

A Performance Monitoring Framework for Android applications

Sharvani B N¹ Rashmi R²

¹M.Tech. Scholar ²Assistant Professor

^{1,2}Department of Information Technology

^{1,2}RVCE Bangalore, India

Abstract— The Android application performance monitoring framework is a shared library project which should serve the purpose of gathering performance related data from the application in the runtime and analyze the bottlenecks in performance to optimize the application source code. It also aims at integrating a number of performance analysis tools for the same. The main issues targeted are memory load, CPU utilization, battery usage and data usage monitoring. The end product would be a shared library that can be integrated into the android application to identify potential risks at early stages of development. The whole purpose completely revolves around facilitating an android application to run smoothly without any disruptions. The purpose is to build a performance monitoring framework to integrate performance tools, collection of logs and log analysis for benchmarking performance of android applications.

Key words: Android application performance, Resource constraints, Memory monitor, MAT (Memory Analysis Tool), Performance evaluation

I. INTRODUCTION

In the recent days mobile devices have gained so much popularity all because of the mobile apps which are a driving force behind it. Mobile applications play a role in extending the functionality of the mobile devices. But the main drawback of mobile devices is the resource constraints [1]. The mobile applications which is built to operate on these platforms have to undergo a laborious review process to manage the resources in an appropriate way. Hence there is always a need to design and implement a performance monitoring framework [1] [2] [4]. The framework is light weight it does not necessitate any participation from the developer, to the operating system. We use performance monitoring framework to instrument android applications, and carry out a number of field trials with several applications for over a period of time to understand the behavior of the application. Report the characteristics of the critical paths that the framework would find in the data it generates. It gives a practical example of how performance monitoring framework helps developers improve the quality of their app [3] [4].

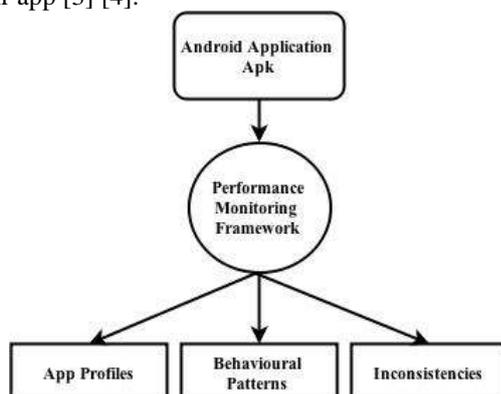


Fig. 1: Overview of the application performance monitoring framework

II. BACKGROUND RESEARCH

Managing mobile user experience and application performance is also an extremely unique and difficult problem, unlike managing application performance on servers. It can be best described as a “full-stack” problem which involves the mobile network, the mobile device, the mobile operating system, the mobile app itself, and everything else that is going on in the device at a point in time. For the above reasons, mobile application performance is a unique problem that must be addressed by vendors focusing exclusively upon this space [4] [5].

Mobile applications development is guided by the careful attention to a number of non-functional requirements, where a good user experience is the ultimate goal. As part of this effort, the developer needs to efficiently use the physical resources of the mobile device, such as CPU time, memory, battery, network etc. However, up until now, there is no structured method for performance evaluation in this domain that provides actionable information to the software developer. Some case studies regarding the performance evaluation process for mobile applications and its impact on development are evident.

With the advent of smart phones, mobile computing has been revolutionized and many new opportunities have been introduced into the world of research. A challenge with today's smart phones is their security issue with regards to third party applications. The way a given malicious application uses the phone resources like CPU, battery, network and private data is different from the others. The Android market is getting to a considerably large share and screening that for malicious applications becoming so hard, if not impossible. The aim is to provide an application profiler for Android platform. This can be achieved by implementing a profiling daemon which gathers the formation related to the usage of system resources like computing, memory, file manipulation, network and other sensing resources/information related to each of these mobile applications [6].

III. METHODOLOGY

The performance monitoring framework is a library project which should serve this purpose of gathering performance related data from the application and analyze the bottlenecks in performance to optimize the application source code. It also aims at integrating a number of performance analysis tools for the same. The main issues targeted are memory load, CPU utilization, battery usage, data usage monitoring and sync monitoring. All these are implemented as distinct services and are tested before integrating it into the library [4].

The end product would be a shared library that can be integrated into the test environment to identify potential risks at early stages of development. The whole purpose of the project completely revolves around facilitating the android app to run smoothly without any disruptions. The idea is to build a performance monitoring framework to integrate performance tools, collection of logs and log analysis for benchmarking performance of the application [5] [6].

Open source platforms like android vary according to device manufacturers and also application developers. There are multiple layers of software in which modifications have to be done including application, libraries, operating system, framework etc. A good user experience is the ultimate goal of any application development. The physical resources underlying the mobile applications have to be used efficiently. As part of this effort, the developer needs to capably use the physical resources of the mobile device, such as CPU time, battery usage, memory, data usage and sync activities. These are the key areas which are addressed in the performance monitoring framework.

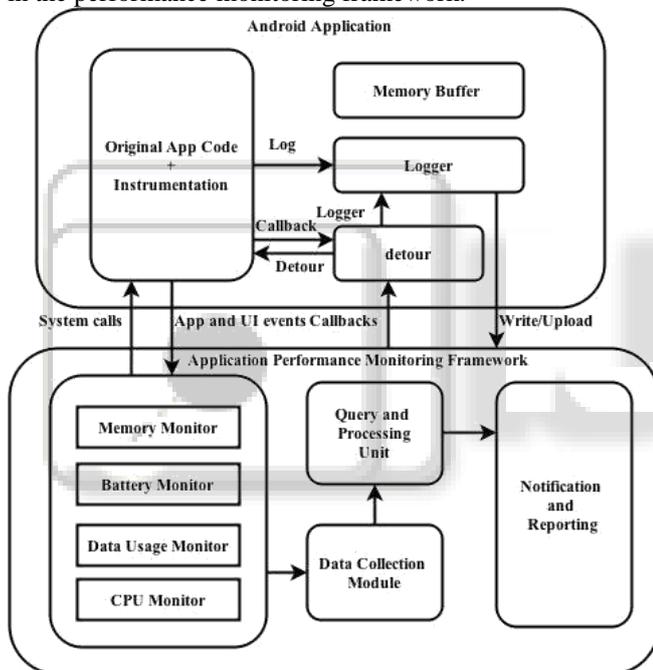


Fig. 2: Methodology of android application performance monitoring framework

IV. IMPLEMENTATION

The implementation of the android performance monitoring library involves the following techniques.

Performance Metrics depicts the ability to measure & report: Response times, Network Statistics, Resource (CPU, Disk, Memory) Usage Statistics, status code, warnings, errors and power consumption. Performance Profiling is the ability to provide automated recommendation to improve performance, method level timing, line level timing and decompiling third party code. Simulation and Reporting is the ability to emulate different types and speed of network, save profiling data for future viewing and generate easy to use reports. Other techniques include Advance Filtering, capability to interface with IDE's and user intuitive. In addition the implementation also includes implementing a number of android apis', adb shell

commands and also making use of some of the standard tools like Systrace, adb, MAT, DDMS from within the performance monitoring framework to get a big picture about the application resource usage statistics [9] [10].

The proposed framework checks adherence to performance best practices and ensures that the application meets the stated performance goals. It identifies and resolves performance bottlenecks and generates actionable feedback to improve performance and implement Performance optimization techniques [11].

V. CONCLUSION

Managing mobile applications to meet performance standards is a difficult issue as mobile devices pose a set of resource constraints such as low memory, power limitations, less powered CPUs. Hence a different perspective of application development is growing where in there a need for performance monitoring for these android applications. Performance is always a concern. The proposed work implements strategies that can improve the performance of the application. The project contains methods to judge memory leakage in an android application in the real time scenario. The objects causing memory leakage from the application are found and hence defects are tracked using log analysis. The power usage, CPU utilization, Data usage and the sync operations are monitored and results of the framework are compared across different tools like DDMS, MAT, Systrace, and Dumpsys etc. By extracting performance data the framework analyses heap memory situation and also memory leakage.

Performance evaluations shall be performed in a systematic way, enabling its reproduction, analysis, and use. This framework presents a methodical performance evaluation of a representative Android application and uses this analysis to evaluate the actual design space of the mobile application. In this study we observed that better use of assessment to eliminate possible combinations of factors that do not significantly alter the behavior of the application can make the evaluation more efficient. Also the difficulties to perform a reliable and meaningful assessment are pointed. This project shows that performance assessment in mobile domain is still an ad-hoc and time consuming task. From the derived results, it was possible to observe the influence of code libraries in application behavior and difficulty for the developer to improve the efficiency of an android application.

ACKNOWLEDGEMENT

I would like to thank Dr. Jitendranath Mungara for his valuable suggestions and expert advice in the process of preparing this paper.

REFERENCES

- [1] Kwon Y, Lee S, Yi H, Kwon D, Yang S, Chun B, Huang L, Maniatis P, Naik M, "Mantis: Efficient predictions of Execution Time, Energy Usage, memory Usage and Network Usage on Smart Mobile Devices", Mobile Computing, IEEE Transactions on (Volume:PP, Issue 99)
- [2] Hyojong Kim, Hongyeol Lim, Manatunga D, Hyesoon Kim, Gi-Ho Park, "Accelerating Application Start-up

- with Nonvolatile Memory in Android Systems”, Micro, IEEE (Volume:35 , Issue: 1), 12 January 2015.
- [3] Bernd Harzog, ”Managing Mobile Application Performance”, April 2014, The Virtualization Practice.
- [4] Myeongjin Cho, Seok Joong Hwang, Ho Jin Lee, Minseong Kim, Seon Wook Kim, “AndroScope for detailed performance study of the android platform and its applications”, Consumer Electronics (ICCE), 2012 IEEE International Conference.
- [5] Shengqian Yang, Dacong Yan, Rountev, A, ” Testing for poor responsiveness in android applications”, Engineering of Mobile-Enabled Systems (MOBS), 2013 1st International Workshop.
- [6] Zhou, Hui, ” Measuring available bandwidth for smart cyber-physical applications”, Tsinghua Science and Technology (Volume: 16, Issue: 6), IEEE 2012.
- [7] Bernd Herzog, ”Managing Mobile Application Performance”, April 2014, The Virtualization Practice.
- [8] Jana, S, Univ. of Texas at Austin, Austin, TX, USA and Shmatikov, V, ” Memento: Learning Secrets from Process Footprints”, Security and Privacy (SP), 2012 IEEE Symposium.
- [9] Kayande D, Shrawankar, U, ” Performance analysis for improved RAM utilization for Android applications”, Software Engineering (CONSEG), 2012 CSI Sixth International Conference.
- [10] amiyama, T, Inamura, H, Ohta, ” A model-based energy profiler using online logging for Android applications”, Mobile Computing and Ubiquitous Networking (ICMU), 2014 Seventh International Conference.
- [11] Shuai Hao, Ding Li, Willam G.J and Ramesh Govindan, ” Estimating android Applications’ CPU Energy Usage via Bytecode Profiling”, GREENS’12 Proceedings of the First International Workshop on Green and Sustainable Software.