TERRANCE (CONGLIN) WANG

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

EDUCATION

University of California, Berkeley | B.A. Data Science *August 2016 - May 2020*

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/C++ · Golang · SQL · HTML&CSS · Javascript · LISP

Frameworks/Tools: Spring · React · Angular · Git · Docker · Vagrant · Valgrind · AWS · GCP · Tomcat · HBase · Redis · Kafka Dubbo · gRPC · JMeter · Hibernate · MyBatis · Guava · Commons · Log4j · Maven · JUnit · Mockito · jQuery · D3 · Highcharts

EXPERIENCE

Software Development Engineer Intern, *Alibaba* *May 2019 - July 2019*

- Interned in the AliPay merchant services team at Alibaba subsidiary Ant Financial (world's most valuable unicorn) to build efficient, scalable, reliable and extensible **Java** backend applications, under a microservices architecture that provides an easy-to-use payment solution to > 1 million small businesses.
- Designed a server-side **Java** application named Z Message Pivotal, a unified message-delivery utility that integrates with firm-wide sender services to simplify content delivery to business owners (clients).
- Utilized **thread pools** and implemented producer-consumer model for concurrent processing.
- Implemented design patterns including **factory, facade and visitor modes** to make code extensible.
- Integrated with packages including **Apache Commons** and **Lombok** to reduce code. Used **Maven** to manage dependencies.
- Improved project test coverage with **JUnit** and **Mockito** to over 80% and conducted smoke tests.
- Worked with teammates on the merchant VIP card project for merchant discount customization. Modeled the system with **UML**, wrote up system analysis documents, performed code reviews and fixed bugs.

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 **Angular, Javascript**, **HTML** and **CSS**, as well as **D3** and **TradingView** libraries.
- Designed and implemented the public price data API with **HTTP** and **WebSocket** protocols.

SELECTED PROJECTS

Gather, a Location-Based Social Network *August 2019 - September 2019*

- Designed and implemented a location-based social network web application with **React**. Built a scalable web service in **Go** to handle posts. Deployed the service to **Google Cloud Platform**.
- Utilized **Elasticsearch** (**GCE**) to support location-based searching such that users could search for posts within a distance.
- Used **Google Dataflow** to implement a daily dump of posts to **BigQuery** table for offline analysis.

Pintos, An Elementary Operating System *February 2019 - May 2019*

- Improved the original **thread scheduling** mechanism of the X86 Pintos educational operating system by implementing lock priority donation, priority scheduling and multi-level feedback queue scheduling algorithms in **C**.
- Used **VirtualBox** as the hypervisor and **Vagrant** to configure the virtualization environment. Used **Valgrind** for memory debugging and profiling.
- Added support for running **user programs** with arguments by implementing **process control system calls** including *halt, exec, wait, practice.* Improved user experience with automatic file path resolution.
- Added a write-back **buffer cache** to the file system to improve reads/writes speed by responding to reads with cached data and coalescing multiple writes into a single disk operation
- Implemented **file operation system calls** including c*reate, remove, open, filesize, read, write, seek, tell, close.*
- Implemented **directory management system calls** including *chdir, mkdir, readdir, isdir*.
- Conducted group design meetings and wrote up design documents, system analysis reports and test reports for each phase.

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. Used **Docker** for containerization and development environment management.
- 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 high-performance sorting and join.
- Added support for **concurrent queries** with multi-granularity locking on different levels of resources.
- Tested the program with **JUnit** to guarantee proper handling of malformed commands, illegal values, large datasets and corner cases to prevent crash.

Mevius, an Event Search and Ticket Recommendation App *October 2018 - November 2018*

- Designed an interactive web page in **HTML**, **CSS** and **JavaScript** with **AJAX** for users to search events and purchase tickets based on user location.
- Improved personalized business recommendation based on search history and favorite records.
- Created **Tomcat Java** servlets with **RESTful** APIs to handle **HTTP** requests and responses.
- Used **MySQL** to store real business data including price, location and category. Migrated to **MongoDB** for better scalability.
- Deployed to **Amazon EC2** to handle 150 QPS tested by **Apache JMeter**.