Work-in-Progress: SmartDTM: Smart Thermal Management for Smartphones

Wook Song*

Software R&D Center
Samsung Electronics
wook16.song@samsung.com

Jihong Kim

Department of Computer Science and Engineering
Seoul National University
jihong@davinci.snu.ac.kr

ABSTRACT

We propose a novel DTM scheme based on the user-perceived response time analysis, called SmartDTM. Unlike existing DTM schemes that can significantly degrade the quality of user experience, SmartDTM takes explicit account of the quality of user experience into making the DTM decisions. Our experimental results on an ODROID-XU+E board show that the proposed technique can improve the user-perceived performance by up to 37% over the Android's default DTM policy without any thermal violations.

CCS CONCEPTS

• **Computer systems organization → Embedded systems**;

KEYWORDS

Smartphone, operating system, thermal management

ACM Reference format:
Wook Song* and Jihong Kim. 2017. Work-in-Progress: SmartDTM: Smart Thermal Management for Smartphones. In *Proceedings of EMSOFT'17 Companion, Seoul, Republic of Korea, October 15–20, 2017,* 2 pages.
DOI: 10.1145/3125503.3125567

1 INTRODUCTION

Modern smartphones have multi-core processors which run at more than 2 GHz. High power densities produce excessive chip temperatures which waste energy and reduce reliability. Thermal management is therefore a crucial design requirement.

As a software-based approach, the dynamic thermal management (DTM) [1] is commonly used in smartphones. A DTM scheme aims to have the CPU temperature below a critical temperature above which the processor chip could be damaged. When the current temperature reaches a predefined trigger temperature, the maximum operating frequency of the processor is reduced. When the CPU temperature drops below the trigger temperature, the maximum operating frequency of the processor is gradually increased to restore system performance.

Although the DTM scheme can effectively mitigate the thermal problems, it makes CPU frequency scaling decisions based on the current temperature only without considering the user's current computing requirement. Therefore, under existing DTM techniques, the quality of the user experience can be significantly degraded if a DTM decision to lower the CPU frequency is made when a high computing requirement should be met for a great user experience. Considering that most user-interactive sessions can be divided into two parts, one where the system performance level directly affects the quality of the user experience (called the display-sensitive part) and the other where the system performance level does not affect the quality of the user experience (called the display-insensitive part) [2], it is possible to make DTM techniques smarter if we can distinguish the display-sensitive part and display-insensitive part of a user-interactive session.

*A large part of this work was done while Wook Song was a Ph.D. student at SNU.

EMSOFT'17 Companion, Seoul, Republic of Korea

DOI: 10.1145/3125503.3125567

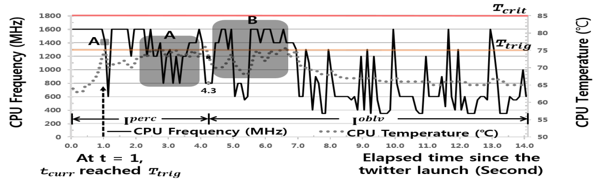

Figure 1: Changes in CPU frequency and temperature under the Android's default DTM policy.

In this paper, we propose such a novel DTM technique for smartphones, called *SmartDTM*, which improves the quality of user experience without violating the thermal requirement. The proposed SmartDTM technique is based on two key components, which form the main contributions of this paper, a user-perceived response-time predictor (urp) and a worst-case temperature predictor (wtp). At the start of each interactive session S, urp estimates I_S^{perc}, the length of the display-sensitive part of the session S, using a history of previous I_S^{perc} values of the display-sensitive part. Based on the estimated I_S^{perc} and the current temperature, wtp predicts the temperature T_S^{end} at the end of the display-sensitive part of S. In order to provide better user experience, even though the current temperature is higher than the trigger temperature, SmartDTM does not lower the maximum operating frequency if T_S^{end} does not exceed the critical temperature. On the other hand, SmartDTM employs an aggressive DVFS policy during the display-insensitive part. Since the system performance level in I_S^{oblv}, the display-insensitive part of the session S, less likely to affect the quality of user experience, the CPU temperature is quickly decreased to a safe level by aggressively scaling down the maximum operating frequency with no negative impact on user experience.

2 MOTIVATION

To explain the basic motivation of our proposed SmartDTM technique, we use an example interactive session S_L which launches the **twitter** app under the Android's default DTM policy with the initial temperature at 65 °C. In this example, we assume that the critical temperature T_{crit} and the trigger temperature T_{trig} were set to 85 °C and 75 °C, respectively. Fig. 1 shows how the CPU frequency and on-chip temperature change during 14.5 seconds after **twitter** is launched as measured in our evaluation board, ODROID-XU+E. The X-axis, the Y-axis on the left side, and the Y-axis on the right side represent the elapsed time, CPU frequency, and CPU temperature, respectively. At $t = 0$, the interactive session S_L is initiated by launching **twitter**. The user-visible interface is fully drawn at $t = 4.3$, and this is the user-perceived response time of S_L. However, since the CPU temperature reached T_{trig} after 1 second, the

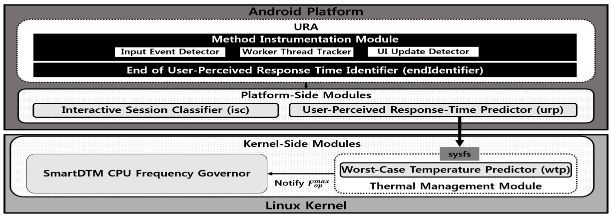

Figure 2: An architectural overview of SmartDTM.

maximum CPU operating frequency F_{op}^{max} was reduced from 1,600 MHz to 1,400 MHz. The temperature then drops below 75 °C, and after 1.8 seconds, the default DTM policy restores F_{op}^{max} to 1,600 MHz. Then the CPU temperature starts rising again, and reaches 75 °C after 2.4 seconds, when F_{op}^{max} is again reduced; but this time, the CPU temperature remains near T_{trig}, and so the default DTM policy keeps reducing F_{op}^{max} in several periods until it reaches 800 MHz. Since the big cluster only handles the performance requirement higher than the CPU frequency of 800 MHz, processing is switched from the big to the littler cluster at this point. Although the default DTM policy can successfully control the CPU temperature below T_{trig}, on average, its decisions are not effective in two aspects. First, as shown in two areas (marked as A) in Fig. 1, the user-perceived launching time was significantly increased because the default DTM policy lowered F_{op}^{max} too aggressively. Second, the CPU temperature tends to drop very quickly in the I^{oblv} interval. If the DTM policy had knew this thermal characteristic *a priori*, it could have avoided lowering F_{op}^{max} within the I^{perc} interval.

3 DESIGN AND IMPLEMENTATION

As shown in Fig. 2, the proposed SmartDTM consists of three main components, **ura** [2], platform-side and kernel-side modules. **ura** is responsible for the identification of the end of $I_{S_i}^{perc}$ during run time from the execution of S_i.

Isc and urp are the platform-side modules in SmartDTM. When a user interacts with the UI components, isc creates a unique identifier at the start point of the interactive session. Once the unique identifier is created, all the sessions that have an identical identifier are grouped. When endIdentifier has determined that $I_{S_i}^{perc}$ has ended, it informs urp, which adds the user-perceived response time to the total associated with the corresponding session identifier. To predict the user-perceived response time, urp uses a statistical analysis of the accumulated user-perceived response time information.

On the kernel side of SmartDTM, there are two modules, wtp and the SmartDTM CPU frequency governor. Wtp in the thermal management module, which is responsible for applying the DTM decisions, estimates the time at which the CPU temperature will reach T_{crit} by performing the worst-case temperature estimation whenever T_{curr} reaches T_{trig}. If T_{curr} will not exceed T_{crit} during the execution of $I_{S_i}^{perc}$, the module does not change F_{op}^{max} in order to improve the user-perceived performance. Otherwise, as with the default DTM policy, F_{op}^{max} is reduced by the thermal management module. When F_{op}^{max} is changed or endIdentifier detects the end of $I_{S_i}^{perc}$, the thermal management module notifies it to the SmartDTM CPU frequency governor. Furthermore, when $I_{S_i}^{oblv}$ starts, the SmartDTM governor employs the lowest CPU frequency, which rapidly reduces the CPU temperature while executing $I_{S_i}^{oblv}$.

4 EXPERIMENTAL RESULTS

We implemented SmartDTM on the Exynos 5410-based ODROID-XU+E board running Android 4.4.2 (Kitkat). We experimented with 7 apps under different usage scenarios. Each app usage scenario consists of two consecutive interactive sessions (an odd-numbered launching session followed by an even-numbered session initiated by a user input). For example, in the case of **band**, S1 represents the launching session of **band** while S2 represents the following session initiated by the user to view an article after **band** is launched.

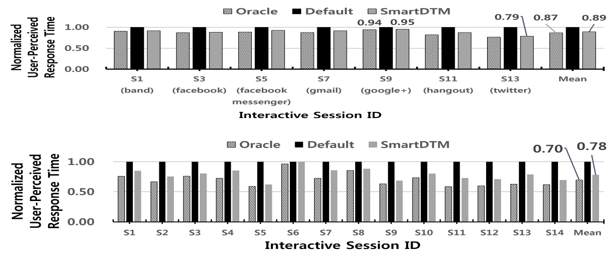

Figure 3: A comparison of normalized user-perceived response times for 7 launching interactive sessions when the initial temperature was 65 °C (top) and for 14 interactive sessions when the initial temperature was 70 °C (bottom).

Fig. 3 (top) shows the effect of SmartDTM on the user-perceived response times for 7 launching interactive sessions. In this experiment, at the start of each session, the CPU temperature was set to 65 °C. On these interactive sessions, SmartDTM decreases the average user-perceived response time by 11% over the default DTM policy. For S13 (**twitter**), the proposed SmartDTM achieves the maximum improvement of 21% in the user-perceived response time. In the worst-case, S9 (**google+**), SmartDTM decreases the user-perceived response time by 5%. Considering that the decrease in the user-perceived response time using Oracle was also only 6% for S9, SmartDTM performs close to Oracle when the initial temperature is set to 65 °C.

As the initial temperature increases, more reductions in CPU frequency are required to avoid the thermal violation during the execution of I_S^{perc}, which further reduces user-perceived performance. Fig. 3 (bottom) shows normalized user-perceived response times for the 14 interactive sessions when the initial temperature is 70 °C. SmartDTM decreases the average user-perceived response time by 22% while Oracle decreases the average user-perceived response time by 30%.

5 CONCLUSIONS

We have presented a novel DTM scheme for smartphones, called SmartDTM. Experimental results show that when the initial temperatures were set to 65 °C and 70 °C, respectively, SmartDTM can improve the user-perceived response time by 11% and 22%, on average, over the Android's default DTM policy under the critical temperature of 85 °C.

ACKNOWLEDGMENTS

This study was based on Core Component Technology Development for HMA-based System Optimization (July 1, 2012 - June 30, 2014) funded by Samsung Electronics Co., Ltd. This study was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2015M3C4A7065645). The ICT at Seoul National University and IDEC provided research facilities for this study. (Corresponding Author: Jihong Kim)

REFERENCES

[1] D. Brooks and M. Martonosi. 2001. Dynamic thermal management for high-performance microprocessors. In *Proceedings of the International Symposium on High-Performance Computer Architecture*. 171–182.

[2] W. Song, N. Sung, B.-G. Chun, and J. Kim. 2014. Reducing energy consumption of smartphones using user-perceived response time analysis. In *Proceedings of the Workshop on Mobile Computing Systems and Applications*. 20:1–20:6.