Yuqi (Gavin) Tao

yuqit1@uci.edu

Education

B.S. Computer Science Graduation Date: June 2018

University of California, Irvine

- Major GPA: 3.187 | Dean's Honor List for 4 quarters
- Specialization: Information

Relevant Coursework

Relational Database System, Data Structure and Algorithm Implement and Analysis, Dynamic Web Application, Information Retrieval, Machine Learning, Principles in Operating System Design, Computer Networks

Working Experience

Software Development Engineer at Huawei Technologies November 2018 ~ Now

- Built OSGi bundles which implements features including parsing URLs, push message to Apache Kafka component, and retrieving messages from vehicle hardware for the Cloud Internet Gateway component in Huawei's Vehicle Suit, which is a part of Huawei's OceanConnect IoT platform.
- Used multiple Linux servers to debug with log files of OSGi Bundles, Apache Kafka, and Apache Camel; located bugs among multiple web service components.

Projects

Proxy Service Providing System to Break Web Censorship (in Progress) August 2018 ~ Now

Built a website to sell proxy service which can break web censorship in China; deployed multiple servers running proxy program V2Ray .

- Deployed and configured multiple servers to run V2Ray, a proxy program similar to VPN.
- Designed and built a database with **MongoDB** to store user information, data usage information, and billing information. Created corresponding Python classes (**Object-Document Mapping**) with MongoEngine.
- Built a website running the business logic layer, including registering, logging in, buying service, changing service, and making payment, with **Python Flask**.
- Called APIs running on V2Ray servers from Flask website with **gRPC**, and thus allowed the website to add user, remove user, and query data usage information remotely.
- Frontend of the website is still in progress.

Distributed Web Crawler and Search Engine April 2017 ~ June 2017

Constructed a distributed web crawler, filtered and managed over 1,000,000 crawled webpages, and built a search engine with multiple ranking algorithms.

- Crawled among webpages within hostname ics.uci.edu, and filtered out crawler traps and invalid URLs by parsing the URLs with **regular expression**.
- Used **natural language processing techniques** including lemmatization, stopwords, and collocation to parse the corpus of the crawled webpages.
- Created an inverted index of the crawled data with multiple ranking algorithms, including **tf-idf** and **tag based weighting**. Stored the index in MongoDB.
- Built a search engine based on the index and a website to run the search engine with **Flask** and **Jinja2**, being able to respond to the query in several milliseconds.
- Performed the whole procedure of information retrieval from crawling data, managing data,

sorting data, to displaying data on a full stack website.
- Deployed the whole project on **Nginx**.

Dynamic Website in MVC Architecture April 2016 ~ June 2016

Used multiple web development techniques to construct a full stack website for shopping movies in **MVC** *architecture.*

- Parsed large XML file using **DOM** to import the XML file into a **relational database** in **MySQL**. Implemented interfaces with **JDBC** to give the Java servlets access to the database.
- Built the backend of the website with **Java servlets**. Implemented features including shopping carts and login/logout by **Session**. Provided basic JSP pages and backend interfaces to the frontend developer in the team.
- Used network security techniques including **HTTPS** to encrypt the data and user input check to prevent SQL injection.
- Built more servlets to implemented features including searching, sorting results by different keys, pagination, and accessing details of specific movies. Cooperated with the frontend developer to use **AJAX** to implement "Auto-completion Search" and "Auto popup window for each movie".
- Designed a **load balancer** to handle massive client requests in a parallel distributed system and used Jmeter to perform a stress testing on the web application. Deployed the whole project in **Tomcat** on **AWS**.

Minimal Unix Shell and Linux Commands April 2017 ~ June 2017

Wrote multiple useful Linux Bash commands for daily usage and built a minimal Unix shell.

- Implemented a trash bin in Bash, including several Bash scripts to substitute rm, e.g. saferm, undorm, lsrm.
- Implemented an enhanced edition of ls command in Bash called lss.
- Re-implemented lss in C, supporting options and arguments including -a, -A, -l, and -L).
- Implemented a parser to parse command line input. Implemented a minimal Unix shell in C. Implemented the feature of command and script execution.
- Implemented the feature of input/output redirection (< and >) and appending (>>) by using **multiple pipes**. Implemented the feature of recognizing comments, error detection and handling.
- In system level, implemented cd command. Using signal handler to implement the feature of exiting with exit code (e.g. Ctrl+C).

Machine Learning: Rainfall Prediction January 2017 ~ March 2017

Built a hybrid data prediction model based on multiple machine learning algorithms to predict the rainfall possibility.

- Filtered, cleaned, and used 100,000 samples of training data on 14 features to train the model.
- Used multiple training algorithms including **K-Nearest Neighbor**, **Random Forest**, and **Gradient Boosting** to train the prediction model.
- Used **ensemble techniques** including **bagging** and **weighted average** to combine the prediction of different training models.
- The Kaggle score of this hybrid machine learning model beat 86% students in school wide championship.