

TERRANCE (CONGLIN) WANG

cw.terrance@gmail.com · 2302 Dwight Way, Berkeley, CA 94704 · 510-993-5921 · conglinwang.github.io

EDUCATION

University of California, Berkeley

August 2016 - May 2020

B.A. Computer Science, B.A. Data Science

Coursework: Data Structures, Computer Architecture, Intro to Database Systems, Operating Systems and System Programming
Efficient Algorithms and Intractable Problems, Intro to Artificial Intelligence, Principles and Techniques of Data Science

SKILLS

Programming Languages: Java · Python · C · Golang · SQL · HTML&CSS · Javascript · LISP · UML

Frameworks/Tools: Spring · React · Angular · Git · Docker · Vagrant · HBase · Redis · Kafka · Dubbo · JMeter · Tomcat · Guava
Commons · Lombok · MyBatis · Log4j · Maven · JUnit · Mockito · AssertJ · jQuery · D3 · Highcharts

EXPERIENCE

Software Development Engineer Intern, Alibaba Group

May 2019 - July 2019

- Interned in the AliPay merchant services team at Alibaba subsidiary Ant Financial to build efficient, scalable, reliable and extensible **Java** backend applications, under a microservices architecture that provides an easy-to-use payment solution to **over one million small businesses**.
- Designed a server-side Java application named ZMP (Z Message Pivotal), a unified message-delivery engine to simplify content delivery to end users. Implemented design patterns including **factory**, **facade** and **visitor modes** to make code extensible.
- Integrated with packages including **Apache Commons**, **Lombok** and **Log4j** to improve code maintainability and used **Maven** to manage dependencies.
- Utilized Java **ExecutorService**, internal middleware and DevOps tools including **fault tolerance**, **message queues**, **distributed databases** for higher concurrency and availability.
- Worked with teammates on the merchant VIP card project. Modeled the system with **UML**, wrote up system analysis documents, performed code reviews and fixed bugs. Improved test coverage with **JUnit** and **Mockito**.

Research Assistant, Berkeley Mechanical Systems Control Lab

September 2018

- Developed the browser-based LIDAR data visualizer used to label 3D point-cloud training data in JSON format as part of the lab's autonomous driving project, using **Javascript**, **HTML** and **WebGL** library.
- Improved user experience by supporting batch-loading JSON files and assistive labeling based on history.

Frontend Software Engineer Intern, Bi.XYZ

May 2018 - August 2018

- Developed and maintained the price chart module of the cryptocurrency exchange startup's web portal in **Javascript**, **HTML** and **CSS**, utilizing **Angular JS**, **WebSocket** protocol, **D3** and **TradingView** libraries.

SELECTED PROJECTS

Gather, a Social Network Based on Geo-index and Image Recognition

June 2019

- Designed and implemented a location-based social network web application with **React JS**.
- Implemented features for users to create posts and browse or search for nearby posts with **Ant Design**, **GeoLocation API** and **Google Map API**.
- Used token-based registration/login/logout flow with **React Router v4** and **JWT** on the server side to improve authentication.
- Built a scalable web service in **Go** to handle posts and deployed to **Google Cloud** (GAE flex) for better scaling.
- Utilized **ElasticSearch** (GCE) to support location-based searching functionality such that users could search for posts within a distance (e.g. 200km).
- Used **Google Dataflow** to implement a daily dump of posts to **BigQuery** table for offline analysis. Aggregated the data at post and user levels using BigQuery to improve keyword-based spam detection.

Pintos, An Elementary Operating System

February 2019 - May 2019

- Improved the original **thread scheduling** mechanism of the X86 Pintos educational operating system by implementing the lock priority donation and the multi-level feedback queue scheduling algorithms in C.
- Added support for running **user programs** with arguments and improved user experience by automatically resolving relative file paths.
- Implemented **process control system calls** including *halt*, *exec*, *wait*, *practice*, **file operation system calls** including *create*, *remove*, *open*, *filesize*, *read*, *write*, *seek*, *tell*, *close*, and **directory management system calls** including *chdir*, *mkdir*, *readdir*, *ids*'.

RDBMS

February 2019 - April 2019

- Built a **relational database system** in **Java** that supports SQL commands including database/table **CRUD**, **conditional selection**, **sorting** and **join** operations.
- Designed the system with **B+ Tree** indexing and **query optimization** to improve query speed.
- Implemented External Sorting, Nested Loop Join and Sort-Merge Join algorithms for faster sorting and join.
- Added support for **concurrent queries** with multi-granularity locking on different levels of resources including databases, tables, pages, and records.
- Tested the program with **JUnit** to guarantee proper handling of malformed commands, illegal values, large datasets and corner cases to prevent crash.