Haibin Zhang

1780 Broadway St., Ann Arbor, MI, 48105
(+1) 734 239 3069
haibinzh@umich.edu
harbinfate.com

EDUCATION

Master in Computer Engineering, GPA: 3.00/4.00 August 2016 - May 2018
University of Michigan
Operating Syst, Computer Architec, Web Dbase, Algorithms, Database Mgt Syst.

B.S. in Electronic Information Engineering, GPA: 3.69/4.00
University of Electronic Science and Technology of China(UESTC)

SKILL

Skill*: Python, Java, shell, C, C++, SQL, HTML, JavaScript, Matlab, LaTeX.*
Environment*: Linux, Mac OS, Windows.*

PROJECT

Info about Linkedin: *Crawler* *Winter 2017*

- Build a system providing suggestions about which linkedin user is more likely to push resume for me.
- Multi-process crawlers. One master assigns jobs to several workers, and each worker uses its own account to sign in and save given users' profile into MongoDB. Analysis data via MapReduce.
- Fault-tolerance: When errors accumulated to a certain number, the master will shutdown workers, save URL in the master into disk, record current environment info into log file.

Operating System: *Projects of Operating System* *Winter 2017*

- Threads: Design a concurrent Thread Library for multi-processor multi-thread applications. Clock and Least Recently Used algorithm are implemented. CPU goes sleep or wake up depending on current thread's request.
- Memory Manager: Design a pager that manages application processes' virtual address spaces, including swap block and file block. Zero-pin page, Copy-on-write and some performance optimization are implemented.
- File Server: Design a multi-threaded, secure network file server. Client processes will interact with it via network messages to access and modify their data. Access without permission will be abandoned.

Web and Database: *Projects of Web and Database* *Fall 2016*

- Online Photo Service(Web Dbase): Building a web server which users can visit public and personal albums, and upload new photos to these albums. 'Sign in' and 'Sign up' options are provided.
- MapRedcue(Web Dbase): Implementing a Hadoop-like MapReduce system, with master and worker nodes for map-reduce operations over large datasets, with a distributed file system, and fault tolerance to address datanode failures.
- Database: Build a Relational database with sqlplus. Use JDBC to get data from database. Export data from Oracle to MongoDB, and use JavaScript to operate data with MongoDB.
- Database: Implement a single-threaded version of ARIES to recover from crash. Page flush is considered.