TIANJIAN HUANG

thuang2@andrew.cmu.edu | (412) 933-9229 | https://www.linkedin.com/in/tianjian-huang/

EDUCATION

Carnegie Mellon University - *M.Sc. in Computer Networking | GPA: 3.95/4.00, Ranking: Top 3%* 09/2021 – 05/2023

Major Coursework: Introduction to Computer System (15-513), Storage System (15-746), Advanced Cloud Computing (15-719), Principles of Software Construction (17-514), C++ Game Programming (15-666)

The Chinese University of Hong Kong - *B.E. in Computer Science | GPA: 3.65/4.00, Ranking: Top 10%* 09/2017 – 07/2021

Major Coursework: Operating System, Data Structure & Algorithm, Distributed & Parallel Computing, Software Engineering, Database System, Robotics System, Cloud Computing

SKILLS AND PROFICIENCIES

Programming Languages	C/C++, Python, Go, Java, JavaScript, MATLAB, HTML, CSS, MySQL
Frameworks and Tools	**System**: Linux, FUSE, Bash, MPI, OpenGL, Git, GCC, GDB, gRPC; **Cloud & DevOps**: AWS EC2/S3, Terraform, Apache Spark, Docker, Kubernetes, VMware Cloud Disaster Recovery (VCDR); **Robotics**: Robot Operating System (ROS), Gazebo; **Full Stack**: Flask, Gin, Design Patterns, OOP;

PROFESSIONAL EXPERIENCE

VMware - Software Engineer Intern Palo Alto, CA | 05/2022 - 08/2022

- Using gRPC framework with multiple languages (C, java, etc.), implemented a task-reporting service that recorded key status of backup tasks; improved the scale of Site Recovery task status tracking from protection-group-level(multiple VMs) to VM-level.
- Improved the CLI tools for VMware Cloud Disaster Recovery (VCDR) to display backup status, and to present data with access control; the new CLI tools provides detailed views of tasks and enhanced customer interaction and debugging experience.
- Refactored source code for Docker-managed backup and restore service; improved readability and consistency of source code.

Carnegie Mellon University - Research Intern Pittsburgh, PA | 06/2020 – 08/2020

- Wrote Python scripts to extract & format GB-level VERBOSE log data of RoboTutor (an Android tablet tutoring APP).
- Used Disco (a process mining tool) to create process models for RoboTutor's log data, performed educational process mining (EPM), and summarized children-tutor interaction behavior patterns.
- **Published on *RISS Journal 2020***: *Using Process Mining to Analyze Children's Interactions with RoboTutor.*

The Chinese University of Hong Kong - Robot Software Engineering Intern Shenzhen, China | 06/2019 – 01/2020

- Built a Linux environment for running Robot Operating System (ROS) on Raspberry Pi 3 Model B, and tailored C++ ROS packages to increase image processing speed from 5 FPS to 10~15 FPS.
- Set up a simulation platform using Gazebo simulator, and constructed a 3D testing car model in Gazebo using URDF.
- Developed visual navigation algorithms on a real hardware platform in order to conduct global path planning using monocular vision and Quick Response (QR) codes.

COURSE PROJECTS

Cloud Computing 01/2022 – 05/2022

- Used **Terraform** to deploy image classification services on AWS EC2 instances and to manage AWS auto-scaling policies.
- Implemented workload-aware auto-scaling controller and request-based microservices with **AWS Lambda**.
- Designed and implemented a distributed Common Crawl ELT program with **Apache Spark**.
- Learned basic **Kubernetes Scheduler Extender** and **Kube-batch API**; implemented heterogeneity-aware, shortest-job-first scheduling policy that aimed at minimizing job completion time.

CloudFS 10/2021 – 12/2021

- Built a simple **hybrid cloud file system (CloudFS)** that stored small files and metadata on local SSD and large file contents in cloud via FUSE and the Amazon S3 C API.
- Performed **file-content-based deduplication** using Rabin Fingerprinting algorithm, achieving reduction of cloud storage cost.
- Developed **file system snapshot** functionality, supporting backup/restore of the entire CloudFS and snapshot previewing.
- Implemented **local caching** for cloud file contents, which reduced total cloud cost by 43.6% compared with the benchmark.

MyFTL 09/2021 – 10/2021

- Designed and wrote a simple C++ **flash translation layer (FTL)** which achieved physical-logical address translation, garbage collection (GC), and wear leveling under common workload situations.
- Developed a **hybrid log-block address mapping scheme** that utilized over-provisioning space to reduce write amplification, inherited the flexibility of page-level mapping and reduced RAM consumption.
- Implemented and comprehensively tested a **cost-benefit GC and wear-leveling policy** that released invalid pages in SSD, aiming at write amplification reduction and flash blocks endurance maximization.