A Whole Layer Performance Analysis Method for Android Platforms

Namseung Lee
LG Electronics,
Seoul, Korea
mainly82@gmail.com

Sung-Soo Lim
School of Computer Science,
Kookmin University,
Seoul , Korea
sslim@kookmin.ac.kr

As the products based on Android platform have been widely spread in consumer electronics market, the needs for systematic performance analysis have significantly increased. Conventional approaches rely on publicly open performance analysis tools in Android SDK or Linux community such as *DDMS (Dalvik Debug Monitor Server), LTTng, Oprofile*, and *Ftrace*. Though the approaches provide analysis or measurement results in certain aspects and specific software layers, any methods do not give a whole software layer view in performance analysis. For example, once a method in an Android application turned out to be a performance bottleneck, it is very hard to locate the code fragments that actually caused the bottleneck in the whole software layers: the application codes do not provide direct reason for the bottleneck, but the underlying native layers including kernel events often cause the bottleneck.

The software layers constituting the Android platform including Android application framework based on Dalvik VM, native system calls processing, and kernel events handling have separate publicly available performance analysis tools or methods. The whole layer performance analysis needs integrating the separate methods and making information exchange channels among the layers. The offline analysis could combine the results extracted from each software layer analysis.

We implemented a whole layer performance analysis tool set combining and integrating available open source performance analysis tools. Android framework layer profiling is performed *DDMS* mechanism with a little modification and the kernel layer profiling is based on *Ftrace*. The reason why we selected *Ftrace* instead of *Oprofile* as kernel layer profiler is that the *Oprofile* often misses the important kernel events to analyze since *Oprofile* is using purely time-based sampling. While *Ftrace* provides accuracy in performance profiling, the additional costs to perform kernel events profiling are significant. Therefore, minimizing the profiling cost maintaining the accuracy of event logging has been a primary issue in performance profiling at kernel layer.

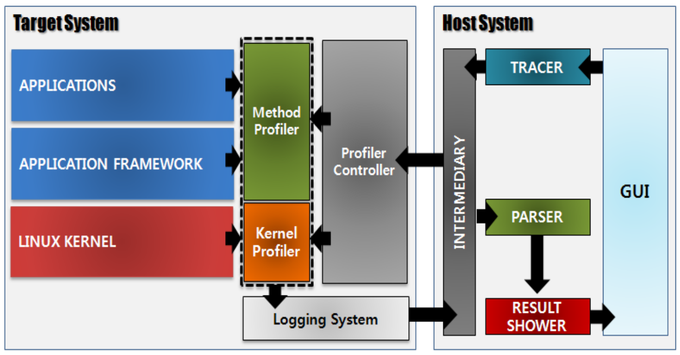

Figure 1 The architecture of the proposed whole layer performance analysis tool set

Figure 1 shows the architecture of the proposed tool set. One feature of the proposed tools is that the target Android applications do not need to be modified to enable the analysis. Through specifying the target method names to be analyzed at host system tool, the methods to be analyzed are identified and profiled automatically by profiler-enabled Android framework.

Users can specify the target methods to analyze from GUI of the host tool and the target method information is transferred to target system through Intermediary interface between the GUI and Profiler controller. Profiler controller actually enables logging systems in both application and kernel layers. Method profiler performs method profiling at Android framework level while Kernel profiler performs kernel event logging. The logged event data are collected together in Logging system to be transferred to hosts system for offline analysis. The analysis results are displayed using diverse presentation methods by GUI system at host.

The current version of the tool set supports Froyo (Android version 2.2) and Gingerbread (Android version 2.3) platforms and has been applied to a number of commercialized products including Nexus One smartphone to optimize the performance of user interfaces. The View system and touch screen user interface response performance have been significantly improved in target products based on the performance analysis results through our tool set.

This research was supported in part by the MKE, Korea, under the ITRC support program supervised by the NIPA (NIPA-2011-C1090-1131-0009) and in part by Basic Science Research Program through the NRF funded by the Ministry of Education, Science and Technology (2011-0004852).