Huy Tran

408-594-8987 | huytran@berkeley.edu | linkedin.com/in/huytt621 | github.com/huytt621

Technical Skills

Languages: Java, C/C++, Go, Python, JavaScript, TypeScript, HTML/CSS, SQL, GraphQL
Frameworks/Libraries: React, Node.js, Express.js, Next.js, Spring Framework, Project Reactor
Developer Tools: Git, GitHub, BitBucket, Jira, Confluence, Docker, Amazon Web Services

Experience

Backend Software Engineer **San Mateo, CA**
Roblox *July 2024 – Current*

- Improved delivery time of Confluence WebHooks 12% by optimizing payload generation and processing
- Eliminated 140 million daily requests to Amazon RDS by removing DB dependency for sending WebHooks
- Reduced volume of WebHook messages sent to SQS by 29%, removing redundancy by batching callbacks
- Increased scope of analytics by proposing and implementing 5+ new metrics for the SignalFX Dashboard
- Raised test coverage in the Confluence Monolith by writing acceptance tests for multiple WebHook events

Backend Software Engineer **San Francisco, CA**
Atlassian *May 2023 – August 2023*

- Improved delivery time of Confluence WebHooks 12% by optimizing payload generation and processing
- Eliminated 140 million daily requests to Amazon RDS by removing DB dependency for sending WebHooks
- Reduced volume of WebHook messages sent to SQS by 29%, removing redundancy by batching callbacks
- Increased scope of analytics by proposing and implementing 5+ new metrics for the SignalFX Dashboard
- Raised test coverage in the Confluence Monolith by writing acceptance tests for multiple WebHook events

Full Stack Software Engineer **Mountain View, CA**
Atlassian *May 2022 – August 2022*

- Leveraged reactive architecture to create a responsive and scalable notifications microservice for Confluence
- Decoupled email settings from Monolith by designing a new REST API with Spring and Project Reactor
- Reduced latency of queries by 19% by optimizing the DynamoDB schema for critical access patterns
- Modernized the email settings page by creating React components that follow the Atlassian Design System
- Integrated the notifications service with the settings page by developing a GraphQL API with Apollo Client

Projects

PintOS

- Collaborated with a team of four to enhance a legacy x86 Operating System codebase written in C
- Implemented execution of user programs by creating process control and file system calls (e.g. fork, open)
- Improved performance by developing user-level threads and synchronization to enable multithreading
- Refined filesystem by implementing file growth and subdirectories, inspired by the Berkeley Fast Filesystem
- Halved runtime of performance tests by designing a kernel buffer cache with the Clock replacement policy

Secure File Sharing System

- Developed an end-to-end secure filesystem that provides confidentiality and integrity of files written in Go
- Reduced bandwidth and asymptotic runtime of file appends using a singly-linked list file format
- Facilitated secure file sharing by designing an algorithm inspired by callbacks and hybrid encryption
- Designed an efficient revocation of file access algorithm inspired by shared memory and flattened trees

Education

University of California, Berkeley
Bachelor of Arts in Computer Science, Minor in Data Science

Relevant Courses: Data Structures, Computer Architecture, Efficient Algorithms, Database Systems, Computer Security, Operating Systems, Programming Languages and Compilers, Computer Networking, Machine Learning