Noah Staveley

206-819-1494 | noahstaveley@gmail.com | https://www.linkedin.com/in/noah-staveley/

Technical Summary

Skills: Java, C, C++, Python, Go, AWS, Docker, Kubernetes, Git, PyTorch, TensorFlow, OpenSearch, Amazon Linux, Vector Databases, Cloud Computing

Work Experience

Cloud Software Engineer August 2022 - present
Intel Corporation *Remote*

- Cloud Software Optimization.
  Modified open-source libraries and tools to optimize performance for Intel hardware, improving execution efficiency and leveraging architecture-specific features.
- Cloud Infrastructure Optimization
  Identified and implemented optimization opportunities in cloud environments, focusing on workload management, resource allocation, and performance tuning for cost and efficiency.
- Delivered tools and whitepapers on optimizing workload performance in VMs running Amazon Linux and BottleRocket operating systems with x86-64 microarchitecture v3 ABI.
- Proficient with Linux-based systems (Ubuntu, CentOS) and virtualization using QEMU/KVM, and containerized environments (Docker, Kubernetes).

Education

University of Washington Seattle, WA
Bachelor of Science in Computer Engineering *June 2022*

Projects

IOT Device | *Python, Embedded C, Flask, SQLite* | Winter 2022

- This health monitoring system conducts a series of tests including an EMG test
- Users can login and view their test results online
- Wearable test sensors are connected to an STM32 ARM Cortex-M Microcontroller
  - Sensor data is transmitted to a Raspberry Pi through UART
- Sensor data is then stored and processed in a ThingSpeak database
- The backend of our website links ThingSpeak data to Users in a SQLite database

xk-OS | *C* | Winter 2022

- Primitive operating system designed to mimic early UNIX systems
- Implemented creation of file systems, inter-process communication and multi-processing
- Created a user interface shell that executes commands such as cat, echo, grep, ls, wc and pipe

Networking Projects | *Java, Python* | Autumn 2021

- Implemented a multi-threaded Web Server
- Created a client application that communicates with the server over UDP and TCP sockets
- Built simple networks using SDN primitives and OpenFlow protocol
- Implemented networks with nodes connected over both TCP Reno and TCP BBR connections
  - Generated graph visualizations to compare performance