Energy-Efficient Speed Scheduling for Real-Time Tasks under Thermal Constraints

Shengquan Wang, Jian-Jia Chen, Zhenjun Shi, and Lothar Thiele *

Abstract

Thermal constraints have limited the performance improvement of modern computing systems in recent years. As a system could fail if the peak temperature exceeds its thermal constraint, overheating should be avoided while designing a system. Moreover, higher temperature also leads to higher leakage power consumption. This paper explores dynamic thermal management to minimize the energy consumption for a specified computing demand under the thermal constraint. We develop energy-efficient speed scheduling schemes for frame-based real-time tasks under thermal constraints. Experimental results reveal the effectiveness of the proposed scheme in terms of energy consumption with comparison to the reactive schemes in the literature.

Keywords: Energy-efficient scheduling, thermal constraint, Dynamic Voltage Scaling, Dynamic Thermal Management.

1 Introduction

Due to the significant increase of power density in modern circuits, low-power and energy-efficient designs have played important roles for designing modern computing systems. The reduction of power consumption can enhance the system reliability, reduce the cost of packaging, prolong the battery lifetime for embedded systems, and cut the power bills for server systems.

There are two major sources of power consumption of a processing unit [15]: dynamic power consumption due to switching activities and leakage power consumption due to the leakage current. Dynamic voltage/speed scaling (DVS) approach has been designed to reduce the dynamic power consumption with different execution speeds. Many technologies, such as Intel SpeedStep® and AMD PowerNOW!™, have implemented dynamic voltage/speed scaling. The power consumption of processors with DVS is a convex and increasing function of processor speeds, which is highly dependent on the hardware designs. Well-known DVS processors for embedded systems are Transmeta Crusoe and the Intel XScale. By applying DVS, the system could adjust its execution speed dynamically to satisfy the performance requirement, such as the response time or the throughput requirements. In the past decade, the minimization of energy consumption for real-time DVS systems has been an active topic, such as [1, 13, 14, 20, 29]. The pursuit of energy efficiency balances the energy consumption and the performance requirement.

Meanwhile, along with the dramatic increase of power density, heat dissipation has become a prominent issue since costly cooling systems must be adopted to prevent from overheating. Current estimates are that cooling solutions are rising at $1 to $3 per watt of heat dissipated [24]. Energy and temperature are both related to power consumption, but they are physical entities with different properties. Energy-efficient scheduling focuses on dealing with the *accumulative* power consumption, while thermal-aware scheduling focuses on handling the *peak* temperature [3, 6, 7]. An energy-inefficient scheduling could exhaust the battery. However, the system could fail if the peak temperature of a device exceeds its thermal threshold.

When the maximum thermal threshold is specified as the thermal constraint, the system should adjust the execution speed or execution mode to satisfy the thermal constraint. To maximize the performance under the thermal constraint, Dynamic Thermal Management (DTM) [5, 11, 24] has been proposed to adopt DVS to prevent the system from overheating. Bansal et al. [2] developed an algorithm to maximize the workload that can complete in a specified time interval without violating the thermal constraints. Wang et al. [26] developed reactive speed control for frame-based real-time

*Wang and Shi are with the Department of Computer and Information Science, the University of Michigan-Dearborn, USA. E-mail: {shqwang,zjshi}@umd.umich.edu. their work is sponsored by NSF CAREER Grant No. CNS-0746906. Chen and Thiele are with Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology, Zurich, Switzerland. Email: {jchen,thiele}@tik.ee.ethz.ch.

tasks and provided schedulability tests, in which all the tasks have the same period. In [25], delay analysis is performed under reactive speed control for general task arrivals. Chen et al. [8] developed proactive speed control to adapt dynamic speeds for different temperature for scheduling real-time tasks. Zhang and Chatha [31] provided approximation algorithms to minimize the completion time, while each task is restricted to execute at one speed. Thermal issues for multiprocessor systems have also been explored [6, 9, 12, 19], in which heat transfer between cores/processors makes the analysis more complicated.

Since there exist both energy-efficient approaches and thermal-constrained approaches in the literature, we are tempted to combine them together in order to achieve thermal-constrained energy-efficient design. However, this does not work. The existing energy-efficient approaches might make the system violate the thermal constraints, while the existing thermal-constrained scheduling approaches might make the system consume too much energy. Moreover, it has been shown that leakage current is dependent upon the temperature and it goes up rapidly when the temperature increases [16, 17]. For systems with temperature-dependent leakage power consumption, energy-efficient scheduling must prevent the system from over-heating so that the energy consumption can be reduced. Specifically, Yuan, Leventhal, and Qu [30] explored how to turn on/off a processor dynamically to cool down the system for energy reduction under a fixed supply voltage. For multiprocessor/multicore systems, Liu et al. [18] and Bao et al. [4] explored how to select voltages/speeds for the cores so that the energy consumption can be reduced without violating the timing constraints.

To the best of our knowledge, known results for thermal-constrained or thermal-aware energy-efficient scheduling, e.g., [4, 17, 30], do not dynamically change the supply voltage on a core since each core is fixed with a supply voltage due to the simplification of calculation. This paper explores how to apply DVS for energy consumption minimization under thermal constraints in uniprocessor systems and design the corresponding speed scheduling. By applying the derived speed scheduling scheme repetitively, we propose energy-efficient scheduling schemes to meet a given workload demand periodically under the thermal constraint. Experimental results reveal the effectiveness of the proposed scheme in terms of energy consumption in comparison to the reactive schemes in the literature.

The rest of this paper is organized as follows: Section 2 shows the system models and problem definition. Section 3 presents our schemes for energy consumption minimization. Practical design issues will be discussed in Section 4. The performance evaluation is presented in Section 5. We conclude this paper in Section 6.

2 System Models and Problem Definition

This section will lay out some system models used in our design, such as processor and power model, cooling model, and then present problem definition.

Processor and power model We explore thermal-aware scheduling on DVS processors. The power consumption $\Psi()$ is contributed by:

- *The speed-dependent power consumption* $\Psi_{dep}()$ mainly resulting from the charging and discharging of gates on the circuits. The speed-dependent power consumption could be modeled as a convex function of the processor speed such as the dynamic power consumption in CMOS processors [21]: $\Psi_{dep}(s) = C_{ef} V_{dd}^2 s$, where $s = \kappa_v \frac{(V_{dd}-V_t)^2}{V_{dd}}$. [1] We can further simplify the formula of the speed-dependent power consumption as $\Psi_{dep}(s) = hs^\gamma$, where h and γ are constants and $\gamma \leq 3$.
- *The speed-independent power consumption* $\Psi_{ind}()$ mainly resulting from leakage current. The speed-independent power consumption function of the system could be modeled as a nonnegative constant when leakage power consumption is independent of the temperature [7,28]. When the leakage power consumption is related to the temperature, we model the leakage power consumption by a linear function of the temperature [6]. Hence, the speed-independent power consumption is as follows: $\Psi_{ind}(\Theta) = \delta\Theta + \rho$, where Θ is the absolute temperature and δ and ρ are constants.

In this paper, we use the following formula as the overall power consumption

$$\Psi(s, \Theta) = \Psi_{dep}(s) + \Psi_{ind}(\Theta) = hs^\gamma + \delta\Theta + \rho. \qquad (1)$$

The speed $s = s(t)$ and temperature $\Theta = \Theta(t)$ are functions of time t. The number of CPU cycles (or workload) completed in the time interval $[t_0, t_1]$ is $\int_{t_0}^{t_1} s(t)dt$. The energy consumed in $[t_0, t_1]$ is $\int_{t_0}^{t_1} \Psi(s(t), \Theta(t))dt$.

Cooling model Although cooling is a complicated physical process, it could be approximately modelled by applying Fourier's Law. This is used in most existing thermal-aware system designs, such as those in

[1] C_{ef}, V_t, V_{dd}, and κ_v denote the effective switch capacitance, the threshold voltage, the supply voltage, and a hardware-design-specific constant, respectively. $V_{dd} \geq V_t \geq 0$; $\kappa_v, C_{ef} > 0$.

[2, 3, 6, 22, 25, 26, 31]. This paper adopts such approximation on the cooling process. The ambient temperature is assumed fixed. Formally, if we define $\Theta(t)$ as the temperature at time instant t, then

$$\begin{aligned} \Theta'(t) &= \hat{\alpha}\Psi(s(t), \Theta(t)) - \hat{\beta}(\Theta(t) - \Theta_a) \\ &= \hat{\alpha}(hs^\gamma(t) + \delta\Theta(t) + \rho) - \hat{\beta}(\Theta(t) - \Theta_a) \\ &= \alpha s^\gamma(t) - \beta\Theta(t) + \sigma, \end{aligned} \quad (2)$$

where Θ_a is the ambient temperature, $\hat{\alpha}$ is the coefficient for heating, $\hat{\beta}$ is the coefficient for cooling, α is $h\hat{\alpha}$, β is $\hat{\beta} - \hat{\alpha}\delta$, and σ is $\hat{\alpha}\rho + \beta\Theta_a$.

Based on (2), for notational simplicity, we define the *adjusted temperature* as $\theta(t) \stackrel{\text{def}}{=} \frac{\Theta(t)}{\alpha} - \frac{\sigma}{\alpha\beta}$. Therefore, the cooling and heating process can be simplified as

$$\theta'(t) = s^\gamma(t) - \beta\theta(t). \quad (3)$$

The overall power consumption can be rewritten in terms of s and θ as

$$\Psi(s, \theta) = hs^\gamma + \delta\alpha\theta + (\rho + \frac{\delta\sigma}{\beta}). \quad (4)$$

To simplify the presentation, for the rest of the paper, if the context is clear, we will use "temperature" to refer to "adjusted temperature", including thermal constraints. Given the initial temperature $\theta(t_0)$ at t_0, based on (3), the temperature at time t can be written as

$$\theta(t) = \int_{t_0}^{t} s^\gamma(\tau)e^{-\beta(t-\tau)}\mathrm{d}\tau + \theta(t_0)e^{-\beta(t-t_0)}. \quad (5)$$

Suppose that θ^* is the maximum thermal threshold in the processor (note that its value has been adjusted by the definition of $\theta(t)$). The *equilibrium* speed s_E is defined as the maximum *constant* speed that maintains the temperature under the maximum thermal constraint. Based on (3), we can obtain s_E as

$$s_E = (\beta\theta^*)^{\frac{1}{\gamma}}. \quad (6)$$

If we remove the constraint that the speed is constant to allow variable speed, we are able to design a better energy-efficient speed scheduling which respects the maximum thermal threshold.

Problem definition In this paper, we consider periodic real-time tasks. A periodic task is an infinite sequence of task instances, referred to as jobs. We focus on a set $\mathbf{T}$ of N frame-based periodic real-time, in which all tasks have the same period P [27]. The amount of the required computation cycles of task $T_i \in \mathbf{T}$ is C_i. The relative deadline of task T_i is D_i, in which $D_i \leq P$. We define Δ_i as the response time for task T_i. We assume an Earlier Deadline First (EDF) scheduling is used for job arrivals.

This paper copes with energy consumption minimization for a specified performance requirement under thermal constraints. We consider the tasks with the same deadline first (the extension to different deadlines are addressed in Subsection 3.3). Specifically, given a set of frame-based real-time tasks with total workload demand C and a common period P, the problem is to derive a speed assignment for providing C computation cycles before Δ under thermal constraints such that the resulting energy consumption is minimized, where $\Delta \leq P$. The maximum thermal threshold θ^* must be satisfied, while the initial temperature of the system is the ambient temperature. A speed scheduling s in time interval $[t_0, t_0+P]$ is said *feasible* if the amount of the completed cycles is no less than C, i.e., $\int_{t_0}^{t_0+P} s(t)dt \geq C$ without violating the maximum thermal threshold.

3 Thermal-Constrained Energy-Efficient Speed Scheduling

3.1 Repetitive Speed Scheduling

To guarantee the schedulability of frame-based real-time tasks, one has to perform schedulability tests for each individual period. Fortunately, the following lemmas show that we can guarantee schedulability tests over all time periods if we find a feasible speed scheduling over a single period with a specific initial temperature constraint:

Lemma 1 (Chen, Wang, Thiele [8]) *Let $s_\dagger$ be a feasible speed scheduling in time period $[t_0, t_0 + P]$ with the temperature $\theta(t_0) = \theta(t_0+P) = \theta_P$. Then, $s_\dagger$ is a feasible speed scheduling over all time periods.*

A feasible speed scheduling $s_\dagger$ is said to have a *converging initial temperature* θ_P if it satisfies the statement in Lemma 1. Moreover, we find out that a feasible speed scheduling that converges at some temperature is also a necessary condition for schedulability as shown in the following lemma:

Lemma 2 (Chen, Wang, Thiele [8]) *If there does not exist any feasible speed scheduling $s_\dagger$ that has a converging initial temperature no more than θ^*, there does not exist any feasible speed scheduling under the thermal and timing constraint.*

By Lemmas 1 and 2, applying a feasible speed scheduling, if there exists, *repetitively* guarantees the schedulability and feasibility of the resulting speed scheduling. In this paper, we explore the *thermal-constrained energy-efficient speed scheduling* problem to derive the feasible speed scheduling with the minimum *converging energy consumption*. The converging energy consumption of a feasible speed scheduling s is the energy consumption $\int_{t_0}^{t_0+P} \Psi(s, \theta)dt$ in a time interval $[t_0, t_0 + P]$ when the temperature $\theta(t_0)$ at time t_0 is the converging initial temperature θ_P of the feasible scheduling. We say that a feasible speed scheduling is *optimal* (in terms of energy consumption minimization) if the converging energy consumption is the minimum among all feasible speed scheduling.

Motivational example Consider the scheduling of a task whose period is $P = 0.1$ sec, relative deadline D is 0.08 sec, and workload demand C is 0.16G cycles. The task can be completed in time by executing at speed 2GHz. The hardware parameters here are assumed as follows: $\hat{\beta} = 12.5$, $\delta = 0.01$, $\rho = 0.1$, $\hat{\alpha} = 17.5$, $h = 6$, $\gamma = 3$, $\Theta_a = 30$°C, $\Theta^* = 89.25$°C, while the dynamic power consumption is normalized to 1GHz. The equilibrium speed in this example is 1.907GHz. To minimize the energy consumption without considering the thermal-aware power consumption, it has been shown that executing at speed $\frac{C}{D}$ minimizes the energy consumption [1]. However, executing at speed 2GHz leads to a scheduling with peak temperature 90.45°C, which violates the thermal constraint.

We could apply the reactive speed approach proposed in [26] to derive speed scheduling under thermal constraints. For the reactive speed approach, the task is executed at a constant speed high speed s_H when the temperature is lower than the thermal constraint θ^*, where $s_H > s_E$. Then, once the temperature reaches the thermal constraint, the task is executed at speed s_E. The delay analysis in [26] can then be applied to analyze the response time. As our objective is to minimize the energy consumption, it is straightforward that we would like to complete the task just in time. Therefore, one possible way is to extend the reactive speed approach by choosing a proper s_H so that the response time is equal to D. In this example, setting s_H as 2.63GHz leads to a feasible reactive speed scheduling with response time equal to D. The feasible reactive speed scheduling with the minimum converging energy consumption consumes 4.383J.

One could also apply the proactive speed approach proposed in [8]. However, it is not clear how to extend the algorithms in [8] directly for energy consumption minimization. In this paper, we will adopt the optimal control framework to derive the proactive speed scheduling. By applying the optimal control framework, we will execute at a higher speed at the beginning of a period, and the slow down smoothly without violating the thermal constraints. The feasible proactive speed scheduling with the minimum converging energy consumption consumes 4.206J.

Figure 1 illustrates the temperature, speed, and power consumption curves for the above schedules when they converge. Therefore, since known results for thermal-constrained or thermal-aware energy-efficient scheduling, e.g., [4,17,30], do not dynamically change the supply voltage on a processor, they can not deal with these cases. In this paper, we will present optimal speed scheduling schemes to derive energy-efficient speed scheduling under thermal constraints.

3.2 Energy-Efficient Speed Scheduling

Given a period $[t_0, t_0 + P]$, the accumulated jobs of all tasks will be executed during $[t_0, t_1]$, where $t_1 = t_0 + D$, where D is the deadline. During the interval $[t_1, t_0 + P]$, the processor is idle and the temperature will cool down.

In order to minimize the energy consumption, first we would like to investigate the energy consumption during the period $[t_0, t_0 + P]$. Based on (4) and (3), the overall power consumption can be rewritten as

$$\begin{aligned} \Psi(s, \theta) &= hs^\gamma + \delta\alpha \frac{s^\gamma - \theta'}{\beta} + (\rho + \frac{\delta\sigma}{\beta}) && (7) \\ &= (h + \frac{\delta\alpha}{\beta})s^\gamma - \frac{\delta\alpha}{\beta}\theta' + (\rho + \frac{\delta\sigma}{\beta}). && (8) \end{aligned}$$

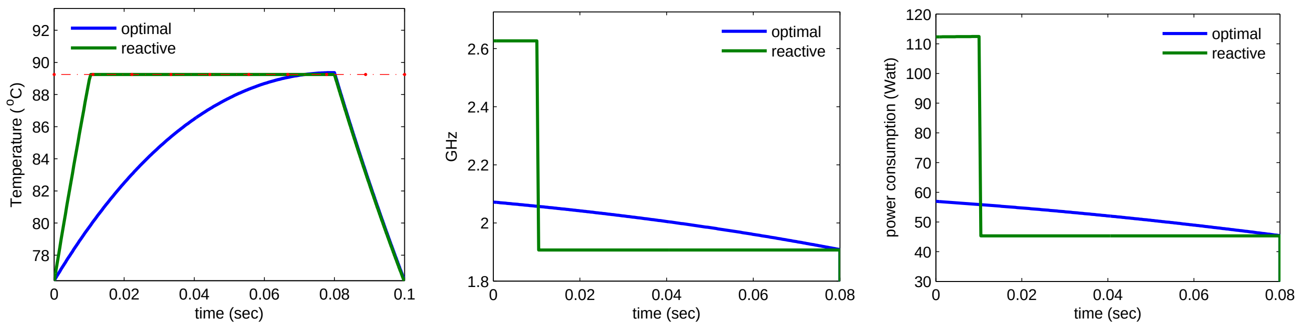
In $[t_0, t_1]$, we have an undetermined speed $s(t)$; In $[t_1, t_0 + P]$ the speed $s(t) = 0$. Therefore, the energy consumption during the interval $[t_0, t_0 + P]$ can be calculated as

$$\begin{aligned} \int_{t_0}^{t_0+P} \Psi(s, \theta)dt &= (h + \frac{\delta\alpha}{\beta}) \int_{t_0}^{t_1} s^\gamma(t)dt \\ &\quad - \frac{\delta\alpha}{\beta}(\theta(t_0 + P) - \theta(t_0)) \\ &\quad + (\rho + \frac{\delta\sigma}{\beta})P. && (9) \end{aligned}$$

Recall that we assume $\theta(t_0)$ eventually converges to θ_P, therefore we can assume $\theta(t_0) = \theta(t_0 + P) = \theta_P$. Then the energy consumption during the interval $[t_0, t_0 + P]$ can be simplified as

$$\int_{t_0}^{t_0+P} \Psi(s, \theta)dt = (h + \frac{\delta\alpha}{\beta}) \int_{t_0}^{t_1} s^\gamma(t)dt + (\rho + \frac{\delta\sigma}{\beta})P. \quad (10)$$

Based on (10), to minimize the energy consumption during $[t_0, t_1]$, we only need to minimize $\int_{t_0}^{t_1} s^\gamma(t)dt$, which is subject to the workload constraint $\int_{t_0}^{t_1} s(t)dt \geq C$ and the thermal equation and other thermal constraints.



(a) temperature curve (b) speed curve (c) power consumption for execution

Figure 1. Temperature, speed, and power consumption curves for two speed schedules, where the dotted line in 1(a) is the thermal constraint.

Given the proceeding discussion, we can state the optimization problem as:

$$\text{minimize} \quad \int_{t_0}^{t_1} s^\gamma(t)dt \tag{11a}$$

$$\text{subject to} \quad \int_{t_0}^{t_1} s(t)dt \geq C, \tag{11b}$$

$$\theta'(t) = s^\gamma(t) - \beta\theta(t), \tag{11c}$$

$$\theta(t) \leq \theta^*, \tag{11d}$$

$$\theta(t_0) = \theta_P. \tag{11e}$$

In (11), (11b) is the workload constraint, (11c) is the Fourier Law, (11d) is the thermal constraint, and (11e) is the initial converging temperature constraint. Since θ_P can be any value, the constraint (11e) can be ignored.

This is an optimal control problem with a pure state inequality constraint. The speed $s(t)$ is the control variable, and the temperature $\theta(t)$ is the state variable. The problem is to find an admissible control $s(t)$, which minimize the objective function (11a) subject to the isoperimetric constraint (11b), the state equation (11c), and the pure state inequality constraint (11d). We adopt the maximum principle with a direct adjoining approach [10, 23]. Since our optimization is a minimization problem, first we can convert this minimization problem into the standard maximum problem by multiplying the objective function with -1.

In the maximum principle, we need to define Hamiltonion and Lagrangian. The Hamiltonion is

$$H = -s^\gamma(t) + \lambda(t)(s^\gamma(t) - \beta\theta(t)), \tag{12}$$

where $\lambda(t)$ is an adjoint variable for the state equation (11c). Now we form the Lagrangian as

$$L = H + \mu s(t) + \nu(t)(\theta^* - \theta(t)), \tag{13}$$

where μ is a constant Lagrangian multiplier for the isoperimetric constraint (11b), which should satisfy

$$\mu \geq 0, \tag{14}$$

and $\nu(t)$ is a Lagrangian multiplier for the pure state inequality constraint (11d), which also should satisfy the complementary slackness conditions

$$\nu(t) \geq 0, \quad \nu(t)(\theta^* - \theta(t)) = 0. \tag{15}$$

Furthermore, the optimal trajectory must satisfy

$$\frac{\partial L}{\partial s} = \gamma(\lambda(t) - 1)s^{\gamma-1}(t) + \mu = 0. \tag{16}$$

Based on the above formula, the speed $s(t)$ can be written as

$$s(t) = \left(\frac{\gamma}{\mu}(1 - \lambda(t))\right)^{\frac{1}{1-\gamma}}. \tag{17}$$

From the Larangian we also get an adjoint equation

$$\dot{\lambda}(t) = -\frac{\partial L}{\partial \theta} = \beta\lambda(t) + \nu(t). \tag{18}$$

- If $\theta(t) < \theta^*$ in some interval, by (15) we have $\nu(t) = 0$. Therefore, based on (18) λ can be written as

$$\lambda(t) = \kappa e^{\beta(t-t_0)}. \tag{19}$$

Furthermore the speed $s(t)$ in (17) can be rewritten as

$$s(t) = \left(\frac{\gamma}{\mu}(1 - \kappa e^{\beta(t-t_0)})\right)^{\frac{1}{1-\gamma}}. \tag{20}$$

In our performance evaluation, we observe that under the optimization the speed function defined in (20) is a non-increasing function and the resulting temperature is increasing in the beginning and might be decreasing later.

- If $\theta(t) = \theta^*$ in some interval, then the speed must be $s(t) = s_E$.

In order to address the optimal control issue, we consider the state (i.e., thermal) constraint (11d) with the following scenarios:

Constant Speed Scheduling If $\theta(t) < \theta^*$ during $[t_0, t_1]$, by an additional boundary constraint $\lambda(t_1)(\theta^* - \theta(t_1)) = 0$, we have $\lambda(t_1) = 0$. Applying this boundary condition into (19), we have $\kappa = 0$, and the speed scheduling becomes a constant speed control as

$$s(t) = (\frac{\gamma}{\mu})^{\frac{1}{1-\gamma}}. \quad (21)$$

Recall that $s(t)$ is subject to the workload constraint (11b). In all our cases, the optimal solution will be reached as the constraint becomes equality. Then we have

$$s(t) = \frac{C}{t_1 - t_0}. \quad (22)$$

In this scenario, the temperature never violates the thermal constraint (11d).

Smooth Speed Scheduling If the thermal violation occurs when we apply the constant speed control in the previous case, then we need to revise the solution. Our first try is to find a non-zero κ and then a decreasing speed $s(t)$ so that the temperature does not violate the thermal constraint (11d) except at t_1 (i.e., $\theta(t) < \theta^*$ during $[t_0, t_1)$ and $\theta(t_1) = \theta^*$), which will minimize the objective function. By the temperature constraint at t_0 and t_1: $\theta(t_0) = \theta_P$ and $\theta(t_1) = \theta^*$. By the temperature formula in (5), we have

$$\theta^* = \int_{t_0}^{t_1} s^\gamma(t) e^{-\beta(t_1 - t)} dt + \theta_P e^{-\beta(t_1 - t_0)}. \quad (23)$$

We also know during $[t_1, t_0 + P]$ the speed is zero and the temperature changes from θ^* to θ_P, then we have $\theta_P = \theta^* e^{-\beta(t_0 + P - t_1)}$. Applying this into (23), we have another thermal constraint:

$$\int_{t_0}^{t_1} s^\gamma(t) e^{-\beta(t_1 - t)} dt = \theta^*(1 - e^{-\beta P}). \quad (24)$$

Together with the workload equality constraint in (11b), we are able to obtain μ and κ in (20).

Piecewise Speed Scheduling If we can not find a feasible speed scheduling with the above approach, then we might have $\theta(t) = \theta^*$ during an subinterval $[u, t_1] \subseteq [t_0, t_1]$, and $\theta(t) < \theta^*$ during $[t_0, u)$. This scenario includes the previous case by setting $u = t_1$.

During $[t_0, u)$, the speed $s(t)$ follows (20). Note that κ might be zero. If $\kappa \neq 0$, then it also requires a smooth switching condition at u, i.e., $\theta(u) = \theta^*$ and $\theta'(u) = 0$. Therefore, we have the smooth switching condition at u

$$s(u) = s_E \text{ as } \kappa \neq 0. \quad (25)$$

In addition, the speed is also subject to the temperature constraints at t_0, u, and t_1: $\theta(t_0) = \theta_P$, $\theta(u) = \theta^*$, and $\theta(t_1) = \theta^* = \theta_P e^{\beta(t_0 + P - t_1)}$. Similar to the temperature analysis in the previous scenario, we have another thermal constraint:

$$\int_{t_0}^{u} s^\gamma(t) e^{-\beta(u - t)} dt = \theta^*(1 - e^{-\beta(P - t_1 + u)}). \quad (26)$$

Recall that the optimization is reached as the workload constraint becomes equality $\int_{t_0}^{t_1} s(t)dt = C$. Then we have an additional constraint

$$\int_{t_0}^{u} s(t)dt + s_E(t_1 - u) = C. \quad (27)$$

Based on Equations (20), (25), (26), and (27), we can solve μ, κ, and u.

In the piecewise speed scheduling, if $\kappa = 0$, this will be the reactive speed scheduling with the best high constant speed during $[t_0, u]$ to minimize the energy. However, in some case, this reactive speed scheduling might not be feasible. Then, we have to set $\kappa \neq 0$, which is more general case of the piecewise speed scheduling.

3.3 Extensions

Tasks with different deadlines In Section 3, we have assumed that all tasks have the same relative deadline. For tasks with different deadlines, with the EDF scheduling, the task instance with the earliest deadline will be executed first. Without loss of generality, we assume task instances in $\mathbf{T}$ are non-decreasingly ordered according to their relative deadlines in time period $[t_0, t_0 + P]$ as: $D_i \leq D_j$ as $i < j$. The workload demand right before D_i is $\sum_{j=1}^{i} C_j$. We define Δ_i as the response time of Task i. To meet timing constraints, we need to ensure that $\Delta_i \leq D_i$ for $i = 1, 2, \ldots, N$.

The derived speed scheduling scheme in Section 3 might not satisfy the timing and thermal constraints for all tasks. We will revise the scheme.

We start with the task T_N. We can apply the approach in Section 3 for all cumulative tasks. If there exists a feasible speed scheduling $s_\dagger$, then we calculate each individual response time Δ_j in the derived energy-efficient scheduling by setting D_N as the deadline. If $\Delta_j \leq D_j$, $\forall j \leq N$, then we return the feasible schedule $s_\dagger$. Otherwise, we find earlier-deadline task T_{i^*} with timing violation. We then apply the approach in Section 3 for the cumulative tasks during $[t_0, t_0 + D_{i^*}]$ by replacing the workload C with $\sum_{j=1}^{i^*} C_j$ and setting the speed at s_E from $t_0 + D_{i*}$ to the completion time. The same procedure is repeatedly applied. It returns either a feasible and energy-efficient speed scheduling or non-schedulable.

Tasks with different arrival-times As shown in [26], the *critical instant* for thermal-constrained scheduling of periodic real-time tasks is at the moment that all instances arrive at the same time when all tasks are with the same relative deadline. As a result, we can perform schedulability test based on the assumption that all the tasks arrive at the same time. However, this might lead to a solution that intends to execute some workload before it arrives. Suppose $s_\dagger$ is the speed scheduling that converges at temperature θ_P. Then, if applying $s_\dagger$ to task set $\mathbf{T}$ leads to a speed scheduling that executes some computation before the workload arrives, we have to set speed to 0 when there is no task instance in the system. When a task instance arrives at time t with instantaneous temperature $\theta(t) < \theta^*$, we refer to the speed scheduling $s_\dagger$ to find time instant w with $\theta(t_0 + w) = \theta(t)$ and $s_\dagger(t_0 + w) > s_E$ and then follow the speed scheduling $s_\dagger$ to serve the incoming tasks. When tasks have the same relative deadline but different arrival times, it is not difficult to see that the above speed scheduling does not make any task miss its deadline under the thermal constraint if $s_\dagger$ is a feasible speed scheduling.

4 Practical Design Issues

As current DVS platforms have limitations on the maximum and minimum available speeds, we will show how to revise the approaches in this paper to deal with systems with bounded speeds. Moreover, since modern commercial processors have discrete speeds only, we will explore how to deal with systems with discrete speeds.

Bounded speeds Suppose that $s_{\max}$ is the maximum available speed of the processor. Without loss of generality, we assume $s_{\max} > s_E$. Otherwise, the solution is trivial. We only consider the case that the derived solution in Section 3 violates the speed constraint $s_{\max}$ at some time instance. In such a case, we have to revise the derivation of speed scheduling. Recall that the derived speed scheduling is an increasing function. Therefore, the processor should run at the speed $s_{\max}$ for a while and then start to slow it down at a time instant w ($w > t_0$). The corresponding temperature curve is increasing until the temperature reaches the threshold θ^* at u (defined in previous section), and then it runs at the speed s_E. During $[w, u]$, the temperature curve follows the speed scheduling in (20). We have assumed $w > u$, otherwise the case is trival.

The workload constraint in (11b) will be revised as

$$\int_w^u s(t)dt + s_{\max}(w - t_0) + s_E(t_1 - u) = C. \tag{28}$$

Denote the temperature at the moment w as θ_w. The temperature constraints at w and u can be revised as

$$\theta_w = \frac{1}{\beta} s_{\max}^\gamma + (\theta_0 - \frac{1}{\beta} s_{\max}^\gamma) e^{-\beta(w-t_0)}, \tag{29}$$

$$\theta^* = \int_w^u s^\gamma(t) e^{-\beta(u-t)} dt + \theta_w e^{-\beta(u-w)}. \tag{30}$$

Then we follow the same procedure in Section 3. In comparison, here we have one more parameter w, but we also have one more equation (29). Therefore we can derive the speed curve.

For the case that there is a minimum speed for execution, the procedure is similar.

Processors with discrete speeds In Section 3, we have assumed that the processor supports continuous speed. For commercial processors, only discrete speeds might be available. Then we can first derive a solution by assuming continuously available speeds as the approaches in Sections 3. Then, we use some available speeds to approximate the speed scheduling by restricting and perform schedulability tests by applying similar procedures.

For instance, we can find a high speed s_H to approximate the non-constant heating speed scheduling derived in Section 3. Suppose that $s_\dagger$ is a speed scheduling that converges at temperature θ_P. We can set s_H by using the following equation: $\theta^* = \frac{1}{\beta} s_H^\gamma (1 - e^{-\beta(u-t_0)}) + \theta_P e^{-\beta(u-t_0)}$, where the temperature reaches the highest at u in speed scheduling $s_\dagger$. Whether the task set is schedulable under the thermal and timing constraints can then be derived. Similarly, for cooling speed scheduling derived in Section 3, we can use a speed $s_L < s_E$ to approximate it. The overhead of switching processor speeds can also be taken into account.

5 Performance Evaluation

5.1 Setup

This section provides performance evaluation of our proposed approaches. Throughout this section, we use the following settings: The power consumption function $\Psi(s, \Theta)$ is assumed to be $6s^3 + 0.01\Theta + 0.1$ Watt, where s

is normalized to GHz. The cooling factor $\hat{\beta}$ of the processor is set as $1/0.08$ sec^{-1}, while $\hat{\alpha}$ is 17.5 K/Joule. The ambient temperature Θ_a is 30°C.

We evaluate three different sets of workloads by varying their periods, relative deadlines, and workload demands. To characterize the workload demand, the *average speed* is defined as the workload demand divided by the relative deadline, i.e., C/D. By varying the relative deadlines, we generate frame-based real-time task sets with a common relative deadline in the range of $[0.5, 0.85]P$, in which the period is in the range of $[0.05, 0.125]$ sec and the average speed is 2GHz. By varying the periods, we generate frame-based real-time task sets with periods in the range of $[0.05, 0.125]$ sec, in which the common relative deadline is $0.7P$ and the average speed is 2GHz. By varying the periods, we generate frame-based task sets with period in the range of $[0.05, 0.125]$ sec and the common relative deadline $0.8P$, where the average speed is in the range of $[1.5, 2.5]$GHz.

As the thermal constraint Θ^* affects the performance, the thermal constraint is set dynamically. Suppose that θ_{peak} is peak (adjusted) temperature by scheduling the task set at the average speed. Clearly, according to our analysis in Section 3, if the peak temperature is lower than the thermal constraint, executing at the average speed can optimize the energy consumption. Throughout the simulation, we will assume that the peak temperature θ_{peak} violates the (adjusted) thermal constraint a little bit by setting θ^* as $0.98\theta_{peak}$.

We evaluate the following algorithms:

- *Reactive*: Apply the reactive scheme proposed in [25, 26] by using two speeds for execution, in which one is the equilibrium speed s_E and the other is a speed $s_H > s_E$. Note that as s_H could be defined as any speed higher than s_E. To make our comparison fair, we find the best s_H to minimize the energy consumption by completing the tasks just in time.

- *Optimal*: Apply the energy-efficient optimal speed scheduling scheme proposed in this paper.

As the reactive speed scheduling is a special case of the optimal speed scheduling, for an input instance, the optimal speed scheduling is always no worse than the reactive speed scheduling. We report the *normalized converging energy* consumptions of the derived speed schedulings in the evaluated algorithms, where the normalized converging energy is defined as the converging energy consumption of the reactive speed scheduling divided by that of the optimal speed scheduling.

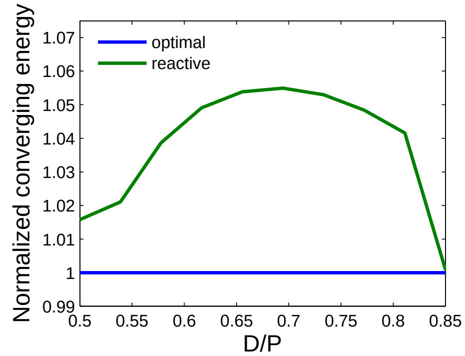


Figure 2. Simulation results for task sets with period in the range of $[0.05, 0.125]$ sec and average speed 2GHz.

5.2 Simulation Results

Figure 2 presents the evaluation results of the normalized converging energy consumption by varying the relative deadlines. By taking a proper speed s_H, the energy consumption of the reactive speed scheduling is, in general, quite close to the optimal solution. When D is larger, since there is more room for optimization, the optimal speed scheduling can improve the energy consumption more. However, when D is quite close to the period, i.e., $D > 0.8P$, the satisfaction of the thermal constraint is more important than the energy consumption minimization. Therefore, the normalized converging energy decreases when $D > 0.7P$.

Figure 3 presents the evaluation results of the normalized converging energy consumption by varying the periods. According to our settings, when the period is larger, we have more slack for cooling. Therefore, compared to the reactive speed scheduling, the speed variance in the optimal speed scheduling is more close to the average speed. As a result, when the period is larger, the normalized converging energy consumption becomes larger. Figure 4 presents the evaluation results of the normalized converging energy consumption by varying the average speeds. As the period and the relative deadline are fixed in

Figure 4, varying the average speed would mostly lead to a speed scheduling with a constant scale up/down on the optimal and the reactive speed schedulings in their speed curves. Therefore, the normalized converging energy does not vary too much when varying the average speed.

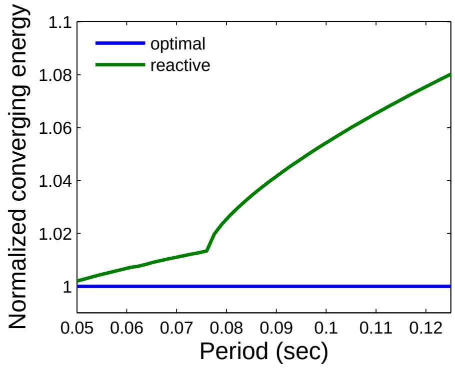


Figure 3. Simulation results for task sets with relative deadline $0.7P$ and average speed 2GHz.

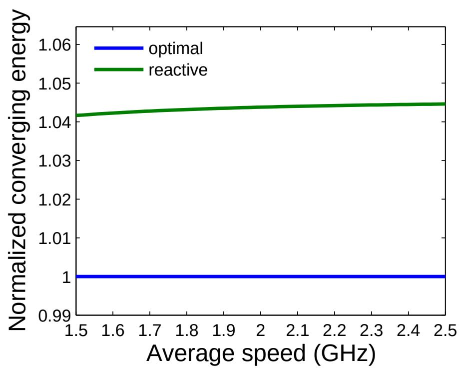


Figure 4. Simulation results for task sets with period in the range of $[0.05, 0.125]$ sec and relative deadline $0.8P$.

6 Conclusion

In this paper, we have presented optimal control speed scheduling schemes for real-time tasks to minimize the energy consumption under both timing and thermal constraints. We first showed that executing at a constant speed just in time minimizes the energy consumption if the thermal constraint is satisfied. However, when the thermal constraint is violated by executing at the constant speed, we presented optimal control speed scheduling schemes (extended from the proactive speed scheduling in [8]), by considering smooth speed scheduling and piecewise speed scheduling. We also presented the extensions to cope with non-ideal processors with bounded speed limitations, discrete speeds, or tasks with different relative deadlines or arrival times. Experimental results showed that the proposed schemes can derive more energy-efficient solutions than the reactive speed scheduling in [26] under the thermal constraint.

References

[1] H. Aydin, R. Melhem, D. Mossé, and P. Mejía-Alvarez. Dynamic and aggressive scheduling techniques for power-aware real-time systems. In *IEEE Real-Time Systems Symposium*, 2001.

[2] N. Bansal, T. Kimbrel, and K. Pruhs. Dynamic speed scaling to manage energy and temperature. In *Symposium on Foundations of Computer Science*, 2004.

[3] N. Bansal and K. Pruhs. Speed scaling to manage temperature. In *Symposium on Theoretical Aspects of Computer Science*, 2005.

[4] M. Bao, A. Andrei, P. Eles, and Z. Peng. Temperature-aware voltage selection for energy optimization. In *DATE*, pages 1083–1086, 2008.

[5] D. Brooks and M. Martonosi. Dynamic thermal management for high-performance microprocessors. In *International Symposium on High-Performance Computer Architecture*, 2001.

[6] T. Chantem, R. P. Dick, and X. S. Hu. Temperature-aware scheduling and assignment for hard real-time applications on MPSoCs. In *Design, Automation and Test in Europe*, 2008.

[7] J.-J. Chen, C.-M. Hung, and T.-W. Kuo. On the minimization of the instantaneous temperature for periodic real-time tasks. In *IEEE Real-Time and Embedded Technology and Applications Symposium*, 2007.

[8] J.-J. Chen, S. Wang, and L. Thiele. Proactive speed scheduling for real-time tasks under thermal constraints. In *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2009.

[9] N. Fisher, J.-J. Chen, S. Wang, and L. Thiele. Thermal-aware global real-time scheduling on multicore systems. In *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, 2009.

[10] R. F. Hartl, S. P. Sethi, and R. G. Vickson. A survey of the maximum principles for optimal control problems with state constraints. *SIAM Rev.*, 37(2):181–218, 1995.

[11] M. Huang, J. Renau, S.-M. Yoo, and J. Torrellas. A Framework for Dynamic Energy Efficiency and Temperature Management. In *International Symposium on Microarchitecture*, 2000.

[12] W.-L. Hung, Y. Xie, N. Vijaykrishnan, M. T. Kandemir, and M. J. Irwin. Thermal-aware task allocation and scheduling for embedded systems. In *ACM/IEEE Conference of Design, Automation, and Test in Europe*, 2005.

[13] S. Irani, S. Shukla, and R. Gupta. Algorithms for power savings. In *ACM-SIAM Symposium on Discrete Algorithms*, 2003.

[14] T. Ishihara and H. Yasuura. Voltage scheduling problems for dynamically variable voltage processors. In *International Symposium on Low Power Electronics and Design*, 1998.

[15] R. Jejurikar, C. Pereira, and R. Gupta. Leakage aware dynamic voltage scaling for real-time embedded systems. In *the Design Automation Conference*, 2004.

[16] W. Liao, L. He, and K. M. Lepak. Temperature and supply voltage aware performance and power modeling at microarchitecture level. *IEEE Trans. on CAD of Integrated Circuits and Systems*, 24(7):1042–1053, 2005.

[17] Y. Liu, R. P. Dick, L. Shang, and H. Yang. Accurate temperature-dependent integrated circuit leakage power estimation is easy. In *DATE*, pages 1526–1531, 2007.

[18] Y. Liu, H. Yang, R. P. Dick, H. Wang, and L. Shang. Thermal vs energy optimization for dvfs-enabled processors in embedded systems. In *ISQED*, pages 204–209, 2007.

[19] S. Murali, A. Mutapcic, D. Atienza, R. Gupta, S. Boyd, and G. D. Micheli. Temperature-aware processor frequency assignment for mpsocs using convex optimization. In *IEEE/ACM international conference on Hardware/software codesign and system synthesis*, 2007.

[20] G. Quan, L. Niu, X. S. Hu, and B. Mochocki. Fixed priority scheduling for reducing overall energy on variable voltage processors. In *IEEE Real-Time Systems Symposium*, 2004.

[21] J. M. Rabaey, A. Chandrakasan, and B. Nikolic. *Digital Integrated Circuits*. Prentice Hall, 2nd edition, 2002.

[22] J. E. Sergent and A. Krum. *Thermal Management Handbook*. McGraw-Hill, 1998.

[23] S. P. Sethi and G. L. Thompson. *Optimal Control Theory: Applications to Management Science and Economics, 2ed.* Kluwer Academic Publishers, 2000.

[24] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan. Temperature-aware microarchitecture. In *International Symposium on Computer Architecture*, 2003.

[25] S. Wang and R. Bettati. Delay analysis in temperature-constrained hard real-time systems with general task arrivals. In *IEEE Real-Time Systems Symposium*, 2006.

[26] S. Wang and R. Bettati. Reactive speed control in temperature-constrained real-time systems. *Real-Time Systems Journal*, 39(1-3):658–671, 2008.

[27] R. Xu, D. Mossé, and R. G. Melhem. Minimizing expected energy in real-time embedded systems. In *EMSOFT*, 2005.

[28] R. Xu, D. Zhu, C. Rusu, R. Melhem, and D. Moss. Energy efficient policies for embedded clusters. In *ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems*, 2005.

[29] F. Yao, A. Demers, and S. Shenker. A scheduling model for reduced CPU energy. In *Symposium on Foundations of Computer Science*, 1995.

[30] L. Yuan, S. Leventhal, and G. Qu. Temperature-aware leakage minimization technique for real-time systems. In *ICCAD*, pages 761–764, 2006.

[31] S. Zhang and K. S. Chatha. Approximation algorithm for the temperature-aware scheduling problem. In *International Conference on Computer-Aided Design*, 2007.