

Keynote 1

Self-Replay Augmented Record and Replay for Android

Zijiang Yang Ph. D

Founder of GuardStrike Inc.

Professor, Western Michigan University, USA

yang@guardstrike.com

Abstract

Record-and-replay tools are indispensable for quality assurance of mobile applications. In this talk, we present a record-and-replay tool targeting a wide adoption. Specifically, a dynamic instrumentation technique is used to accommodate rich sources of inputs in the application layer satisfying various constraints requested from industry. A self-replay mechanism is proposed to record more information of user inputs for accurate replaying without degrading user experience. In addition, an adaptive replay method is designed to enable replaying events on different devices with diverse screen sizes and OS versions. Through an evaluation on 53 highly popular industrial Android applications and 265 common usage scenarios, we demonstrate the effectiveness in recording and replaying rich sources of inputs on the same or different devices.

Biography

Zijiang Yang received his PhD from the Univ. of Pennsylvania, M.S. from Rice Univ. and B.S. from the University of Science and Technology of China, all in computer science. He is the founder of GuardStrike Inc and professor at Xi'an Jiaotong University. Before founding GuardStrike he was a professor at Western Michigan University. He is the chair of IEEE Technical Committee on Electric and Autonomous Driving, general chair of the 2019 IEEE International Conference on Software Testing, Verification and Validation, program committee chair of the 2018 International Conference on Reliable Software Engineering. Dr. Yang

received ACM SIGSOFT outstanding paper award, ACM TODAES best Journal Paper Award.

As of February 2021 he published 98 papers and a co-inventor of 10 US patents.

Keynote 2

Heterogeneous Computing: Tackling Various Computational Operations with Differential Software and Hardware

Hai Jiang, Ph.D.

Professor, Department of Computer Science

Arkansas State University, USA

hjiang@astate.edu

Abstract

In modern computers, all computational operations are specified in programs and executed in corresponding pipelines in unified Central Processing Units (CPUs). To improve system efficiency and speed up application execution, heterogeneous computing decentralizes the processing process and utilizes different software and hardware components on various computational operations. Other than CPU, many autonomous co-processors, accelerators, device drives, and special circuits have been developed to address domain specific tasks. In recent years, DSPs (Digital Signal Processors), GPUs (Graphics Processing Units), APUs (Accelerated Processing Units), TPUs (Tensor Processing Units), FPGAs (Field-Programmable Gate Array) and ASICs (Application-Specific Integrated Circuits) have demonstrated their effectiveness in domain specific processing. Both data and computations are arranged to achieve “Near Resource Computing” for efficient and accelerated application execution. This talk focuses on the architectural and programming trends in Heterogeneous Computing.

Biography

Dr. Hai Jiang is a professor in the Department of Computer Science at Arkansas State University, USA. He received his Ph.D. degrees from Wayne State University, USA. His research interests include Parallel & Distributed Systems, Cloud Computing, Big Data, High Performance Computing & Communication, Cryptography, and Computer & Network

Security. Dr. Jiang is a professional member of ACM and IEEE computer society. He has published four books and more than 100 papers in refereed journals, conference proceedings and multiple book chapters. He has been involved in more than 100 conferences and workshops as a program/workshop chair or as a program committee member. He serves as an editor or guest editor for multiple international journals such as IEEE Transactions on Network Science and Engineering. He has been chairing IEEE Technical Committee on Scalable Computing (TCSC) PhD Dissertation Award Selection Committee since 2017.