JESSE TUĞLU

Ann Arbor, MI | 845-274-8468 | tuglu@umich.edu | linkedin.com/in/jessetuglu | github.com/ulgut

EDUCATION

University of Michigan — **Ann Arbor, MI**
B.S Computer Science, B.S Economics; GPA 3.94/4.0 — *September 2021 - May 2024*

St. Andrew's High School — **Middletown, DE**
Summa Cum Laude — *September 2017 - June 2020*

Relevant Coursework
Computer Organization • Data Structures and Algorithms(both C++ and Java) • Linear Algebra • Vector/Multivariate Calculus • Discrete Math • Statistics/Probability • Computer Vision • Theory of Computer Science • Compiler Construction • Networks • Algorithms • Operating Systems • Econometrics • Calculus I-III • Micro/Macro Economics

EXPERIENCE

Amazon Web Services (AWS) — **Palo Alto, CA**
Software Development Engineer Intern — *May 2023 - Present*

- Worked on the AWS OpenSearch Serverless Security team building out distributed traffic throttling and reverse proxy systems for multi-tenant environments in Golang.
- Designed and built a low-latency, customizable throttling infrastructure that relies on a distributed cache replicated across 100s of machines.
- My code accepts/denies requests at the ingress of the service, protecting against the noisy neighbor issue (only throttling the noisy customers) while having minimal latency impact (an added **p99 of 150us** to overall request latency).

University of Michigan College of Engineering — **Ann Arbor, MI**
EECS 376: Theory of Computer Science Instructional Aide — *August 2023 - Present*

- Course topics include: Algorithm families, Computability theory (Turing Reductions and Decidability), Complexity theory (P and NP, NP Hardness, NP Completeness), Search & Approximation algorithms, Randomized algorithms, Monte Carlo methods and Concentration bounds, and Encryption schemes/secret sharing.

Shopify Inc. — **New York, NY**
Backend Software Engineer Intern - Advanced Edits Team — *May 2022 - August 2022*

- Implemented multi-threading and synchronization into multiple Shopify CLI theme commands using a combination of thread pools and condition variables to coordinate work between worker threads, resulting in a **10x** speedup in execution for commands.
- Streamlined theme development by enhancing Shopify's Storefront Renderer with local theme hot-reloading features using Server-Sent Events and DOM patch-up which saved **15 seconds** for merchants when running **theme serve**.

BC Distributed Computing Group — **Boston, MA**
Distributed Systems Research Assistant — *November 2020 - January 2022*

- Worked with research team under Dr. Lewis Tseng, co-authored paper which presents randomization as a novel solution to achieving distributed consensus within data centers.
- *[SOSP 2021]: Rabia: Simplifying State-Machine Replication Through Randomization* – **15.5% acceptance rate**.
- Profiled and modified Golang code for competing consensus protocols (Raft and EPaxos) to gauge Rabia's relative performance.

Geologie Inc. — **New York, NY**
Fullstack Software Engineer — *March 2020 - January 2022*

- Used Ruby, Ruby on Rails, and Python to build APIs backed by PostgreSQL and Google BigQuery.
- Created/updated multiple reliable, fast, high-traffic API endpoints which receive 100000s of client requests per day.
- Architected hosted application that processes 60% of company data using CRON jobs and temporary PostgresSQL staging tables, and then stores it securely in our data warehouse.

PROJECTS

FCOS Object Detection Implementation

- Implemented a single-stage object detector using PyTorch that uses a feature pyramid network (FPN) to produce bounding box predictions at multiple spatial scales for any given image.
- Based on the paper: FCOS: Fully Convolutional One-Stage Object Detection.

Decaf to MIPS32 Compiler

- Created a fast, optimized compiler for the Decaf programming language in C++17 targeting the MIPS32 assembly language.
- Leveraged Flex and Yacc to build out a frontend architecture which handled parsing, lexing, and semantic analysis.
- Handwrote custom IR optimization heuristics including optimal register allocation techniques, redundant/dead instruction removal, and copy propagation.

ETF Arbitrage Bot

- Built a bot in Python which listens to live quote updates for 50+ US ETFs.
- The program takes advantage of mispricings between an ETF's share price and NAV value and opens a position anticipating the price will eventually reflect the true NAV again.

TECHNICAL SKILLS

Languages C++ • C • Go • Javascript/Typescript • Ruby • Python • SQL • Bash • Elixir
Technologies/Frameworks React.js • Ruby on Rails • React Native • GCP • Docker • NumPy • PyTorch