Zian Wang

zianw@andrew.cmu.edu (650)-224-8264 Linkedin.com/in/zianwang/ Mountain View, CA

Education

Carnegie Mellon University **01/2020-05/2021**
Master's of Science in Electrical and Computer Engineering; GPA: 4.00/4.00
Relevant Courses: Cloud Computing, Introduction to Computer Systems, Foundation of Software Engineering

Nanjing University of Posts and Telecommunications **09/2015-06/2019**
Bachelor's of Engineering in Internet of Things Engineering; GPA: 3.67/5.00, top10%
Relevant Courses: Data Structures and Algorithms, Operating System, Internet of Things Security

Programming Skills

- **Languages:** Java, C, Python, Scala, C++, Nodejs
- **Technologies:** Docker, Kubernetes, MySQL, Reddis, MapReduce, Spark, Kafka, Samza, Terraform, Linux, CircleCI, GitHub

Experience

Study on Trust Service Coordination in the Social Internet of Things **09/2017-06/2018**
Undergraduate assistant researcher, supervised by Dr. Jin Qi NJUPT, Nanjing, China

- Collaborated with 1 PhD to build the QoS-driven trust service coordination model which realizes the multi-index evaluation adaptation, resulting in a publication to Sensors 2018 and presented findings to Dr. Jin Qi
- Optimized the multi-objective gray-wolf algorithm to find the pareto-optimal solution

Publication

Jin Qi, **Zian Wang**, Bin Xu, Mengfei Wu, Zian Gao and Yanfei Sun, "QoS-Driven Adaptive Trust Service Coordination in the Industrial Internet of Things", *Sensors 2018*, Volume 18, Issue 8, 2449. (Link: https://doi.org/10.3390/s18082449)

Projects

Computer System in C (CMU, Mountain View, CA) **07/2020-08/2020**

- Implemented a **cache** system which uses the LRU replacement policy, and follows a write-back, write-allocate policy in C
- Realized the **malloc** function supporting 64-bit systems with a utilization of 74.3%, and a throughput of 16809 KOPS in C
- Customized my own **shell**, supporting foreground jobs and background jobs through **multi-process control** & **signal handling**
- Wrote a **multi-threaded** caching **web proxy** following the LRU replacement policy in C

Twitter User Recommendation System/API (CMU, Mountain View, CA) **05/2020-06/2020**

- Collaborated with 2 software engineers to design this **multi-tier** web service system, and applied **Java Spring Boot** in the web-tier, with **MySQL** in the storage-tier
- Implemented the extract, transform, and load(**ETL**) process from 10TBs of twitter data into the storage-tier using **Apache Spark** in **Scala** on the **GCP** platform
- Explored different types of **web framework** and **database schema** and make web-tier **multi-threaded** to maximize the system's throughput performance under a limited budget

Distributed Key-Value Store with Replication & Strong Consistency (CMU, Mountain View, CA) **04/2020-06/2020**

- Deployed **cross-region** coordinator servers in front of data store servers with a true-time server with **Terraform** on **AWS**
- Implemented the read-preferring read-write lock with **Java multi-threading**

Emergency Social Network (ESN) Web APP (CMU, Mountain View, CA) **01/2020-06/2020**

- Collaborated with 4 software engineers to design and develop a social network for citizens during covid-19 pandemic, with functions including joining virtual communities, chatting privately and publicly, searching information, checking statistics
- Applied **Node.js** (**JavaScript**) and **Express** in the backend, with **Bootstrap**, **CSS** and **jQuery** in the frontend
- Implemented **CI/CD** with **CircleCI** and deployed on **HeroKu**; Wrote unit tests and integration tests using **Jest**

Docker and Kubernetes (Microservice) (CMU, Mountain View, CA) **01/2020-02/2020**

- Built **docker** images out of Java applications with **Spotify docker-maven-plugin**
- Deployed three microservices of a real-time chat service, login, chat, and profile service into the **Kubernetes** cluster with **Helm Charts**, and routed traffic into different microservices using **NGINX ingress controller**
- Realized the **auto-scaling** of **Kubernetes** with **Horizontal Pod Autoscaler**, and multiple cloud deployment (**GKE** & **AKS**)