Rashmi Mudduluru

rashmi4@cs.washington.edu
PhD student
CSE2 380, Paul G. Allen School of Computer Science & Engineering

Research Interests

My research interests lie broadly in the areas of Software Engineering, Programming Languages and Verification.

Education

PhD Candidate, Computer Science and Engineering, *University of Washington*, Seattle.
CGPA: 3.7/4, Advisor: Prof. Michael Ernst

M.Sc(Engg), Computer Science, *Indian Institute of Science*, India.
Received 2015, CGPA: 6.3/8, Advisor: Dr. Murali Krishna Ramanathan
Thesis: Efficient Instrumentation for Object Flow Profiling

Conference Publications

- Rashmi Mudduluru, Pantazis Deligiannis, Ankush Desai, Akash Lal, Shaz Qadeer **Lasso detection using partial-state caching**, Formal Methods in Computer-Aided Design (FMCAD), 2017.
- Rashmi Mudduluru, Murali Krishna Ramanathan, **Efficient Flow Profiling for Detecting Performance Bugs**, 25th International Symposium on Software Testing and Analysis (ISSTA), 2016.
  [Received the **ACM SIGSOFT Distinguished paper award**].
- Pantazis Deligiannis, Matt McCutchen, Paul Thomson, Shuo Chen, Alastair F. Donaldson, John Erickson, Cheng Huang, Akash Lal, Rashmi Mudduluru, Shaz Qadeer, Wolfram Schulte, **Uncovering Bugs in Distributed Storage Systems during Testing (Not in Production!)**, 14th USENIX Conference on File and Storage Technologies (FAST), 2016.
- Monika Dhok, Rashmi Mudduluru, Murali Krishna Ramanathan, **Pegasus: Automatic Barrier Inference for Stable Multithreaded Systems**, International Symposium on Software Testing and Analysis (ISSTA), 2015.
- Rashmi Mudduluru, Murali Krishna Ramanathan, **Efficient Incremental Static Analysis Using Path Abstraction**, 17th International Conference on Fundamental Approaches to Software Engineering (FASE), 2014.

Professional Experience

Research Fellow, *Microsoft Research, India*, Bangalore, 2015–2017.
Mentor: Dr. Akash Lal

Research Projects

- Built a type system that can express and verify determinism properties of sequential programs. The key ideas in the type system are type qualifiers for nondeterminism, order-nondeterminism, and determinism; type well-formedness rules to restrict the typings for collections; and enhancements to polymorphism that improve precision when analyzing collection operations. We implemented our type system for Java. Our type checker warns if a program is nondeterministic or verifies that the program is deterministic. In a case study of a 24,000-line software project, it found previouslyunknown nondeterminism errors in a program that had been heavily vetted by its developers.
- Created a runtime analysis that identifies liveness bugs in distributed systems. The analysis detected repeating states in a given execution trace that violated a liveness property. Several features of the tool were designed with the aim of keeping the runtime overheads at a minimum. The tool identified real bugs in production code like Microsoft Azure storage vNext System.
- Implemented a dynamic data race detector for programs written in P#, a DSL for developing and testing asynchronous systems using the happens before algorithm. I tracked all runtime memory reads and writes with the help of Extended Reflection, a library that facilitates dynamic instrumentation by inserting callbacks for MSIL instructions.
- Designed and built a profiler that tracks the precise data path taken by objects in Java programs. I implemented this tool in Java on top of the Calfuzzer/Soot framework. This tool helped identify performance issues caused by memory bloat in several Java programs. Fixing these issues significantly improved the performance of the programs under test.

Graduate Course Projects

- Built an interactive night light with Arduino that responds to physical inputs such as ambient light and a pressure sensor.
- Built an interactive data visualization tool in d3 that summarizes the relationship between pairs of genome structures as part of a three member team.
- Built a code search tool that matches a given input query on the parse tree of the source code. This was done in a team of three.
- Implemented a static null pointer dereference analysis tool for Java programs using the Soot bytecode analysis framework.

Awards and Scholarships

- ACM SIGSOFT Distinguished paper award for the paper *Efficient Flow Profiling for Detecting Performance Bugs* at ISSTA 2016.
- Received the PLMW scholarship for attending POPL 2015.

Service

- Teaching Assistant for the under graduate course *Data Structures and Algorithms* at the University of Washington.
- Teaching Assistant for the under graduate course *Software Engineering* at the University of Washington.
- Teaching Assistant for the graduate course *Automata Theory and Computability* at the Indian Institute of Science.
- Served in the Artifact Evaluation Committees of PLDI 2017, ISSTA 2017, and ISSTA 2018