

Zibai (Matthew) Wang

zw737@cornell.edu (preferred contact) | (+1) 607-279-1958

masasukam.github.io | github.com/Masasukam | linkedin.com/in/matthew-wang-9847331b7/

EDUCATION

Cornell University, Ithaca, New York

Jan 2024 - Dec 2024

Master of Engineering: Computer Science

Highlighted Courses: Parallel Computing, Computer Vision, 3D Reconstruction, Computer Networks, Operating Systems

University of British Columbia, Vancouver, BC

Sep 2019 - May 2023

Bachelor of Science: Computer Science and Mathematics; With Distinction

Highlighted Courses: Applied AI & ML, Databases, Algorithms, Object-Oriented Programming, Distributed, Linear Algebra

SKILLS

Languages	C++, Python, Java, C, C#, JavaScript, TypeScript, Kotlin, PHP, HTML/CSS, Bash
DataBase & Cloud	MySQL, MongoDB, Oracle, NoSQL, DynamoDB
Frameworks	PyTorch, TensorFlow, React, Node.js, Cuda, MPI, OpenMP, NumPy, Flask, Spring, Maven
Tools	Linux, Unix, Git, AWS (Cloud Practitioner), Docker

WORK EXPERIENCES

Software Research & Development Intern

Sep 2021 - April 2022

INTEL Corporation, Vancouver, BC

- Developed and implemented **C**-based optimization settings for quality-speed tradeoffs of low-delay streaming and video compression in the widely-used **SVT-AV1** encoder, decreased ~10% bitrate loss and increased ~8% speed.
- Developed test scripts using **Python** and ran on **AWS EC2 Linux** instances for evaluating bitrate/speed tradeoffs for existing **SVT-AV1** features; Collaborated across teams to perform comparative analysis among video encoders in the market.
- Designed and implemented an optimized video decoder program using **C**. Simplified the 5 decoding levels to a more maintainable 2-level system by evaluating the decoder speed against existing solutions in the market.
- Implemented unit tests using **Check framework**, achieved test coverage of 95%+ and packaged the program using **CMake**.

Software Developer Intern

July 2019 - Aug 2019

Tencent Holdings Ltd, Shenzhen, China

- Built an event-driven notification system using **Python** and **Flask framework** to keep track of keywords and feedback given by users on stock forums. Created a user subscription interface using **React**.
- Extracted dynamically generated content from **JavaScript**-based stock forums by integrating **Python** libraries **Scrapy** and **Splash**, enabling server-side **JavaScript** execution and rendering for full **HTML** access. Utilized dynamic **IPs** and controlled crawling rate to avoid throttling.
- Persisted users post data into **MySQL** databases consisting of >5G user data for further analysis and relational database management. Crafted schema and employed strategic indexing on crucial attributes for efficient data retrieval.

PROJECTS

High-Performance Computing(HPC) - Fish School Search (FSS) Optimization

Jan 2024 - Apr 2024

- Led a team of three to develop a **C++** serial implementation, then applied GPU acceleration using **CUDA**, and optimizing CPU usage with **OpenMP** and **MPI** in parallel implementations, emulating fish foraging behavior to find optimal solutions.
- Achieved 80% parallel efficiency by leveraging multi-core/thread communication, **CPU SIMD** instructions, and sparse linear algebra, significantly reducing computation time from 2 hour to 10 minutes and enhancing scalability when applied to large-scale problems (e.g., parameter optimization in machine learning, resource allocation in computer networks).

UBC Student Center

Sep 2020 - Dec 2020

- Led the development of a **RESTful** full-stack web App for querying and managing UBC courses, room assignments, and grade calculations using **TypeScript** for the backend and **React** for the frontend.
- Developed back-end with dataset controller to parse **JSON** and **HTML** files, and query-controller to handle queries formatted as **JSON** strings, leveraging **MongoDB** for scalable data management.

Hospital Management System

Sep 2019 - Dec 2019

- Designed and implemented a full-stack hospital management system using **Java**, **Spring** framework, and **MySQL**, applying **OOP** principles and the **Observer pattern** to improve tracking of patient appointments, health status, and doctor availability, while ensuring transparent and decoupled interactions between doctor and patient objects.
- Utilized the **Spring RestController** to build **Restful APIs** for patient data retrieval and update. Created **GUI** using Java **Swing Framework** for user interaction, and employed **Maven** for dependency management.