Comprehensive Multi-Platform Dynamic Program Analysis for the Java and Dalvik Virtual Machines

Walter Binder, Yudi Zheng, Lubomír Bulej, Haiyang Sun

Università della Svizzera italiana (USI)
Faculty of Informatics
Switzerland
firstname.lastname@usi.ch

Petr Tůma

Charles University
Czech Republic
petr.tuma@d3s.mff.cuni.cz

Abstract: Despite its importance for many software engineering tasks, dynamic program analysis is only insufficiently supported on the Java platform [KABM12]. Existing Java Virtual Machines (JVMs) as well as Android's Dalvik Virtual Machine (DVM) lack dedicated mechanisms for expressing arbitrary dynamic program analysis tasks at a high abstraction level, for ensuring complete code coverage of the analysis, and for isolating analysis tasks from the observed program to prevent interference. For example, the JVM Tool Interface requires analysis tasks to be written in low-level native code, and some virtual machines (e.g., DVM) do not support it. As a consequence, dynamic program analysis tools are often implemented using low-level mechanisms, resulting in error-prone code that is difficult to maintain, and support only a particular virtual machine. Moreover, many analysis tools produce unsound profiles (due to interference of the analysis with the observed program) or incomplete profiles (due to limited code coverage).

We present a novel dynamic program analysis framework that offers high-level abstractions for comprehensive, multi-platform analysis for the JVM and DVM. Our framework ensures complete bytecode coverage and isolates the execution of the analysis code from the observed program. It is based on the concepts developed for DiSL [MVZ+12, MZA+15, SZA+14], ShadowVM [MKZ+13, SZB+15], and FRANC [AKZ+13]. The domain-specific aspect language DiSL is used to specify the events of interest for an analysis. The events captured during program execution are transmitted to the ShadowVM, where the user-defined analysis code processes the events. Different event-ordering semantics are supported, avoiding synchronization for analyses that do not require global event ordering across all threads. In addition to events specified by DiSL code, our framework also generates lifecycle events and inter-process communication events. The latter are particularly important for the analysis of Android applications, as they typically involve multiple processes. Several state-of-the-art analysis tools have already been ported to our framework, including code coverage testing tools, calling-context profilers, and object liftetime profilers.

This work was also presented as an invited talk at PPPJ'14 [ZSB+14].

Acknowledgments

The research presented here has been supported by the Swiss National Science Foundation (project CRSII2_136225), by a Sino-Swiss Science and Technology Cooperation (SSSTC) Institutional Partnership (project IP04–092010), and by the European Commission (contract ACP2-GA-2013-605442).

References

[AKZ+13] Danilo Ansaloni, Stephen Kell, Yudi Zheng, Lubomír Bulej, Walter Binder, and Petr Tůma. Enabling Modularity and Re-use in Dynamic Program Analysis Tools for the Java Virtual Machine. In *ECOOP '13: Proceedings of the 27th European Conference on Object-Oriented Programming*, volume 7920 of *LNCS*, pages 352–377. Springer-Verlag, 2013.

[KABM12] Stephen Kell, Danilo Ansaloni, Walter Binder, and Lukáš Marek. The JVM is Not Observable Enough (and What to Do About It). In *VMIL '12: Proceedings of the 6th ACM Workshop on Virtual Machines and Intermediate Languages*, pages 33–38. ACM, 2012.

[MKZ+13] Lukáš Marek, Stephen Kell, Yudi Zheng, Lubomír Bulej, Walter Binder, Petr Tůma, Danilo Ansaloni, Aibek Sarimbekov, and Andreas Sewe. ShadowVM: Robust and Comprehensive Dynamic Program Analysis for the Java Platform. In *GPCE '13: Proceedings of the 12th International Conference on Generative Programming: Concepts and Experiences*, pages 105–114. ACM, 2013.

[MVZ+12] Lukáš Marek, Alex Villazón, Yudi Zheng, Danilo Ansaloni, Walter Binder, and Zhengwei Qi. DiSL: A Domain-specific Language for Bytecode Instrumentation. In *AOSD '12: Proceedings of the 11th Annual International Conference on Aspect-oriented Software Development*, pages 239–250. ACM, 2012.

[MZA+15] Lukáš Marek, Yudi Zheng, Danilo Ansaloni, Lubomír Bulej, Aibek Sarimbekov, Walter Binder, and Petr Tůma. Introduction to Dynamic Program Analysis with DiSL. *Science of Computer Programming*, 98, part 1:100–115, 2015.

[SZA+14] Aibek Sarimbekov, Yudi Zheng, Danilo Ansaloni, Lubomír Bulej, Lukáš Marek, Walter Binder, Petr Tůma, and Zhengwei Qi. Dynamic program analysis: Reconciling developer productivity and tool performance. *Science of Computer Programming*, 95, part 3:344–358, 2014.

[SZB+15] Haiyang Sun, Yudi Zheng, Lubomír Bulej, Alex Villazón, Zhengwei Qi, Petr Tůma, and Walter Binder. A Programming Model and Framework for Comprehensive Dynamic Analysis on Android. In *MODULARITY '15: Proceedings of the 14th International Conference on Modularity*. ACM, 2015.

[ZSB+14] Yudi Zheng, Haiyang Sun, Lubomír Bulej, Petr Tůma, and Walter Binder. Comprehensive Multi-platform Dynamic Program Analysis for the Java and Dalvik Virtual Machines. In *PPPJ '14: Proceedings of the 2014 International Conference on Principles and Practices of Programming on the Java Platform: Virtual Machines, Languages, and Tools*, pages 4–4. ACM, 2014.