IntroPerf: Transparent Context-Sensitive Multi-layer Performance Inference using System Stack Traces
Chung Hwan Kim, Junghwan Rhee*, Hui Zhang*, Nipun Arora*, Guofei Jiang*, Xiangyu Zhang, Dongyan Xu Purdue University and CERIAS, NEC Laboratories America*

Motivation
- Performance bugs are frequently observed in commodity software.
- Performance bugs may escape the development stage, and incur problems in a post-development setting.
- Commodity software consists of many interdependent components across multiple system layers.
- Software is often deployed in a binary format which lacks source level semantics.

Approach
- Transparent performance diagnosis with low overhead in the post-development stage.
- All components in the vertical software layers are analyzed with a system-wide scope.
- OS tracers are commonly used in modern operating systems for troubleshooting and advanced OS tracers provide system-wide stack traces.
- IntroPerf infers context-sensitive application performance and analyzes performance bugs by leveraging stack traces from OS tracers.

IntroPerf Architecture
- IntroPerf converts system stack traces to a set of function latencies.
- Performance bug candidate functions are ranked regarding dynamic calling contexts.

Visualization of Hot Call Paths

Coverage of Program States
- The experiments with Apache, MySQL, 7zip show that stack traces generally cover 5.3–49.4% of dynamic calling contexts and 0.6–31.2% of function instances.
- However, the coverage of calling contexts and instances for top 1% slowest functions are respectively 34.7~100% and 16.6~100% depending on applications.
- IntroPerf focuses on the functions with large latencies for performance diagnosis.