

Yigong Hu

Curriculum Vitae

Department of Computer Science
Johns Hopkins University
500 W University Parkway 9s,
Baltimore, MD 21210
✉ yigongh [at] gmail.com

Research Interests

System Reliability, Mobile System, Distributed System

Education

- 8.2017- Ph.D. Student. Computer Science, *Johns Hopkins University*, Baltimore, MD, USA.
 - **Advisor: Ryan Huang.**
- 9.2013-6.2017 B.E. Computer Science, *Huazhong University of Science and Technology*, Wuhan, Hubei, China
 - **Advisor: Hai Jin.**
 - GPA: 91.75/100, **Rank: 1/258**
 - Core Courses: C programming, Object Orient Programming, Data structure, Algorithm, Assembler Language, Digital Circuit Design, Operation Research
- 6.2016-8.2016 Visiting Student. Computer Science, *University of Chicago*, Chicago, IL, USA.
 - **Advisor: Shan Lu.**
- 9.2015-5.2016 Visiting Student. Computer Science, *University of Michigan*, Dearborn, MI, USA
 - GPA: 4.0/4.0
 - Core Courses: Computer Network and Distributed system, Operating System, Database, Computer Organization, Compiler Design, Computer Architecture

Awards and Honors

- 5.2015 Outstanding Undergraduates Exchange Fellowship, China Scholarship Council
 - Award ratio: Top 0.01% of undergraduate students in China.
- 10.2014 National scholarship, Ministry of Education of China
 - Award ratio: Top 0.2% of undergraduate students in China.
- 10.2015 Outstanding student scholarship, Huazhong University of Science and Technology
 - 1 of 8 awarders among 258 students.
- 10.2014 Outstanding student scholarship, Huazhong University of Science and Technology
 - 1 of 8 awarders among 258 students.
- 10.2013 Outstanding Freshman Scholarship, Huazhong University of Science and Technology
 - 1 of 15 awarders among 258 students.

Research Experience

- 6.2016-8.2016 Research Assistant, *University of Chicago*, Chicago, IL, USA.
 - Advisor: Shan Lu.**
 - OMAS, a tool for analyzing applications built on ORM(Object Relational Mapping) framework.*
 - Worked with professor Shan Lu and Ph.D student Zeiwei Chu
 - Motivation: ORM based applications are vulnerable to performance problems.
 - Solution: Our projects built a tool call OMAS using Ruby language on Linux system which can examines the application logic and its interaction with databases via the Ruby on Rails ORM framework and analyze the executes information of the application dynamically by generating data for Rails applications. We tested our tool across 26 open-source applications

- Outcome: Our results show that many applications expose the same or similar performance problem patterns, which can help developers to diagnose and fix performance problems.

6.2016-8.2016 Research Assistant, *University of Chicago*, Chicago, IL, USA.

Advisor: Shan Lu.

Bug study in populate open source distribute system

Worked with professor Shan Lu and Ph.D student Wei Yuan

- Motivation: Concurrency bugs lurking in distributed software infrastructures manifest non-deterministically, and hence are extremely difficult to detect, diagnose, and fix.
- Solution: We manually studied 72 distributed concurrency bugs from five widely used and important distributed systems: Hadoop MapReduce, Hadoop Distributed File System, HBase, Cassandra, and ZooKeeper. For each bug in our study, we did the following work: identifying the error symptom from the log and the relationship among errors; studying what information is required for debugging, and what is provided by the log; analyzing the patch based on different fixing types
- Outcome: Our study provides researchers with more than 72 thoroughly taxonomized DC bugs and many fixed patterns to ease diagnosis and fixing.

9.2015-3.2016 Research Assistant, *Huazhong University of Science and Technology*, Wuhan, Hubei, China

Advisor: Hai Jin.

Protect share library by isolating share library and providing interfaces

Worked with professor Deqin Zou and Ph.D student Pan Zhang

- Motivation: State-of-the-art Code Reuse Attack Protection Techniques usually consume a large amount of physical memory. We need a compromise between security and practicability.
- Solution: We built a tool which did the following work: dividing processes into two domains—application domain and library domain; building interfacing and callback function for library domain using C language; application domain need to use specific interfaces to call functions of library domain.
- Outcome: Our project reduces the memory usage with little damage to the security ability

Computer Skills

- Programming

Mainstream: C, C++, Java, Python, Ruby

Assembly: x86, MIPS, 68000

Hardware: Verilog

- Platform

Hardware: Logisim, ISE

Web: Ruby on Rails

System: Linux Ubuntu

Mathmatic: Matlab