Haiyang Yu

https://www.linkedin.com/in/haiyang-yu-246a213b/
haiyangy@andrew.cmu.edu
+1-408-839-3245
Cupertino, CA

Education

- **Carnegie Mellon University Silicon Valley** — Mountain View
  Master of Science in Software Engineering — *Jan. 2018 – May. 2019*
- **Nanyang Technological University** — Singapore
  Bachelor of Engineering, Electrical and Electronic Engineering; GPA: 4.61/5.0 — *Aug. 2010 – May. 2014*
- **Courses**: Foundations of Computer Systems, Data Structure and Algorithm, Practical Data Science, Foundations of Software Engineering

Technical Skills

- **Languages**: Java, Python, C, Shell, HTML, JavaScript, SQL
- **Technologies**: Android, Angular, Angular Material, Ceph, Consul, Docker, Docker Compose, Express.js, Gradle, gRPC, Google Guava, HOCON, Linux, MapReduce, MongoDB, Node.js, NumPy, Pandas, Registrator

Work Experience

- **Moqi.ai (Java, Shell)** — Beijing, China
  Software Engineer Intern — *May 2018 - Aug 2018*
  Designed and implemented fingerprint system fault tolerance that improved system uptime from 93% to 99%
  - **Ceph Filesystem**: Set up **Ceph Filesystem** with **Erasure Coding** as main distributed storage for segments cached in matching servers' memory.
  - **Fingerprint System Docker Compose**: Simplified fingerprint system testing through creating **docker compose** file to run the entire fingerprint system on a single server. The fingerprint system consists of heterogeneous **gRPC** services written in Java, Python, C++ as well as Redis, Cassandra and SeaweedFS.
  - **Matching Server Fault Tolerance**: Designed and implemented auto discovery of online/offline **dockerized** matching servers through **Consul** and **Registrator**, which triggers segment redistribution among matching servers and segment recovery from Ceph Filesystem.
  - **Controller Fault Tolerance**: Designed and implemented Controller **Active/Standby Failover** through **Consul's Leader Election**.
- **Barclays Capital Services (Java)** — Singapore
  Software Engineer — *Jun 2014 - Dec 2017*
  - **Trade Reporting Processor**: Developed Trade Reporting Processor that retrieves and applies trade reporting obligations. Implemented caching of index constituents using **Google Guava** that cut down query time by 50%.
  - **Market Object and Static Services**: Developed a greenfield market data calculation application which used **Redis** for caching, **Elasticsearch** for logging and **Akka** for concurrency.
  - **Sparta Automatic Deployment**: Fully automated Sparta deployment to 24 servers across 5 countries that used to be manual and sluggish. This consisted of migrating Sparta from **Perforce** to **Git** version control system, and implementing continuous deployment through **TeamCity** and **Nolio**.

Projects

- **Emergency Social Network (HTML, JavaScript)**: Developed a chat application designed for times of disaster that provides functions including public chat, private chat, post announcement, share status, user administration and emergency contacts. Used **Angular** and **Angular Material** for frontend, **Express.js** and **Node.js** for backend, **MongoDB** for database.
- **Online Bulletin Board (React, Redux, Tornado)**: Implemented post sorting by tag and post searching features using **React**, **Redux** and **Tornado**.
- **Face Fengshui App (Android)**: Independently developed an **Android** application that performs face recognition on a person's selfie to predict his/her fortune and temperament based on face reading techniques.
- **Malloc (C)**: Independently implemented a **dynamic memory allocator** using **segregated free list** that achieved 74.1% memory utilization and 20,000 throughput.

- **Movie Box Office Prediction (Python)**: Crawled raw html of over 8,000 movies from Box Office Mojo and parsed them into meaningful data using **lxml**. Processed 30 million pieces of movie information using **MapReduce**. Extracted features using **Pandas**, **NumPy** and **One Hot Encoding**. Trained 2 prediction models using **Linear Regression** and **Decision Tree Regression**. Compared their accuracy based on **mean squared error**.