

# Thermal-Aware Design and Management of Embedded Real-Time Systems

Youngmoon Lee

Department of Robotics, Department of Smart City Engineering  
Hanyang University, Korea  
youngmoonlee@hanyang.ac.kr

**Abstract**—Modern embedded systems face challenges in managing on-chip temperature as they are increasingly realized in powerful system-on-chips. This paper presents thermal-aware design and management of embedded systems by tightly coupling two mechanisms, *thermal-aware utilization bound* and *real-time dynamic thermal management*. The former provides the processor utilization upper-bound to meet the chip temperature constraint that depends not only on the system configurations and workloads but also chip cooling capacity and environment. The latter adaptively optimizes rates of individual task executions subject to the thermal-aware utilization bound. Our experiments on an automotive controller demonstrate the thermal-aware utilization bound and improved system utilization by 18.2% compared with existing approaches.

## I. INTRODUCTION

Embedded systems are increasingly realized in state-of-the-art system-on-chips (SoCs) for real-time processing and control such as in autonomous vehicles. As a result, power densities in such devices are increasing at a significant rate, thus increasing the on-chip temperature to dangerously high levels. Increasing the on-chip temperature causes functional errors and adversely affects long-term reliability. Furthermore, increasing the leakage power dissipation may lead to thermal runaway [1], which is detrimental to safety-critical embedded systems. Modern processors rely on dynamic thermal management to manage the chip temperature by using hardware throttling such as dynamic voltage/frequency scaling (DVFS) and clock gating. However, such mechanisms delay time-critical task executions and cause unacceptable performance degradation for real-time applications where computational jobs must complete within their deadlines.

To remedy this performance degradation, proposals for thermal-aware real-time DVFS [2], [3] and task scheduling [4], [5] aim to optimally schedule task executions and voltage/frequency scaling over time. However, existing proposals for thermal-aware scheduling require an accurate processor thermal model at the time of system design while the chip temperature is greatly affected by runtime workload and surrounding environment. Given factors impacting the system thermal behavior, we consider a simple yet fundamental question: *can we determine an upper bound of processor utilization such that the thermal requirements are met if the processor utilization remains below this bound?*<sup>1</sup> Such a bound depends not only

<sup>1</sup>Similar to the *schedulable utilization bound*; real-time schedulers guarantee all the task deadlines if the processor utilization is less than the bound.

on the hardware configurations and application workloads but also ambient temperature and chip cooling capacity.

This paper presents a *thermal-aware utilization bound*, which is an upper bound on the utilization that guarantees the temperature constraint, and *real-time dynamic thermal management (RT-DTM)*, which is task-level thermal management to adaptively control heat generation by individual task executions via period adaptation. We first present a task-level power and thermal model to determine the thermal-aware utilization bound by experimentally characterizing heat generation of the individual tasks on the devices. For the task-level thermal model, we first identify the device switching activity factor during each individual task execution. We then develop RT-DTM that adaptively optimizes the task period subject to the thermal-aware utilization bound. Finally, we have implemented and evaluated RT-DTM on an automotive controller [6] using a suite of embedded benchmarks [7]. The proposed thermal model was validated and used to determine the thermal-aware utilization bound for different workloads, ambient temperatures, and chip thermal characteristics. RT-DTM provides an 18.2% improvement of processor utilization over an existing solution [8] while meeting the temperature constraint.

The main contributions of this paper include:

- Characterization of a thermal-aware utilization bound by studying the impact of different tasks, ambient temperature, and chip cooling capacity;
- Development of a RT-DTM for runtime thermal control;
- Demonstration of the thermal-aware utilization bound and proposed RT-DTM on an industrial controller.

## II. RELATED WORKS

Thermal management has received attention in recent years and architecture-level thermal simulators have been widely used to predict processor thermal behavior [9]. Thermal management in time-critical systems focuses on thermal-aware scheduling [4], [5], real-time DVFS [2], [3], and task thermal shaping [10]; however, these previous studies did not address the chip temperature constraint. Various studies [11] have considered the temperature constraint to guarantee thermal schedulability for periodic tasks. A thermal-aware server framework has been proposed to regulate the operating temperature for GPUs [12] and mixed-criticality systems [13]. To ease thermal analysis, these studies assume heat generation

only depends on the processor speed regardless of the workload thermal characteristics. To the best of our knowledge, little has been done to model the predictable thermal behavior of embedded real-time systems by capturing the dynamic and leakage power caused by task workloads.

### III. THERMAL-AWARE DESIGN

This section presents how we model and characterize thermal behavior, and derive the thermal-aware utilization bound. **System Thermal Model.** Embedded control systems run periodic workloads such as image processing, communication and control functions. In the standard periodic task model [11], each task  $\tau_i$  has an associated period,  $p_i$ , and an execution time,  $e_i$ . Therefore, the processor utilization of  $\tau_i$  is  $e_i/p_i$ . For a periodic task, task-specific dynamic power dissipation increases with the busy period or processor utilization of the task. For a given task set  $\{\tau_i\}_{i=1}^n$ , the dynamic power dissipation is characterized as follows:

$$P_{dyn} = \sum_{i=1}^n P_{dyn}(\tau_i) = \sum_{i=1}^n \frac{e_i}{p_i} \alpha_i V^2 f \quad (1)$$

where,  $V, f, e_i, p_i, \alpha_i$  are the operating voltage, frequency, execution time, period, and switching activity factor of task  $\tau_i$ , respectively. On the other hand, the leakage power increases with temperature, and we consider the leakage power model in [1]. The total power dissipation is the sum of dynamic and leakage power dissipation written as:

$$P_{tot} = P_{dyn} + P_{leak} = \sum_{i=1}^n \frac{e_i}{p_i} \alpha_i V^2 f + V(\beta_0 T_{chip}^2 e^{\frac{\beta_1}{T_{chip}}} + \beta_2). \quad (2)$$

where,  $\beta_0, \beta_1$ , and  $\beta_2$  are the platform-specific leakage parameters that can be identified from offline thermal characterization.

To translate this power dissipation to temperature, we consider the RC thermal circuit [9] as:

$$C_{th} \frac{dT_{chip}(t)}{dt} + \frac{T_{chip}(t) - T_a}{R_{th}} = P_{tot}, \quad (3)$$

where  $T_{chip}(t)$  denotes the chip temperature measured by an on-chip thermal sensor,  $T_a$  denotes the surrounding ambient temperature,  $R_{th}$  is the chip thermal resistance, and  $C_{th}$  is the chip thermal capacitance. In the steady-state, i.e.,  $\frac{dT_{chip}(t)}{dt} = 0$ , the temperature reaches the steady-state temperature determined by the total power dissipation written as:

$$T_{chip} = P_{tot} \cdot R_{th} + T_a. \quad (4)$$

**Thermal Characterization.** Using an on-chip thermal sensor available on commodity chips [6], we can experimentally characterize embedded platforms and workloads via *system identification*, i.e., identify task-specific switching activity ( $\alpha_i$ ) and platform-specific leakage parameters ( $\beta_0, \beta_1, \beta_2$ ) by measuring  $T_{chip}$ . We first characterized the leakage parameters by measuring the steady-state temperature under varied ambient temperature (20°C to 60°C). We placed the target platform in a thermal chamber to precisely control the ambient

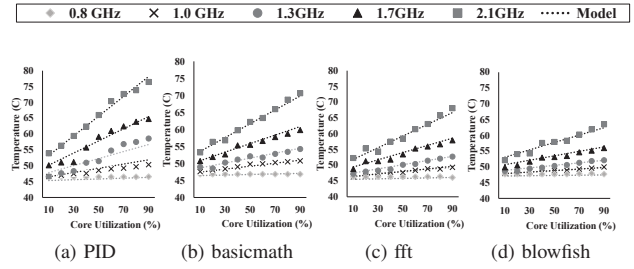


Fig. 1: Steady-state temperature from periodic tasks with varied configurations of utilization (10% to 90%) and DVFS (0.8GHz to 2.1 GHz)

temperature and made all cores *sleep* such that the dynamic power dissipation did not increase the chip temperature. With these steady-state temperature measurements, unknown leakage parameters can be identified using the standard regression method. We then characterized individual tasks by sweeping the operating frequency and utilization under constant room temperature. A periodic task was executed to measure the steady-state temperature for each discrete operating frequency level (0.8GHz to 2.1GHz). The experiment is repeated for each task individually, demonstrating the switching activity factor of individual tasks from different steady-state temperatures resulting from different voltage/frequency configuration. Sec. V provides more details on the experiment setups and thermal workloads.

Fig. 1 plots the measured steady-state temperature (points) from periodic task execution vs. varied utilizations and DVFS levels. The experiment of each benchmark application individually was repeated to characterize the switching activity factor of individual applications. Our thermal model well predicted the chip temperature measurement (dotted line) that characterized the switching activity factor. The tasks show significantly different temperature increases, showing that tasks generate different amounts of heat depending on the low-level switching activity of the device. For example, *PID* is a simple feedback control function with a CPU-intensive loop that results in the highest steady-state temperature of 76.5°C. On the other hand, *blowfish* has the lowest IPC resulting in the lowest temperature, 63.7°C. Thus, *PID* is characterized as the highest switching activity factor. Our thermal model achieves 99% R<sup>2</sup> fit in contrast to previous studies with 83% [8].

**Thermal-Aware Utilization Bound.** Using this thermal model experimentally-derived on real chips, we can derive the utilization upper-bound in a real-time system design that guarantees the threshold temperature. By plugging threshold temperature  $T_{max}$  into  $T_{chip}$  in Eq. (4), the processor utilization of the given task set must meet the chip temperature constraint as follows:

$$\sum_{i=1}^n \frac{e_i}{p_i} \alpha_i V^2 f + V(\beta_0 T_{max}^2 e^{\frac{\beta_1}{T_{max}}} + \beta_2) \leq \frac{T_{max} - T_a}{R_{th}} \quad (5)$$

where the left-hand side is the total power dissipation and the right-hand side is the available thermal budget ( $T_{max} - T_a$ )

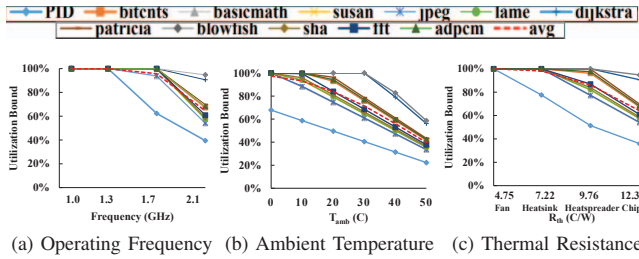


Fig. 2: Thermal-aware utilization for different operating frequencies, ambient temperatures and thermal resistances

divided by the thermal resistance ( $R_{th}$ ), i.e., the chip power budget. The processor utilization is bound by thermal budget and chip cooling capacity, depending on the voltage/frequency configurations and task workloads.

Fig. 2 illustrates this utilization upper-bound for different tasks and thermal design configurations, i.e., the utilization upper-bound for various tasks vs. different operating frequencies, ambient temperatures and chip thermal resistances. Fig. 2a shows that, with lower operating frequencies of 1.0 to 1.3GHz, the processor can be fully utilized across all applications. Dynamic power dissipation increases with operating voltage/frequency, i.e.,  $P_{dyn} \propto V^2 f$ , so processor utilization is rapidly bound by the temperature constraint. At maximum operating frequency, only 39% of the processor time can be used for the hottest task (*PID*), which is significantly lower than 95% for the coolest task (*blowfish*). On average, up to 69% of the processor utilization can be used across the benchmark applications on the target platform.

In the case of the maximum frequency operation, Fig. 2b plots the utilization bound for different ambient temperatures. From the default ambient temperature of 25°C, the thermal budget linearly decreases with an increase in the ambient temperature. In our thermal model, the increase/decrease of ambient temperature decreases/increases the thermal budget with the utilization bound. At an ambient temperature of 0°C, up to 97.5% of processor utilization can be used across applications on average, while only up to 40% at 50°C.

Fig. 2c illustrates the impact of the cooling capacity on the utilization bound. Fig. 2c plots the utilization bound for varied chip thermal resistance corresponding to different cooling devices, i.e., a fan, heat sink, heat spreader, and bare chip. According to our thermal model, an increase in thermal resistance decreases the utilization bound inverse proportionally, i.e.,  $\frac{e_n}{p_n} \propto \frac{1}{R_{th}}$ . A passive heat sink with a thermal resistance of 7.22 °C/W allows most of the tasks to fully utilize the processor except for the hottest task, which is bound to 78% of utilization. For the hottest task, the chip requires lower thermal resistance to be fully utilized such as a cooling fan rated as 4.75 °C/W, suggesting that the cooling system design must consider the thermal workload on the platform.

#### IV. THERMAL-AWARE MANAGEMENT

We now present a runtime management scheme called *Real-Time Dynamic Thermal Management* (RT-DTM), which adjusts task periods to meet the thermal-aware utilization bound.

Specifically, we present how to formulate the optimization problem, perform runtime thermal control, and implement RT-DTM.

**Problem Definition.** Given an  $m$ -core multi-core chip and operating frequency  $f \in [f_{min}, f_{max}]$ , a real-time task set  $\{\tau_i\} (i = 1, \dots, n)$ , and a temperature threshold  $T_{max}$ , we want to determine the task periods  $\{p_i\}$  that maximize the processor utilization as:

$$\text{maximize } \sum_{i=1}^n \frac{e_i}{p_i} \quad (6)$$

$$\text{subject to } \sum_{i=1}^n \frac{e_i}{p_i} \alpha_i V^2 f + V(\beta_0 T_{max}^2 e^{\frac{\beta_1}{T_{max}}} + \beta_2) \leq \frac{T_{max} - T_a}{R_{th}} \quad (7)$$

$$\sum_{i=1}^N \frac{e_i}{p_i} \leq m \frac{f}{f_{max}} \quad (8)$$

$$\forall \tau_n \quad p_i^{\min} \leq p_i \leq p_i^{\max}. \quad (9)$$

Eq. (6) defines the objective function maximizing the processor utilization for throughput maximization. Cost-conscious embedded systems are designed to maximize the processor utilization and achieve higher throughput with limited computing resources. Eq. (7) specifies the upper bounds of the chip temperature based on the thermal model. For real-time systems with periodic task workloads, the on-chip temperature remains close to the steady-state level that can be efficiently predicted. Eq. (8) specifies the  $m$ -core schedulable utilization bound to guarantee the deadlines of real-time tasks. When DVFS is applied, the system must meet this scaled schedulable utilization bound [14]. Eq. (9) defines the minimum and maximum bounds of an allowable task period within  $[p_n^{\min}, p_n^{\max}]$  determined by system design. Here,  $p_n^{\max}$  guarantees the minimum requirement of the task execution whereas  $p_n^{\min}$  is the optimal task period.

**Runtime Thermal Management.** Real-time embedded systems are often exposed to a wide range of external temperature because of surrounding components and environments. Thus, we need to adaptively control the chip temperature under the dynamically-changing ambient temperature. Our thermal model abstracts the surrounding temperature as the ambient temperature, which is the steady-state chip temperature reached in the ambient state. RT-DTM periodically measures the on-chip temperature to obtain the ambient temperature then adapt task periods and operating frequency. When the on-chip temperature is measured every  $\Delta t$ , the ambient temperature can be computed from Eq. (3) as:

$$T_a = \frac{RC_{th}}{\Delta t} (T_{[k]} - T_{[k-1]}) + T_{[k-1]} - R_{th} \cdot P_{tot}. \quad (10)$$

where  $T_{[k]}$  and  $T_{[k-1]}$  are the two consecutive temperature measurements, and  $P_{tot}$  is the total power dissipation. By obtaining the ambient temperature at runtime, the task periods can be adapted according to the optimization for the current ambient temperature.



## V. EVALUATION

**Experiment Setup.** Our evaluation platform was Nvidia Tegra K1 with quad-core ARM A15 running between 0.2GHz to 2.1GHz rated at an 8 W thermal design power (TDP). For periodic task workloads, we used Mibench embedded benchmarks targeting real-time applications. To demonstrate the performance of our RT-DTM, we consider baseline without any thermal management (BASE) and dynamic thermal management using real-time DVFS (RT-DVFS) [14]. Under RT-DVFS, the operating frequency is reduced after the chip temperature hits the temperature threshold while the proposed scheme (RT-DTM) adapts task periods and lowers the frequency only if there is no feasible task period. Both schemes were only allowed to reduce the frequency to no less than the minimum scaling level that guarantees real-time deadlines. While both schemes meet real-time deadlines in our experiments, we focus on comparing the temperature control and utilization. The temperature threshold was set to 60°C for RT-DVFS and RT-DTM to ensure the threshold is reached, while no threshold was set for BASE.

**Performance Evaluation.** Fig. 3 shows the real-time chip temperature, operating frequency, and utilization. The trace for BASE showed that the processor operated at the maximum frequency with a processor utilization of 99.7%, which resulted in a high chip temperature of 71.3°C with a temperature variation of 1.51°C. With the thermal throttling disabled, the test workload may increase the chip temperature as much as possible, which may cause a thermal runaway or system failure. Under RT-DVFS, the chip temperature hits the threshold at 70s, so the operating frequency is reduced to 0.8GHz. However, the chip temperature remains close to the threshold so the operating frequency is reduced again to 0.4 GHz at 200s, resulting in the lowest utilization of 67.2%. With the reduced operating frequency, the chip temperature is dramatically reduced to the lowest temperature of 54.7°C with a high temperature variation of 2.0°C. This large variation indicates that discrete DVFS was too coarse to maintain a stable temperature. RT-DTM is shown to control the temperature close to the threshold, optimizing utilization under the temperature constraint. RT-DTM as opposed to RT-DVFS was able to maintain the maximum operating frequency most of the time by optimally adjusting task periods, achieving 79.4% processor utilization, an 18.2% improvement over RT-DVFS. Under RT-DTM, the chip temperature stably controlled operating at a 59.0°C with a low temperature variation of 0.8°C.

## VI. CONCLUSION

Thermal management of real-time embedded systems poses new challenges: execute real-time workloads while meeting the processor thermal constraints. To address this problem, we propose a *thermal-aware utilization bound* that allows a proactive bound of the processor utilization in the system design before the temperature reaches the threshold. The proposed *RT-DTM* adaptively optimizes the processor utilization to meet both real-time and thermal constraints. Our experimental

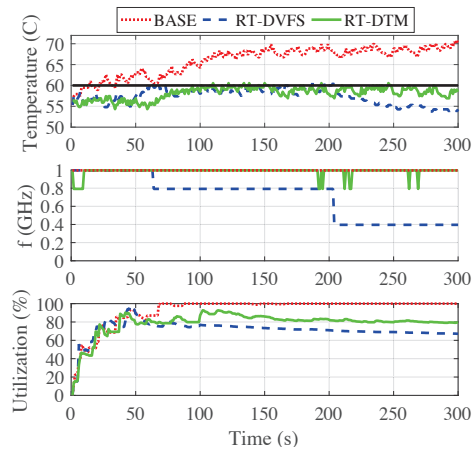


Fig. 3: Real-time temperature, operating frequency, utilization for different thermal management methods.

demonstration on an automotive controller showed that RT-DTM can significantly improve processor utilization over the existing methods.

## ACKNOWLEDGMENTS

This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the Grand ICT Research Center support program (IITP-2020-0-101741) supervised by the IITP(Institute for Information & communications Technology Planning & Evaluation).

## REFERENCES

- [1] Y. Liu, R. P. Dick, L. Shang, and H. Yang, "Accurate temperature dependent integrated circuit leakage power estimation is easy," in *DATE*, 2007.
- [2] V. Hanumaiah and S. Vrudhula, "Reliability-aware thermal management for hard real-time applications on multi-core processors," in *DATE*, 2011.
- [3] Y. Lee, H. S. Chwa, K. G. Shin, and S. Wang, "Thermal-aware resource management for embedded real-time systems," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 37, no. 11, pp. 2857–2868, 2018.
- [4] R. Ahmed, P. Ramanathan, and K. K. Saluja, "Temperature minimization using power redistribution in embedded systems," in *VLSI Design*, 2014.
- [5] Y. Lee, K. G. Shin, and H. S. Chwa, "Thermal-aware scheduling for integrated cpus-gpu platforms," *ACM Transactions on Embedded Computing Systems*, vol. 18, no. 5s, pp. 1–25, 2019.
- [6] Nvidia, "NVIDIA Tegra K1 Data Sheet," 2015.
- [7] M. R. Guthaus, J. S. Ringenberg, D. Ernst, T. M. Austin, T. Mudge, and R. B. Brown, "MiBench: A free, commercially representative embedded benchmark suite," in *Workshop on Workload Characterization*, 2001.
- [8] Y. Fu, N. Kottenstette, C. Lu, and X. D. Koutsoukos, "Feedback thermal control of real-time systems on multicore processors," in *EMSOFT*, 2012.
- [9] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware microarchitecture," in *ISCA*, 2003.
- [10] P. Kumar and L. Thiele, "Cool shapers: Shaping real-time tasks for improved thermal guarantees," *DAC*, 2011.
- [11] G. Quan, Y. Zhang, W. Wiles, and P. Pei, "Guaranteed scheduling for repetitive hard real-time tasks under the maximal temperature constraint," in *CODES*, 2008.
- [12] S. Hosseinimotlagh and H. Kim, "Thermal-aware servers for real-time tasks on multi-core gpu-integrated embedded systems," in *RTAS*, 2019.
- [13] S. Hosseinimotlagh, A. Ghahremanzhad, and H. Kim, "On dynamic thermal conditions in mixed-criticality systems," in *RTAS*, 2019.
- [14] P. Pillai and K. Shin, "Real-time dynamic voltage scaling for low-power embedded operating systems," in *SOSP*, 2001.