Jiahao (Jayce) Guo

(917) 293-5089 | jguo23@buffalo.edu | https://www.linkedin.com/in/gjiahao/ | https://github.com/Y2Nk4

Education

University at Buffalo | Computer Science BS *Sep 2020 - May 2024*
GPA: 3.82/4.0 Buffalo, NY

Skills

Language: C, C++, JavaScript, TypeScript, Java, Python, Go, HTML, CSS
Framework: Flask, Swagger, Spring, Spring Boot, Pandas, Koa.js, Vue.js, React.js, Express.js, NumPy, SciPy, Scikit-Learn
Database: MySQL, MongoDB, Redis, AWS DynamoDB, ETCD
Tools: Git, Linux, Docker, Postman, IntelliJ, VS Code, Wireshark, SSH, Maven, DevOps, Apache Thrift

Professional Experiences

Amazon.com, Inc. *Sunnyvale, CA*
Software Engineering Intern – Alexa Automatic Speech Recognition *May 2023 - Aug 2023*
Tech Stacks: ***Java, AWS Step Function, Lambda Function, AWS EventBridge, DynamoDB, React, A/B Testing, Agile Development***

- Designed and implemented an automatic gating system based on benchmarking metrics.
- Utilizing AWS Lambda function and EventBridge to cooperate with other internal benchmarking services to automate the A/B testing process to reduce time of A/B testing in the releasing process by 60% and safeguarding the releasing process.
- Implemented a dashboard using React.js and applied React Hooks to initialize, monitor and manage A/B testing and configurations.
- Managed the deployment of infrastructure for the team's A/B testing services through AWS CDK, efficiently deploying and managing the stacks for A/B testing services and workflows.

FunStone Network (China) *Remote*
Software Engineering Part-time | Technologies: ***PHP, Node.JS, Thrift, ETCD, MySQL, Docker, CI/CD*** *May 2018 - July 2019*

- Implemented a distributed computing management system using Node.JS, Apache Thrift, and ETCD to evenly distribute computing jobs into multiple computing nodes.
- Integrated GitHub CI and Docker to manage and automate the deployment of the services.
- Implemented the backend of an e-shop website that served over 10,000 customers, using PHP, Laravel and MySQL. This system utilized MySQL locks to ensure data consistency and avoid concurrency issues.
- Utilized Redis to implement a Cache system to improve the web responding time by 64%.

Projects

Text Chat Application | *Technologies:* ***C++, TCP Socket Programming, P2P File Share***

- Implemented the client and server components of a text chat application, which consists of one chat server and multiple chat clients over TCP connections. The clients and server will encode the message in a certain structure and send them through the TCP socket. The packet structure is optimized to reduce transmission overhead.
- Implemented a P2P file transfer under TCP, the client sending the message will firstly send a TCP packet to the server fetching the listening ports of the client who receives the file, then encode the file in certain structure and sends to the other client. The client uses SHA1 hash to ensure the integrity of the transmitted file.

Image Type Classification Neural Network | *Technologies:* ***Python, NumPy, PyTorch, Neural Network, Deep Learning***

- Built and trained a Convolutional Neural Network with 11 layers to classify images into 3 types using an image dataset with over 30k images using PyTorch.
- Used CUDA and ML techniques (e.g., regularization, learning rate scheduler) to accelerate the training process.
- Implemented a dashboard that allows the users to upload images and predicts the types of images using React.js and Flask.

Ecommerce Website (Full Stack) | *Technologies:* ***TypeScript, Koa, MongoDB, JWT, Vue.JS, NGINX, Redis, RabbitMQ***

- Designed and implemented a scalable e-commerce website backend with Node.js, Koa and Docker. The load balancer monitors the loads of each Docker container instance and automatically up-scale or down-scale based on loads.
- Implemented the checkout system that can handle multiple discounts and multiple types of taxes.
- Implemented front-end pages (product description, login/register page) using Vue.JS and Vuetify.
- Implemented an order processing pipeline with RabbitMQ to Implemented an order processing pipeline with RabbitMQ to increase the order handling capacity in peak hours by 74%.

Distributed Database with Raft Consensus | *Technologies:* ***Go, Raft, Distributed System, Distributed Database***

- Implemented a Distributed Database in which changes are finalized upon obtaining a quorum of server acknowledgments, ensuring the persistence of committed changes if node failures remain below 50%.
- Implemented the mechanism that leader elections will be automatically initiated by nodes in event of leader node failure.
- Implemented a heartbeat mechanism that every node broadcasting its status to all other nodes to detect node failure.