Xinghao Zhou

Email: **xinghaoz@andrew.cmu.edu** Phone: **(650)625-7120**
Linkedin: **www.linkedin.com/in/xinghaoz** Address: **670 Ashton Ave, Palo Alto, CA, 94306**
Personal Website: **xingz.me** Job preference: **Software Development Engineer**

Education

Carnegie Mellon University, USA
- *Master of Science*: **Electrical and Computer Engineering** *Sep. 2015 – Dec. 2016*

University of Electronic Science and Technology of China (UESTC), Chengdu, China
- *Bachelor of Engineering*: **Computer Engineering** *Sep. 2011 – Jul. 2015*
- GPA: 3.86/4.0; Rank: 18/619

Skills

- *Computer Languages:* Java, Python, C++, C, bash, awk, JavaScript, Jquery, HTML, CSS, Scala
- *Database:* MySQL, PostgreSQL, MongoDB
- *Familiar Areas:* Cloud Computing, Distributed Systems, Zookeeper, Spark, Map/Reduce, Hadoop, AWS, Web Development, Machine Learning, Operating System

Professional Experience

Use Apache ZooKeeper to Build Distributed Applications *Jan. 2017 – present*
- Group membership and name services: Implemented a naming system as a replacement for DNS within the cluster. Joined nodes are recorded in a group path. Nodes that go down are automatically removed from the list, thus cluster always has an up-to-date directory of the active nodes.
- Centralized configuration manager: Use ZooKeeper to store the cluster's configuration. Whenever the cluster's configuration changes, nodes in the cluster can be notified immediately thus they can update the configuration in real time.
- Distributed mutexes: Implemented distributed locks which can avoid Herd Effect.

AWS-Based Tweeter Analytics Web Service *Sep. 2015 – Dec. 2015*
- Developed ETL process over a 1-TB data set by using **Hadoop Map/Reduce** and then validated the results.
- Developed a scalable RESTful web application in Java Servlet and Apache HTTP Server. Distributed the workload evenly among multiple EC2 instances by using a **load balancer**. This system can handle up 15,000 request per minute.
- Optimized the MySQL database by applying **partition** technique, modifying the **cache** and refining the **schema**, which result in increasing the throughput by 25% compared to the previous design.

Building a Multi-player Online Web Game by Django *Oct. 2016 – Dec. 2016*
- Developed a web application by Django which included the implementation of a user management system, the use of web socket (channels) and the use of the new features in HTML5.
- Used web socket to achieve real time communication among web users. Implement a mutual exclusion mechanism (Token passing in a ring structure) to handled concurrency and consistency issues.
- Developed a game engine to provide the Object Oriented Programming support in Javascript for creating the characters, maps and events.
- Used **Nginx** as the server to deployed it in AWS **EC2** instances.

P2P Application – Peer to Peer Pair Programming (P2P3) *Feb. 2016 – May. 2016*
- Built an collaborative coding tool which enabled multiple users to edit a same file concurrently upon a **Peer-to-peer network**, using **Python** as the implementation language.
- Structured the network as a **ring**. All the nodes and objects could be accessed through a 128-bit **GUIDs** which generated by SHA-1 hash. Applied **Pastry**, a prefix routing approach as the routing algorithm.
- Implemented **woot**, which is an algorithm that inherently ensures reliable merge of documents.
- Provided the support for online compiling of some languages (C, C++, Lua, Python, and Ruby.)

Movie Recommendation System Implemented By Using Spark *Mar. 2016 – Apr. 2016*
- Using over 500 000 user's rating of movies from Movielen as the data sample, divided it into training set, validation set and test set.
- Applied **ALS** algorithm and AWS EMR to run **Spark** to analyze these sets of data.