



BEN BLUM

Practical Concurrency Testing or: How I Learned to Stop Worrying and Love the Exponential Explosion

Wednesday, October 17, 2018 – 12:00 p.m. – GHC 4405

Concurrent programming presents a challenge to students and experts alike because of the complexity of multithreaded interactions and the difficulty to reproduce and reason about bugs. Stateless model checking is a concurrency testing approach which forces a program to interleave its threads in many different ways, checking for bugs each time. This technique is powerful, in principle capable of finding any nondeterministic bug in finite time, but suffers from exponential explosion as program size increases. Checking an exponential number of thread interleavings is not a practical or predictable approach for programmers to find concurrency bugs before their project deadlines.

In this thesis, I develop several new techniques to make stateless model checking more practical for human use. I have built Landslide, a stateless model checker specializing in undergraduate operating systems class projects. Landslide extends the traditional model checking algorithm with a new framework for automatically managing multiple state spaces according to their estimated completion times, which I show quickly finds bugs should they exist and also quickly verifies correctness otherwise. I evaluate Landslide's suitability for inexpert use by presenting the results of many semesters providing it to students in 15-410, CMU's Operating System Design and Implementation class, and more recently, students in similar classes at the University of Chicago and Penn State University. Finally, I extend Landslide with a new concurrency model for hardware transactional memory, and evaluate several real-world transactional benchmarks to show that stateless model checking can keep up with the developing concurrency demands of real-world programs.

**Thesis Committee:
Garth Gibson, Chair
David Eckhardt
Brandon Lucia**

Haryadi Gunawi, University of Chicago