Qingzhou Luo

PhD Student
Department of Computer Science
University of Illinois at Urbana-Champaign
2107 Siebel Center, 201 N. Goodwin Ave, Urbana, IL, USA
+1-217-778-8942
qluo2@illinois.edu
http://mir.cs.illinois.edu/~qluo2

Research Interests

Testing and verification of concurrent programs in particular and program analysis and verification techniques applied in Software Engineering in general

Education

University of Illinois at Urbana-Champaign, USA August 2010 - Now (Ongoing)
PhD Student in Computer Science
Working with Prof Darko Marinov and Prof Grigore Rosu
Area of Study: Software Engineering and Programming Languages, focus on testing and verification of multithreaded programs

Shanghai Jiao Tong University, Shanghai, China August 2007 - March 2010
M.S. in Computer Science
Worked with Prof Jianjun Zhao
Thesis Topic: Concurrent bugs reproduction and detection based on replay analysis

Shanghai Jiao Tong University, Shanghai, China September 2003 - June 2007
B.E. in Software Engineering

Working Experience

Google Software Engineering Intern
Mountain View, California May 2014 - August 2014
Worked with **John Micco** on implementing features for handling flaky tests on Maven and Jenkins, similar to Google TAP. Patch to Maven Surefire has been accepted and a Jenkins plugin has been released.

Google Software Engineering Intern
Mountain View, California May 2013 - August 2013
Worked with **John Micco** on Google's continous integration and testing infrastructure. Implemented a few new features to mitigate the impacts of flaky tests. All the code have been shipped and used at Google internally.

Microsoft Research Silicon Valley Research Intern
Mountain View, California May 2012 - August 2012
Worked with **Mihai Budiu** to parallelize a large data analyzing tool to improve its performance and scalability. The new implementation is distributed both across multi-cores on a single machine and multiple machines in a cluster, and able to load larger data sets with significantly faster speed.

Software

Publicly released software from research projects and internships:

Jenkins Plugin: Flaky Test Handler
`https://wiki.jenkins-ci.org/display/JENKINS/Flaky+Test+Handler+Plugin`

Contribution to Maven Surefire: rerunFailingTestsCount option
`https://github.com/apache/maven-surefire/commit/fefaae7f0534a59f52c046a64c96987e8561dd48`

IMUnit: Improved unit testing of multithreaded programs
`http://mir.cs.illinois.edu/imunit`

ReEx: Re-execution based exploration of multithreaded programs
`http://mir.cs.illinois.edu/reex`

JavaMOP: Runtime Verfication Framework for Java
`https://github.com/runtimeverification/javamop`

GitHub Accounts:
`https://github.com/seriousamlqz`
`https://github.com/qingzhouluo`

Past Projects

Handling Flaky Tests on Maven and Jenkins
Google
Worked on implementing features for handling flaky tests on Maven and Jenkins, similar to features in Google TAP. This includes adding option in Maven Surefire to re-run failing tests and output corresponding reports, and a Jenkins plugin for collecting, displaying and triggering re-run of failed builds. Patch to Maven Surefire has been accepted and a Jenkins plugin has been publicly released.

Mitigating the Impacts of Flaky Tests
Google
Worked on Google's continous integration and testing infrastructure. Implemented a few new features to mitigate the impacts of flaky tests, including collecting historic information to skip/prioritize flaky tests and reducing flake detecion load. All the code have been shipped and used at Google internally.

Distribution and Parallelization of a Data Analyzing Tool
Microsoft Research Silicon Valley
Implemented a parallelized version of a large data analyzing tool to improve its performance and scalability. Used Microsoft TPL to achieve the parallelization, and WCF to communicate between multiple machines.

IMUnit: Improved Multithreaded Unit Testing
University of Illinois at Urbana Champaign
A framework to specify and enforce thread schedules when unit testing multithreaded Java programs. Translated over 200 existing sleep-based tests with IMUnit and achieved over 3x speedup.

CAPP: Change Aware Preemption Prioritization
University of Illinois at Urbana Champaign
Proposed a framework to prioritize thread schedules in state space exploration to find multithreaded regression bugs faster when program evolves, based on changes between two program revisions. Implemented several heuristics in Java PathFinder (JPF) and participated in the implementation of a re-execution based framework to enumerate all possible thread schedules.

EnforceMOP: Enforcing Properties in Multithreaded Programs
University of Illinois at Urbana Champaign
A framework to specify and enforce properties (written in various logic formalisms) in multithreaded programs by controlling thread schedules. Implementation is based on generation of AspectJ code to control threads.

Course Project for Compiler Construction
University of Illinois at Urbana Champaign
Used LLVM with other tools to implement various stages in a compiler: code parsing, intermediate code generation, several optimization passes and register allocation on a MIPS 32 architecture.

ConCrash: Making Concurrent Failures Reproducible
Shanghai Jiao Tong University
Implemented a framework to generate test cases (by recording info in method entries using ReCrash) together with thread schedules (by recording global order of accesses to global variables) to reproduce failures in multithreaded Java programs when they occur. Implementation is based on Java bytecode instrumentation with SOOT.

Publications

[1] **An Empirical Analysis of Flaky Tests**
Qingzhou Luo, Farah Hariri, Lamyaa Eloussi and Darko Marinov
22nd ACM SIGSOFT International Symposium on the Foundations of Softwar Engineering (**FSE 2014**)

[2] **RV-Monitor: Efficient Parametric Runtime Verification with Simultaneous Properties**
Qingzhou Luo, Yi Zhang, Choonghwan Lee, Dongyun Jin, Patrick O'Neil Meredith, Traian Florin Serbanuta and Grigore Rosu
14th International Conference on Runtime Verification (**RV 2014**)

[3] **ROSRV: Runtime Verification for Robots**
Jeff Huang, Cansu Erdogan, Yi Zhang, Brandon Moore, **Qingzhou Luo**, Aravind Sundaresan and Grigore Rosu
14th International Conference on Runtime Verification (Tool Paper) (**RV 2014**)

[4] **EnforceMOP: A Runtime Property Enforcement System for Multithreaded Programs**
Qingzhou Luo and Grigore Rosu
22nd International Symposium on Software Testing and Analysis (**ISSTA 2013**)

[5] **Efficient mutation testing of multithreaded code**
Milos Gligoric, Vilas Jagannath, **Qingzhou Luo** and Darko Marinov
Software Testing, Verification and Reliability (**SVTR**), 23(5):375-403, Aug. 2013

[6] **Ballerina: Automatic Generation and Clustering of Efficient Random Unit Tests for Multithreaded Code**
Adrian Nistor, **Qingzhou Luo**, Michael Pradel, Thomas R. Gross and Darko Marinov
34th International Conference on Software Engineering (**ICSE 2012**)

[7] **Improved Multithreaded Unit Testing**
Vilas Jagannath, Milos Gligoric, Dongyun Jin, **Qingzhou Luo**, Grigore Rosu and Darko Marinov
8th Joint meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on the Foundations of Software Engineering (**ESEC/FSE 2011**)

[8] **Automatic GUI Refactoring and Test Script Repair**
Brett Daniel, **Qingzhou Luo**, Mehdi Mirzaaghael, Danny Dig, Darko Marinov and Mauro Pezze
1st International Workshop on End-to-End Test Script Engineering (**ETSE 2011**, co-organized with ISSTA 2011)

[9] **Change-Aware Preemption Prioritization**
Vilas Jagannath, **Qingzhou Luo** and Darko Marinov
20th International Symposium on Software Testing and Analysis (**ISSTA 2011**)

[10] **A Lightweight and Portable Approach to Making Concurrent Failures Reproducible**
Qingzhou Luo, Sai Zhang, Jianjun Zhao, Min Hu
13th International Conference on Fundamental Approaches to Software Engineering (**FASE 2010**)

Presentations

Automatic GUI Refactoring and Test Script Repair
At ETSE, Toronto, Canada, July 2011

A Lightweight and Portable Approach to Making Concurrent Failures Reproducible
At FASE, Paphos, Cyprus, March 2010

Skills

Languages: Java, AspectJ, C/C++, C#, Shell Script, LaTex
Tools: Java PathFinder, ASM, SOOT, LLVM, Jenkins, Maven, Git