Thermal-Aware Slack Distribution for Real-Time Systems

Changha Lee, Young-Ho Lee and Jihong Kim
School of Computer Science and Engineering
Seoul National University, Korea
Email: {hadess, buriburi205, jihong}@davinci.snu.ac.kr

***Abstract*—We propose a thermal-aware slack distribution technique for real-time systems based on variable-voltage processors. Our proposed technique considers tasks steady-state temperature as a primary parameter in distributing available slacks to the tasks. By exploiting the thermal characteristics of real-time tasks, the proposed technique assigns more slacks to hot tasks than cold tasks, thus enabling more efficient peak temperature control. Experimental results show that the proposed technique can reduce the peak temperature by up to 7°C.**

I. Introduction

As the power density of modern microprocessors increases exponentially, temperature has become a major concern in designing high-performance systems. In order to safely control the operating temperature, diverse power/thermal management techniques have been proposed. Since most power/thermal management techniques throttle performance to lower the temperature, efficient thermal control should aim at mitigating inevitable performance loss. Especially in real-time systems, such performance loss could violate the deadlines of real-time tasks. Therefore, the thermal-aware real-time scheduling should not only minimize the peak temperature to ensure thermal safety but also meet the timing requirements. In this paper, we focus on the slack distribution policy for thermal-aware DVFS real-time scheduling.

II. Thermal-Aware Slack Distribution Technique

Our thermal-aware slack distribution technique consists of two steps. At the off-line step, the input tasks are categorized into 'hot' and 'cold' types. During the on-line step, whenever slack times are available, our slack distribution policy assigns *appropriate* slack time to the next activated task.

Given an input task set, a task classification policy determines whether the input tasks are classified into 'hot' or 'cold' types. We define $T = \{\tau_1, \tau_2, ..., \tau_n\}$ as the set of n periodic real-time tasks, where a task τ_i is characterized by $\{p_i, d_i, acet_i, wcet_i, st_i\}$. For classifying tasks, we use the threshold temperature T_θ, which is an average steady-state temperature of the input tasks (i.e., $\sum_{i=1}^{n} st_i$). A real-time task is categorized into 'hot' if its steady-state temperature st_i exceeds T_θ. Otherwise, the task is classified into 'cold'.

In a slack distribution policy, the goal is to determine appropriate slack time to the next activated task. The basic idea is to assign more slack time to the next activated hot task. However, simply allocating all slack times to the next hot task would not be the best policy because other following hot tasks may lose a chance to use some of the estimated slack time. To rectify this problem, we introduce the speed bound ratio of the next activated task τ_i, which bounds the actual operating frequency of τ_i into $[lb_i, ub_i]$. Note that hot tasks have lower speed bound ratios than cold tasks. Thus, more available slack time can be assigned to hot tasks, while hot tasks do not exhaust all the slack times due to their upper speed bound ratios. Once the range of the speed bound ratio is calculated, the range of available slack time for τ_i can also be determined as $[min(wcet_i/ub_i - wcet_i, es_i), min(wcet_i/lb_i - wcet_i, es_i)]$, where es_i is the estimated slack time. To choose the actual slack time of the next activated task τ_A, we consider the next activated task τ_B which will be executed right after τ_A. When τ_A is cold, τ_A runs at the maximum operating frequency regardless of the type of τ_B, because τ_A is thermally safe. If τ_A is hot, however, the actual slack time of τ_B depends on whether τ_B is hot or cold. If τ_B is cold, τ_A is assigned the maximum available slack time, since the temperature increase by τ_A will be quickly lowered during the execution of τ_B. If both τ_A and τ_B are hot, the actual slack time for τ_A is determined to reduce the peak temperature after the execution of τ_A and τ_B. For temperature estimation after the execution of τ_A and τ_B, we use a thermal model proposed in [2].

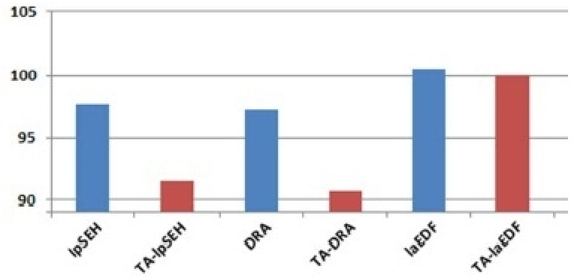

Fig. 1. Experimental results

III. Experimental Results

For evaluation, we choose the following energy-aware DVFS algorithms [1] based on the EDF scheduling policy: lpSEH, DRA, laEDF. Fig. 1 shows the experimental results. TA achieves the peak temperature reduction of lpSEH and DRA as much as 6°C (TA-lpSEH) and 7°C (TA-DRA), respectively. The reason is that both lpSEH and DRA employ aggressive slack estimation techniques which enable TA to apply more effective thermal control.

IV. Conclusion

We have proposed a thermal-aware slack distribution technique for real-time systems. Experimental results show that the proposed technique can reduce the peak temperature by up to 7°C.

Acknowledgment

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government(MEST) (No. 20100018873, and No. 2010-0020724). This work was also supported by the Brain Korea 21 Project in 2011. The ICT at Seoul National University and IDEC provided research facilities for this study.

References

[1] D. Shin, W. Kim, J. Jeon, J. Kim, and S. Lee, "SimDVS: An Integrated Simulation Environment for Performance Evaluation of Dynamic Voltage Scaling Algorithms," In Proc. of Workshop on PACS, 2002.
[2] Y. Wang, K. Ma, and X.Wang, "Temperature-Constrained Power Control for Chip Multiprocessors with Online Model Estimation," In Proc. of ISCA, 2009.