

Zhuo Jia

Portfolio: <http://www.zhujia7.com>

Email: jiazhao0528@yahoo.com

Education

Duke University

North Carolina, USA

Master of Science, Computer Engineering, GPA: 3.72/4.00

Aug.2014 - May.2016

Courseworks: C++, Algorithm, Distributed System, Operating System, Network Architecture, Computer Vision.

Chongqing University

Chongqing, China

Bachelor of Science, Electrical Engineering, GPA: 3.56/4.00

Sept.2010 - July.2014

Selected Awards: National Academic Scholarship (Top 2%), Meritorious Winner Award in 2013 MCM/ICM of USA.

Technical Skills

- **Programming languages:** C++, C, Python, Go, JavaScript.
- **Web development:** HTML, CSS, React.js, Meteor.js, Node.js, Bootstrap.
- **Database:** SQL, MongoDB, SAP HANA, Redis.
- **System/tools:** Linux, Git, AWS, Docker, Heroku, Nginx, gdb, Valgrind, gRPC, Google Test, etc.

Working Experience

Software Developer, SAP Labs, Bay Area, CA, USA

Nov.2017 - Present

- Working for Native Disk Storage project of SAP HANA database internal, and the mission is to reduce memory footprint of big volume data by using page load technology for disk storage.
- Designed efficient search algorithm in page-loadable data vector and implemented table load unit conversion.
- Extended paged capability for compressed attributes with different compression algorithms, and improved optimized compression with in-place writer, to avoid unnecessary copying of some storage components.
- Responsible for daily git branch management and being a component code reviewer to help product delivery.

Software Developer, SAP Labs, La Crosse, WI, USA

June.2016 – Oct.2017

- Contributed defensive and testable code to extend ability of SDQ (Smart Data Quality) and parallel task framework back-end of **SAP HANA**, an in-memory computing database platform.
- Submitted bug fixes and feature enhancements for HANA using **C++11** and **SQL**. Conducted **Google C++ unit tests** and **Python** E2E tests, increasing test coverage from 50% to 88% of SDQ component.
- Improved around 45% time efficiency for a multithread scenario of SDQ by finding hotspots to optimize lock.

Projects

Tensor News

Feb.2017 – Apr.2017

- Built a news scrapper and a message pipeline individually based on **micro web services**, which are implemented in **Python** and **RPC**, and also used **Redis** as cache, **Rabbit MQ** as message broker queue, and **MongoDB** as database.
- Designed a time-decay user preference model and log processors to update user preference based on user behaviors.
- Implemented a 2-layer CNN news classifier with **Tensor Flow** and NLP techniques, reaching about 81% accuracy.
- Designed a news recommend system based on above preference model and trained news classifier, using front end skills like **React.js**, **Node.js**, **Express**, and also enabled features including user sign up/sign in, lazy-load news, etc.
- Deployed the app to **AWS** with **Nginx** as caching reverse proxy, and kept the app alive with PM2 process manager.

Duke Flea Market, <https://dukemarket.herokuapp.com>

Apr.2016 – May.2016

- Designed an eCommerce web app individually for school flea market based on MeteorJS stack.
- Implemented responsive UI with **HTML**, **CSS**, **Bootstrap** and Spacebars HTML template engine.
- Enabled multiple features mainly including user credential, post/edit/delete of information, comments, like button, collection button, and social media integration with **JavaScript** and **MongoDB**.
- Optimized Pub/Sub service, used content delivery network (CDN) to speed static assets and deployed it to Heroku.

Hola Chat, <https://holachat.herokuapp.com>

Feb.2016

- Implemented a real-time chat app for code interview based on **Node.js**, **Express**, **Socket.io**, and **Redis**.
- On right side, built a collaborative code editor showing different user cursors and their real-time typing actions, and applied **Redis** to restore snapshot of editing contents so that new users in the room can catch up editor context.
- Deployed and host it on Heroku.

Raft, Distributed System Course Project

Apr.2015

- Implemented Raft distributed consensus protocol, including leader election and log maintenance process.
- Minimized and resolved split-votes issue by using randomized election timeout in leader election.
- Maintained consistency between logs on different servers through 2-phase commit process in master-slave system.

Chord, Distributed System Course Project

Mar.2015

- Implemented a scalable P2P lookup protocol in C++ and minimized rehashing when arbitrary nodes join or leave.
- Reduced disruption of nodes joining and leaving the system concurrently through using consistent hashing.
- Achieved query in **O(logN)** time cost and high probability to keep load balance with “finger table”.