitrage and ADR pairs-trade strategies* **Stack:** Python, socket.py

- Competed at Jane Street's ETC, implements a high-frequency trading model trading across a live trading environment.