Investigation of CPU Resource Consumption in Android

Kota KUMAKURA[†a)], ***Non-Member***, **Takeshi KAMIYAMA**[††], **Masato OGUCHI**[†††], ***and*** **Saneyasu YAMAGUCHI**[†], ***Member***

1. Introduction

Power saving is one of the essential themes in smartphones. There is a trade-off between CPU processing performance and CPU power consumption. It is desirable to control the CPU clock rate to increase when necessary and decrease when unnecessary. In smartphones, the foreground applications' CPU resource consumption dominantly determines the entire system's CPU resource consumption. In addition, the CPU utilization of an application strongly depends on the running method. In the work of [1], we proposed a concept for controlling CPU clock rate based on the invoked method in the foreground application and a method for investigating the CPU utilization of each method invocation. In this paper, we verify the CPU utilization investing method and discuss its accuracy and stability.

2. Related Work

In the work of [1], we proposed to control the CPU clock rate based on the invoked method in the foreground application. In the proposed system, if a CPU resource-consuming method is invoked, the system increases the clock rate. To investigate the relationship between invoked methods and CPU resource consumption, we proposed monitoring method invocations in ART (Android Runtime). In the work, we evaluated the proposed investing method with practical Android applications as an application benchmark and black box verification. However, the investigating system was not studied in the microbenchmark.

3. Evaluation

We implemented a simple test program for microbenchmarks. It has two methods. One is a CPU-intensive method (methodA). It iterates a loop that contains floating-point operations. The other is a non-CPU intensive method (methodB). It contains only the thread sleeping method. The CPU usages of these methods are quite different while the execution times of these are not largely different. We invoked these two methods with the default CPU governor, the fixed minimum CPU clock rate, and the fixed maximum CPU clock rate. Fig. 1 shows the results. The figure shows that the method could detect the large CPU usage difference. The measured method execution times of the non-CPU intensive method were stable. This is mainly because the times were exactly determined by the sleep method. The measured CPU usages were not relatively stable but were stable because these values were very small. Focusing on the CPU-intensive method, we can see that the CPU usages were stable although their execution times were not stable. Fortunately, the CPU usages of the CPU-intensive methods were important and were measured accurately. Thus, we can expect that the CPU clock rate control proposed in [1] will work effectively.

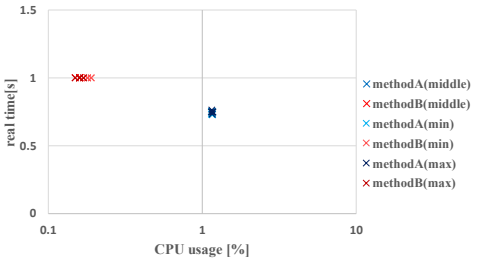

Fig. 1 CPU usage by each methods

4. Conclusion

In this paper, we introduced an investigating method for measuring the CPU usage of each method invocation and discussed its accuracy and stability of measurements.

Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers 21K11854 and 21K11874.

References

[1] K. Kumakura, A. Sonoyama, T. Kamiyama, M. Oguchi and S. Yamaguchi, "Observation of Method Invocation in Application Runtime in Android for CPU Clock Rate Adjustment," 2021 Ninth International Symposium on Computing and Networking Workshops (CANDARW), 2021, pp. 481-483, doi: 10.1109/CANDARW53999.2021.00090.

[†]The author is with Kogakuin University
[††]The author is with Nagasaki University
[†††]The author is with Ochanomizu University
[a)] E-mail: cm22022@ns.kogakuin.ac.jp