Abstract:
Modern computer software systems are prone to various classes of elusive faults due to their reliance on features such as multi-core processors and peripheral devices such as sensors. Testing remains a common method for uncovering faults in these systems. However, commonly used testing techniques that execute the program with test inputs and inspect program outputs to detect failures are often ineffective on these systems. In addition, existing testing techniques focus primarily on single applications, neglecting elusive faults occur at the whole-system level due to complex system events interactions.

In this talk, I present an approach that allows engineers to effectively test for elusive faults in modern software systems by providing them with deep observability and controllability. I propose a testing framework that first employs dynamic analysis techniques to observe system execution and identify program locations of interest. Next, the framework employs virtualization to achieve fine-grained controllability, allowing it to exercise event inter-leavings that are likely to expose faults. The ultimate benefit of this approach is its ability to test entire systems for broad classes of elusive faults. I illustrate the generality of the framework by discussing two of its instantiations: detecting process-level races and estimating worst-case interrupt latencies. I conclude by sharing my vision for new research directions related to the approach and to other aspects of software reliability and software security.

Bio:
Tingting Yu is a PhD candidate at the University of Nebraska-Lincoln, working with the software engineering research group. Her research interests span software engineering and software systems, with emphasis on the application of program analysis, monitoring, and virtualization techniques to problems in testing and maintenance of complex software systems, and empirical studies. Her work has been published in several highly selective software engineering venues.