Program Debugging: Research & Practice

Panel Moderator

W. Eric Wong
Department of Computer Science
University of Texas at Dallas
ewong@utdallas.edu

Regardless of the effort spent on developing a computer program, it may still have bugs. In fact, the larger and more complex a program, the higher the likelihood of it containing bugs. When the execution of a program on a test case fails, it reveals that there are bugs in the program. Then, the burden is on the programmers to locate and fix these bugs. However, program debugging can be extremely time-consuming and tedious, especially given the size and complexity of software we have today. Manual debugging is certainly not the right approach.

With this realization, researchers have proposed various techniques to assist programmers in finding and fixing bugs more effectively and efficiently. Yet, many questions still remain open and need to be further explored:

- How usefully can these techniques be transferred from laboratory settings to real-life industry environments?
- Are automated debugging techniques actually helping practitioners?
- Are the assumptions adopted by research-oriented debugging techniques valid in practice?
- Do we have solid data from rigorous case studies to prove the feasibility and advantage of using exiting research prototypes?
- What are the gaps between the best-of-breed practices in industry and the most advanced techniques proposed in academia?
- What are the limitations and challenges of current research in program debugging and what are the most urgent needs in practice?

Each panelist first gives a short presentation to report his or her experience in applying research methodologies and techniques to debugging large and complex real-life software systems and the challenges that they have had to overcome. The floor is then open for the audience to express their concerns and provide comments on the current research and practice of program debugging. A discussion for possible solutions is also conducted.