Empirically Investigating Strategies and Tools for Debugging Multithreaded Programs

Scott D. Fleming

Date: Feb.27th, 2009

Place: 256, Coates Hall

Time: 2:00 PM

ABSTRACT

Concurrency can provide significant performance benefits to software systems. Unfortunately, these benefits often come with an increase in software complexity that makes concurrent systems notoriously difficult to develop and maintain. Debugging concurrent software is particularly difficult. The presence of multiple control and data flows confounds program comprehension. Errors may be difficult to reproduce. The problem is compounded by a lack tools to address the challenges of concurrency. Furthermore, the literature offers little guidance on what strategies lead to success on such debugging tasks. In this talk, I present my recent empirical work that aims to catalog and evaluate strategies for debugging concurrent software. I will report on the contributions of this work, which include the identification and evaluation of several debugging strategies. Of particular interest is the discovery of a previously-undocumented strategy that leads to success on debugging tasks. I will also present empirically-founded recommendations for the design of debugging tools, and new empirical methods and metrics.

BIOGRAPHY

Scott Fleming is currently completing his PhD in Computer Science at Michigan State University. His research is in the area of software engineering with an emphasis on helping developers create and evolve concurrent software efficiently and without introducing defects. From 2001 to 2003, he worked as an instructor in the Department of Computer Science at Western Michigan University. He received the MS in Computer Science and the BA in Fine Art from Western Michigan University in 2001 and 1999, respectively.