Sicong Liu

siconl3@uci.edu • (949) 232-6970 • Creston Ln,San Jose, CA 95122
https://github.com/lsclovecode https://www.linkedin.com/in/sicong-liu-8a300b62

Education

University of California, Irvine *Irvine, CA*
- M.S. in *Computer Science* | GPA: **3.86/4.00** *Sep. 2016 - Mar. 2018*

Western University *London, Canada*
- M.S. in *Engineering Science* | GPA: **3.90/4.0** *Sep. 2014 - Jun. 2016*

Skills

- **Programming Languages:** Java, Python, JavaScript, Scala, HTML, CSS, SQL, LaTeX
- **Platform and Tools:** AWS EC2, Google Cloud, Linux, Flask, NodeJS, AngularJS, Bootstrap, MySQL, MongoDB, Spark, Kafka, Hadoop, Redis, Cassandra, Scikit-Learn, RabbitMQ, ElasticSearch, Maven, Spring, Spring Boot, Docker, Git

Work Experience

Full Stack Engineer Intern, Amazon *Seattle, WA*
Catalog Quality team *Jun. 2017 - Sept. 2017*
- Optimized previous UI by merging separated web pages into one single page with modals using **AngularJS**, **Node.js**, **AJAX**, **HTML5**, **Bootstrap** with unit tests using **Jasmine** and **Karma**.
- Developed a data management platform for storing and updating task information using **Spring** Framework, **MyBatis**, **RESTful API** with **AWS RDS** as database.
- Developed new features for task management of catalog correction, an inner web tool for merging duplicate items, so that users can automatically see assigned tasks by managers ordered by priorities after login, used by thousands of Amazon catalog quality associates.

Backend Software Engineer Intern, Xiaohongshu(RED) *Shanghai, CN*
Supply Chain Platform team *Apr. 2018 - Aug. 2018*
- Built a supply chain management system by using MVC framework with **Flask**, **MongoDB**, **MongoEngine**, **RabbitMQ** and **ElasticSearch**, and realized features like real-time replenishment and provider ranking.
- Efficiently implemented back-end services based on Microservices structure and RPC framework using **Apache Thrift** and incorporated RabbitMQ as message queue to decouple back-end microservices.
- Developed an optimized simulated annealing algorithm for provider bidding, which reduced **10 %** purchase cost for the company.

Project Highlights

Big Data Platform for Real-time Stock Price *Jan. - Mar. 2018*
- Implemented a high-performance data processing platform using **Apache Kafka**, **Apache Cassandra**, **Redis** and **Apache Spark Streaming** to analyze stock data.
- Optimized payload size using **Google Protocol Buffer** to improve system throughput by 30%.
- Developed a dashboard web app using **Node.js**, **D3.js** and created a scalable cloud deployment environment using **Docker**.

Smart Zillow *Sep. - Dec. 2017*
- Implemented a real estate search website and value prediction system using Service-oriented Architecture.
- Implemented a distributed real estate **web scraping system** together with Zillow API to collect real-time property information using **MongoDB** and **RabbitMQ**.
- Developed the web service with **cache** using **Node.js/Express**, **Redis** and **Bing Map API**.
- Utilized **TensorFlow** to predict real estate value by linear regression.

Real-time Running Location Monitoring System *May - Jun. 2017*
- Designed and developed a real-time car location monitoring system using **Java**, **Spring Boot**, **Spring Data**, **Spring Cloud**, **Maven**, **JPA**, **Hibernate**, **Tomcat**, **RabbitMQ**, **MongoDB**, **WebSocket**, **JavaScript**, **Bootstrap**
- Effectively implemented server side REST APIs such as car location simulator and persistence handler using **Spring Data**, **Spring Boot** and **Spring MVC**.
- Developed the single page front-end to integrate with backend using HTML, CSS, JavaScript, REST and **WebSocket** to show real-time location changes.

Elastic Distributed Databases *Feb. - Apr. 2017*
- Developed an elastic distributed database using **MySQL** replication and used it to serve as the backend for a multi-tier web application running **TPC-W benchmark**.
- Developed a **load balancer** to route the browsing queries to master and route the ordering queries to slaves in a **round-robin** manner
- Monitored the workload/database/operating system and demonstrate metrics using **CanvasJS**.