

Studying Cross-component Bugs in Cloud System

- **Project proposer:** You-Liang Huang
- **Open source?** Yes
- **Mentors:** You-Liang Huang <yliangh@bu.edu>
- **Additional Advisors:** n/a

Preferred past experience:

- **Required (for most or all team members):** Linux, Java
- **Required (at least one team member):** Java, Containers/Docker, Hadoop, Program analysis
- **Nice to have:** Spark, Yarn, Kafka, Hive, Flink, Soot/Sootup

Project background:

Modern cloud systems are orchestrations of independent and interacting (sub-)systems, each specializing in important services (e.g., data processing, storage, resource management, etc.). Hence, cloud system reliability is affected not only by the reliability of each individual system, but also by the interplay between these systems. In recent days, this type of bugs has emerged and became one of the major threats to the cloud system. For this project we focus on studying real bug cases that existed in Apache Hadoop software library and devising a tool that can detect potential bugs of certain types.

The following is a typical example of a bug that happens in the interplay of two components in a cloud system. In Hadoop, when using command “spark-commit” with a remote HDFS file to submit a Spark job, Spark downloads the remote .jar to the local disk but launch the job with remote .jar. However, in the buggy version, i.e., Spark-2.2.0, Spark performs an unnecessary (unexpected) upload that re-transfers the .jar in local to the remote. This bug is very difficult to detect through traditional tests since it involves the interplay of two software. Therefore, it calls a new bug detection technique specifically designed to improve computer system reliability.

Project description:

1. Read and study bug descriptions/reports.

2. Install software of exact versions and write scripts to reproduce bugs.
3. Read source code from the specific version, with the help of the bug reports and reproduction scripts, try to find the root cause of bugs.
4. Repeat step 1-3 to study multiple bugs to generalize bug patterns.
5. ****Stretch goal****. Develop a program analysis tool that can detect potential bugs.

What team members will learn:

- DevOps and Environment Management
- Advanced Debugging & Root Cause Analysis
- Program Analysis (The Stretch Goal)