Jiahao (Jayce) Guo

(917) 293-5089 | jiahaohuo4@gmail.com | 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

Professional Experiences

University at Buffalo – Enterprise Application Services *Buffalo, NY*
Application Developer | Technologies: ***Java, AngularJS, Adobe Experience Manager, Agile Development*** *Jun 2024 - Now*

- Designed and implemented reusable page templates using Java and HTL for the university's home page, effectively serving 35,000 students and faculty members.
- Migrated over 3,000 web pages from an outdated architecture by developing a Java-based tool.

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

- Reduced 60% A/B testing time and improved the reliability of the model release process by designing and implementing an automatic gating system based on benchmarking metrics.
- Automated the A/B testing process by integrating the system with internal benchmarking services using AWS Lambda Function and EventBridge.
- Provided customer teams with real-time insights and control over the testing, enhanced decision-making capabilities by developing a comprehensive dashboard using React to initialize, monitor, and manage A/B testing and configurations
- Managed the infrastructure deployment for the team's A/B testing services using AWS CDK, efficiently deploying and managing the stacks for A/B testing services and corresponding workflows.

FunStone Network *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.
- Developed the backend of an e-shop website serving over 10,000 customers, using PHP, Laravel, and MySQL. Utilized MySQL locks to maintain data consistency and prevent concurrency issues, resulting in a reliable e-commerce platform.
- Improved the web response time of multiple APIs by 64% by implementing a Redis-based cache system.
- Integrated GitHub CI and Docker to manage and automate the deployment of the services.

Projects

E-commerce Website (Full Stack) | *Technologies:* ***Java, Spring Boot, MongoDB, Swagger, NGINX, Redis, RabbitMQ***

- Designed and implemented a scalable e-commerce website backend with Java, Spring Boot, and Docker. The load balancer monitors the loads of each Docker container instance and automatically up-scales or down-scales based on loads.
- Implemented the checkout system that can handle multiple discounts and multiple types of taxes.
- 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%.

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 messages 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 first send a TCP packet to the server fetching the listening ports of the client who receives the file, then encode the file in a certain structure and sends it to the other client. I also implemented a mechanism to ensure the integrity of the transmitted file, utilizing the SHA1 hash.

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 the event of leader node failure.
- Implemented a heartbeat mechanism that every node broadcasts its status to all other nodes to detect node failure.

Summaries

Skills: A/B Test, Distributed Systems, Micro-services, Restful API, DevOps, C, C++, JavaScript, TypeScript, Java, Python, Go
Frameworks: Flask, Swagger, Spring, Spring Boot, Koa.js, Vue.js, React.js, Express.js, NumPy, SciPy, PyTorch, Pandas
Databases: MySQL, MongoDB, Redis, AWS DynamoDB, ETCD, SQL / Non-SQL DBs
Tools: Git, Linux, Docker, Postman, IntelliJ, VS Code, Wireshark, SSH, Maven, Apache Thrift