Zeping He

zepinghe@umd.edu | https://github.com/finallyegg | 716-580-0767

Education

Master of Science, Computer Science *Expect Dec 2021*
University of Maryland College Park, College Park, MD

Bachelor of Science, Geographical Information Science *May 2019*
University at Buffalo, The State University of New York, Buffalo, NY

Skills

Programming Languages: *Java, Python, JavaScript, Go, C/C++, Swift* Support Tools: *Docker, Git, Bash*

Back-end: *Flask, Express.js, AWS Lambda, DynamoDB, Google Firebase* Front-end: *React.js, Node.js*

Experiences

Software Engineer Intern | *Yahoo!* *June 2021 – August 2021*

- Designed and implemented a text parser using **Java Maven** ANTLR library to extract desired information from JSON and ECMAScript files
- Developed recommendation models using **PySpark** ALS, KMeans Clustering and Binary Classification model and achieved root mean squared error of 0.3

Graduate Research Assistant | *University of Maryland Software Engineering Group* *January 2021 – Present*

- Refactored live-update functions in decision making project by adding **web-Socket** architectures, improved user experience and set up as base for other components
- Designed and developed a reference utility using React and Flask that records tech stacks and code sample as an experiment base, coding in **JavaScript** and **Python**

Software Engineer Intern | *Jasper Financial* *May 2020 – August 2020*

- Built user registration and fraud verification api through REST API in **AWS** API-gateway, Lambda and DynamoDB, served as a critical part of web app with low latency
- Integrated third party APIs into data pipeline and documented API usage and revised functions based on team need

Projects

BitTorrent Client – *Python based BitTorrent client can be corporate with other BitTorrent clients* *May 2021*

- Designed and build a python-based app supports downloading, updating, and sharing files
- Used multiple threads to download different file chunks simultaneously from peer clients
- Supported both **TCP** and **UDP** protocol to maximize operation efficiency
- Implemented **optimistic unchoking**, **rare-first** strategy and **endgame** mode to have the best performance
- Achieved the highest project score among the Computer Network class

My storage - *Simple key-value storage mounted on OS with versioning* *Dec 2020*

- Built a key-value storage in **Go** to split file into dynamic size blobs and save them in local SQLite DB
- Supported **CURD** operation through messaging between client and server using **HTTP** request
- Constructed file tree at server-side then added **sync** and **comparison** operation to replicate at lower cost
- Added security layer by using **RSA key pair** to verify client's signature before writing
- Extended client usage to Linux file system via **FUSE** interface. Supported file archive and time travel by chaining

Post - Comment App - *experimental project using **React.js**, **Express.js** and **Firebase*** *Sept 2020*

- Built user services, handled authentication via middleware and local store which uses JWT token and use middleware to embed in header section of HTTP POST to identify themselves
- Stored posts and comments in separate tables and used scanning to fetch all comments in a post to ensure performance
- Wrote frontend using React.js, support page updates through **React Hook** and **Redux**

Relevant Coursework

Computer Network, Mobile Programming, Distributed System, Database Architecture and Implementation