J Supercomput (2015) 71:2565–2584
DOI 10.1007/s11227-015-1405-0

Energy calculation for periodic multi-core scheduling in system thermal steady state with consideration of leakage and temperature dependency

Ming Fan[1] · **Rong Rong**[2] · **Shuo Liu**[3] · **Gang Quan**[3]

Published online: 29 March 2015

Abstract Energy estimation is a fundamental problem in energy-aware design and analysis. How to effectively and efficiently calculate the energy consumption, particularly when moving from uni-core platform to multi-core platform, is a critical issue. Moreover, when taking the interdependency between power and temperature into account, the estimation of energy consumption becomes more challenging. In this paper, we present a closed-form analytical solution to calculate the system thermal steady-state energy consumption for a periodic voltage schedule on a multi-core platform, with the leakage/temperature dependency taken into consideration. Our experiments show that the proposed method can achieve an average speedup of 15× over the existing related work, with a relative error no more than 1.5 %.

Keywords Energy calculation · Multi-core platform · Leakage/temperature dependency · Thermal steady state · Periodic scheduling

✉ Ming Fan
mingfan@broadcom.com

Rong Rong
rrong001@fiu.edu

Shuo Liu
sliu005@fiu.edu

Gang Quan
gang.quan@fiu.edu

1 Broadcom Corporation, San Jose, USA

2 School of Computing and Information Sciences, Florida International University, Miami, USA

3 Department of Electrical and Computer Engineering, Florida International Univesity, Miami, USA

1 Introduction

Multi-core architecture has been widely accepted as the most important technology in the future industrial market. By providing multiple processing cores on a single chip, multi-core systems, compared with the traditional single-core systems, can significantly increase the computing performance while relaxing the power requirement. Most of the major chip manufacturers have already launched 16-core chips into the market, i.e., AMD $Opteron^{TM}$ *6300 Series* [1]. It is not surprising that in the coming future, hundreds or even thousands of cores will be integrated into a single chip [26]. The quickly emerging trend toward multi-core platforms brings urgent needs for effective and efficient techniques for the design of multi-core scheduling.

The increasing popularity of multi-core systems and the rising performance demand have made energy efficiency a critical design objective for system designers. Catalyzed by continuous transistor scaling, an exponential increase in transistor density for higher performance platforms has led to a sharp rise in power/energy consumption [3,5]. The continuously increased power consumption has resulted in soaring chip temperature[18], which adversely impacts the performance, reliability, and packaging/cooling costs [11,19]. More importantly, as design paradigm shifts to deep sub-micron domain, high chip temperature leads to a substantial increase in leakage power consumption [12]. For instance, Liao et al. [15] showed an increase in leakage power consumption by 38 % with chip temperature rising from 65 to 110 °C. This signifies the need for incorporating leakage/temperature dependency into the system power model.

A fundamental problem in energy-aware design is calculating the energy consumption for a design alternative. To accurately and also quickly estimate the energy consumption for a voltage/frequency scheduling on multi-core platforms, there are two major challenges: (1) how to address the interdependency of leakage and temperature appropriately, and (2) how to deal with the heat transfer among different processing cores. First, by considering the leakage/temperature dependency, the leakage power consumption (and thus the overall power consumption) varies with the temperature, and temperature changes with the power consumption as well. This interdependency between leakage and temperature makes the power calculation, and thus the energy calculation, much complicated and difficult. Second, by further considering the heat transfer among different cores, the solution of power consumption becomes even more challenging, i.e., leading to the problems of matrix exponential operation and its corresponding integration, which may not always have explicit analytical solutions.

In this paper, we study the energy estimation problem on multi-core platforms. Specifically, our research problem can be described as: given a periodic voltage schedule on a multi-core platform, how to effectively and efficiently calculate the energy consumption within any scheduling period in system thermal steady state, with consideration of the interdependency between leakage power and temperature. Compared with the related existing work, we have made a number of distinct contributions:

– First, to facilitate our approach, we develop an analytical method to rapidly calculate the temperature at any time instant, particularly in the system steady state.

– Secondly, we develop a closed-form analytical solution for calculating the overall energy consumption within any scheduling period of a periodic voltage schedule, with constraints of the interdependency between leakage and temperature. Moreover, based on our temperature calculation method, we further formulate the energy consumption of one scheduling period in system thermal steady state.
– We also conducted experiments to evaluate the accuracy and time efficiency of our proposed energy calculation method. The experimental results showed that our method can achieve an average speedup of 15× over the existing related work, with a relative error no more than 1.5 %.

To the best of our knowledge, this is the first work to present an analytical solution of energy calculation for a periodic schedule on multi-core platforms. It is also important to point out that our proposed energy calculation method is rather general and fundamental, and thus can be applied for different architectures (i.e., homogeneous and heterogeneous multi-core platforms) and applications.

The rest of this paper is organized as follows. We first discuss the related work in Sect. 2, and then introduce the system models used in this paper in Sect. 3. We introduce our temperature calculation method in Sect. 4. Our analytical solution of energy calculation is presented in Sect. 5. We show our experimental results in Sect. 6 and conclude this paper in Sect. 7.

2 Related work

Energy estimation or calculation is a fundamental problem in energy-aware design and analysis. Earlier research, e.g., [14,25], has been exclusively focused on dynamic energy consumption. Some later research such as that in [13] takes the leakage power into consideration, but assumes that leakage power is constant. Under this assumption, the calculation of energy consumption for a given voltage schedule is trivial, since the overall power consumption remains the same as long as a system keeps the same running voltage and frequency. However, when considering the leakage/temperature dependency, the problem substantially becomes more challenging since the leakage power consumption (and thus the overall power consumption) varies with the temperature, and temperature changes with the power consumption as well. The energy calculation problem becomes even more complicated for multi-core platforms when the leakage power of one core depends not only on its own temperature, but also on temperatures from other cores as well. As a result, many existing researches on thermal and energy management (e.g., [22]) do not explicitly formulate energy consumption.

To calculate the overall energy consumption accurately, particularly for today's multi-core platforms, we need to take leakage/temperature dependency into consideration. A great number of studies have been published on solving the energy-aware multi-core design with consideration of leakage/power dependency [6,8,9,11,17,24,28]. However, the fundamental problem of how to effectively and efficiently calculate the energy consumption is still open. One intuitive and commonly adopted approach is to use the numerical method. According to this method, the entire voltage schedule is split into a set of small time intervals, such that within each interval the voltage/frequency and temperature of all cores can be regarded as invariant. The

temperature and power trace, and thus the energy consumption, for a schedule can be obtained accordingly. For example, Liu et al. [16] formulated the energy minimization under a peak temperature as a non-linear programming problem, and then employed the above-mentioned method to calculate the energy consumption. Bao et al. [2] also used a similar approach to keep track of temperature variations and proposed an energy minimization method by dynamically selecting the supply voltage. One major problem with this approach is that the accuracy significantly depends on the variation rate of power and temperature. To achieve high accuracy, the length of the interval needs to be kept very small and thus the computation cost can be very high. Huang et al. [11] proposed a different approach to calculate the energy consumption. Based on leakage/temperature dependency model proposed in [20], they developed an analytical closed-form energy estimation method for a voltage schedule. However, their work can only be applied for single-core platforms but not for multi-core platforms, since there was no consideration of heat transfer among different cores in their model. Recently, Fan et al. [7] developed a closed-form solution for energy calculation on multi-core platforms, which, however, only applied for a single scheduling period. In this paper, we study the problem of energy calculation on multi-core platforms for a periodic schedule with the dependency between leakage and temperature taken into consideration. In the next section, we will introduce some background and preliminary concepts closely related to this work.

3 Preliminary

3.1 Processing core and task model

The real-time system considered in this paper consists of M cores, denoted as $\mathbb{P} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_M\}$. Each core has N *running modes*, each of which is characterized by a pair of parameters (v_k, f_k), where v_k and f_k are the supply voltage and working frequency under mode k, respectively.

Let $\mathcal{S}$ represent a *voltage schedule* or *speed schedule* which indicates how the supply voltage and working frequency are varied for each core at different times. We assume $\mathcal{S}$ is known. For example, $\mathcal{S}$ can be a design alternative during the design space exploration process, or an energy-efficient solution based on a certain heuristic. In this paper, we use *voltage schedule* and *speed schedule* interchangeably.

Next, we define the concept of *state interval* as below:

Definition 1 Given a voltage schedule $\mathcal{S}$ for a multi-core system, an interval $[t_{q-1}, t_q]$ is called a *state interval* if each core runs only at one mode during that interval.

According to Definition 1, a voltage schedule $\mathcal{S}$ essentially consists of a number of non-overlapped state intervals, i.e., Q state intervals. Assume the length of one schedule period of $\mathcal{S}$ to be L, then we have that

1. $\bigcup_{q=1}^{Q}[t_{q-1}, t_q] = [0, L]$.
2. $[t_{q-1}, t_q] \bigcap [t_{p-1}, t_p] = \emptyset$, if $q \neq p$.

In addition, for a single state interval $[t_{q-1}, t_q]$, we use κ_q to denote the interval mode, which consists of the running modes of all cores in that interval, i.e., $\kappa_q = \{k_1, \ldots, k_M\}$ where k_i is the running mode of core $\mathcal{P}_i$ in that interval.

3.2 Power model

The overall power consumption (in $Watt$) is composed of dynamic power P_{dyn} and leakage power P_{leak}. In our power model, P_{dyn} is independent of the temperature, while P_{leak} is sensitive to both temperature and supply voltage. The dynamic power consumption is proportional to the square of supply voltage and linearity of working frequency [21]. In this paper, we assume that the working frequency is linearly proportional to supply voltage; thus, the dynamic power consumption of core $\mathcal{P}_i$ can be formulated as [11,19]

$$P_{\text{dyn},i} = \gamma_{k_i} \cdot v_{k_i}^3, \tag{1}$$

where v_{k_i} is the supply voltage of core $\mathcal{P}_i$ and γ_{k_i} is a constant, both of which depend on the running mode of core $\mathcal{P}_i$, i.e., mode k_i.

While the circuit-level study reveals a very complicated relation between leakage power and temperature, Liu et al. [27] found that a linear approximation of the leakage temperature dependency is fairly accurate. As such, similar to the work in [19], we approximate the leakage power of core $\mathcal{P}_i$ as follows:

$$P_{\text{leak},i} = \left(\alpha_{k_i} + \beta_{k_i} \cdot T_i(t)\right) \cdot v_{k_i}, \tag{2}$$

where α_{k_i} and β_{k_i} are constants depending on the core running mode, i.e., mode k_i.

Consequently, the total power consumption of core $\mathcal{P}_i$ at time t, denoted as $P_i(t)$, can be formulated as:

$$P_i(t) = \left(\alpha_{k_i} + \beta_{k_i} \cdot T_i(t)\right) \cdot v_{k_i} + \gamma_{k_i} \cdot v_{k_i}^3. \tag{3}$$

For convenience in our presentation, we rewrite the above formula by separating the elements into temperature independent/dependent parts, such that

$$P_i(t) = \psi_i + \phi_i \cdot T_i(t), \tag{4}$$

where

$$\psi_i = \alpha_{k_i} \cdot v_{k_i} + \gamma_{k_i} \cdot v_i^3 \tag{5}$$

$$\phi_i = \beta_{k_i} \cdot v_{k_i}. \tag{6}$$

As such, the power consumption for a multi-core system can be represented as

$$\begin{bmatrix} P_1(t) \\ \vdots \\ P_M(t) \end{bmatrix} = \begin{bmatrix} \psi_1 \\ \vdots \\ \psi_M \end{bmatrix} + \begin{bmatrix} \phi_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \phi_M \end{bmatrix} \begin{bmatrix} T_1(t) \\ \vdots \\ T_M(t) \end{bmatrix} \tag{7}$$

or

$$\mathbf{P}(t) = \mathbf{\Psi} + \mathbf{\Phi}\mathbf{T}(t). \tag{8}$$

In our paper, we use the bold text for a vector/matrix and the unbolded text for a value, e.g., $\mathbf{T}$ represents a temperature vector, while T represents a temperature value.

3.3 Thermal model

The thermal model used in this paper is similar to the one used in related research [22,23]. Figure 1 illustrates the thermal model for a 4-core system. C_i and R_{ij} denote the thermal capacitance (in Watt/°C) of core $\mathcal{P}_i$ and the thermal resistance (in $J/^oC$) between core $\mathcal{P}_i$ and $\mathcal{P}_j$, respectively. Let T_{amb} denote the ambient temperature; then in general, the thermal phenomena of core $\mathcal{P}_i$ can be formulated as:

$$C_i \cdot \frac{\mathrm{d}T_i(t)}{\mathrm{d}t} + \frac{T_i(t) - T_{\text{amb}}}{R_{ii}} + \sum_{j \neq i} \frac{T_i(t) - T_j(t)}{R_{ij}} = P_i(t). \tag{9}$$

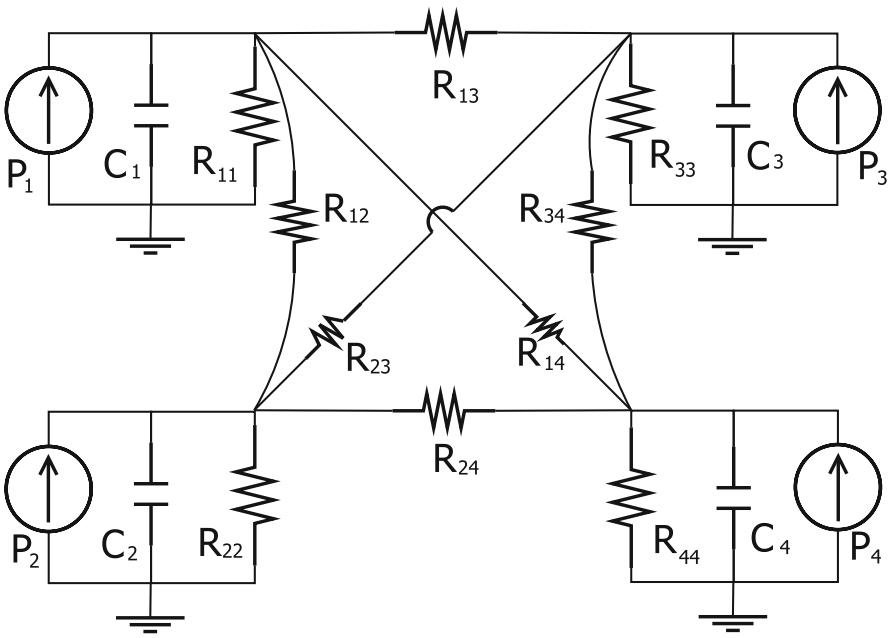


Fig. 1 Illustration for thermal phenomena on multi-core system

Let $\delta_i = \frac{T_{\text{amb}}}{R_{ii}}$ and

$$g_{ij} = \begin{cases} \sum_{j=1}^{M} \frac{1}{R_{ij}}, & \text{if } j = i \\ \frac{-1}{R_{ij}}, & \text{otherwise} \end{cases}. \tag{10}$$

Then the thermal model in Eq. (9) can be rewritten as

$$C_i \cdot \frac{\mathrm{d}T_i(t)}{\mathrm{d}t} + \sum_{j=1}^{M} g_{ij} \cdot T_j(t) = P_i(t) + \delta_i. \tag{11}$$

Accordingly, for the entire system, the thermal model can be represented as

$$\mathbf{C}\frac{d\mathbf{T}(t)}{\mathrm{d}t} + \mathbf{g}\mathbf{T}(t) = \mathbf{P}(t) + \boldsymbol{\delta}, \tag{12}$$

where $\mathbf{C}$ and $\mathbf{g}$ are MxM matrices

$$\mathbf{C} = \begin{bmatrix} C_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & C_M \end{bmatrix}, \quad \mathbf{g} = \begin{bmatrix} g_{11} & \cdots & g_{1M} \\ \vdots & \ddots & \vdots \\ g_{M1} & \cdots & g_{MM} \end{bmatrix} \tag{13}$$

and $\boldsymbol{\delta}$ is an $Mx1$ vector.

$$\boldsymbol{\delta} = \begin{bmatrix} \delta_1 \\ \vdots \\ \delta_M \end{bmatrix}. \tag{14}$$

Note that $\mathbf{C}$, $\mathbf{g}$ and $\boldsymbol{\delta}$ are all constants that only depend on the multi-core architecture, i.e., capacitance and/or conductance. It is worth mentioning that our thermal model is very general and accounts for the heat transfer impacts among different cores. It can be used for thermal analysis for both the temperature-transient states as well as the temperature-stable state.

4 Temperature calculation on multi-core platforms

As leakage power is dependent on temperature, to calculate the energy consumption, it is necessary to effectively calculate the temperature first. In this section, we first present how to formulate the temperature in thermal-transient state for a constant-voltage schedule interval and then present our proposed analytical solution to calculate the temperature in a thermal steady state for a periodic voltage schedule interval.

4.1 Temperature calculation for thermal-transient state

Note that, by applying the power model [see Eq. (8)] into the thermal model [see Eq. (12)], we can directly obtain that

$$\mathbf{C}\frac{d\mathbf{T}(t)}{\mathrm{d}t} + \mathbf{g}\mathbf{T}(t) = \mathbf{\Psi} + \mathbf{\Phi}\mathbf{T}(t) + \boldsymbol{\delta}. \tag{15}$$

Let $\mathbf{G} = \mathbf{g} - \mathbf{\Phi}$, then the above equation can be rewritten as:

$$\mathbf{C}\frac{d\mathbf{T}(t)}{\mathrm{d}t} + \mathbf{G}\mathbf{T}(t) = \mathbf{\Psi} + \boldsymbol{\delta}. \tag{16}$$

Since $\mathbf{C}$ is the capacitance matrix with no zero values only on the diagonal, we know $\mathbf{C}$ is nonsingular. Thus, the inverse of $\mathbf{C}$, i.e., $\mathbf{C}^{-1}$ exists. Then Eq. (16) can be further represented as:

$$\frac{d\mathbf{T}(t)}{\mathrm{d}t} = \mathbf{A}\mathbf{T}(t) + \mathbf{B}, \tag{17}$$

where $\mathbf{A} = -\mathbf{C}^{-1}\mathbf{G}$ and $\mathbf{B} = \mathbf{C}^{-1}(\mathbf{\Psi} + \boldsymbol{\delta})$. The system thermal model shown in Eq. (17) has a form of first-order *ordinary differential equations* (ODE), which has the following solution under constant coefficients:

$$\mathbf{T}(t) = e^{t\mathbf{A}}\mathbf{T}_0 + \mathbf{A}^{-1}(e^{t\mathbf{A}} - \mathbf{I})\mathbf{B} \tag{18}$$

where $\mathbf{T}_0$ is the initial temperature.

Specifically, for a state interval $[t_{q-1}, t_q]$, with κ_q the corresponding interval mode, once the temperatures at the starting point, i.e., $\mathbf{T}(t_{q-1})$, are given, according to Eq. (18), the ending temperatures of that interval, i.e., $\mathbf{T}(t_{q-1})$, can be directly formulated as:

$$\mathbf{T}(t_q) = e^{\Delta t_q \mathbf{A}_{\kappa_q}}\mathbf{T}(t_{q-1}) + \mathbf{A}_{\kappa_q}^{-1}(e^{\Delta t_q \mathbf{A}_{\kappa_q}} - \mathbf{I})\mathbf{B}_{\kappa_q}, \tag{19}$$

where $\mathbf{A}_{\kappa_q} = -\mathbf{C}^{-1}\mathbf{G}_{\kappa_q}$, $\mathbf{B}_{\kappa_q} = \mathbf{C}^{-1}(\mathbf{\Psi}_{\kappa_q} + \boldsymbol{\delta})$, and $\Delta t_q = t_q - t_{q-1}$. Note that since $\mathbf{A}_{\kappa_q}$ and $\mathbf{B}_{\kappa_q}$ are only dependent on the core running modes, i.e., κ_q, within a state interval $[t_{q-1}, t_q]$, both $\mathbf{A}_{\kappa_q}$ and $\mathbf{B}_{\kappa_q}$ are constant.

4.2 Temperature calculation for thermal steady state

Consider a periodic voltage schedule $\mathcal{S}$ and the corresponding initial temperature $\mathbf{T}(0)$. For an arbitrary state interval $[t_{q-1}, t_q]$, to obtain its steady-state temperature, one intuitive way is to trace the entire schedule $\mathcal{S}$ by successively calculating the temperature from the first scheduling period until the system reaches its steady state. However, when the time that the system needed to achieve its steady state is too long, the computational cost can be extremely expensive. In what follows, we present a

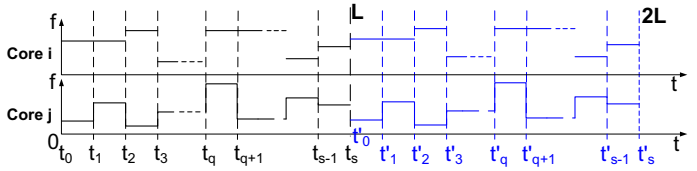


Fig. 2 A speed schedule within two scheduling periods

closed-form solution that can rapidly calculate steady-state temperatures for a periodic voltage schedule.

Let us first consider the temperature variation at the end of each scheduling period, i.e., $t = nL$, where $n \geq 1$. Let the scheduling points of $\mathcal{S}(t)$ in the first period be $t_0, t_1, \ldots, t_s$, respectively. After repeating $\mathcal{S}(t)$, let the corresponding points in the second scheduling period be $t'_0, t'_1, \ldots, t'_s$, respectively (see Fig. 2). Note that $t_0 = 0$, $t'_0 = t_s = L$ and $t'_s = 2L$. According to Eq. (19), at time t_1 and t'_1, we have

$$\mathbf{T}(t_1) = e^{\mathbf{A}_{\kappa_1}\Delta t_1}\mathbf{T}(t_0) + \mathbf{A}_{\kappa_1}^{-1}(e^{\mathbf{A}_{\kappa_1}\Delta t_1} - \mathbf{I})\mathbf{B}_{\kappa_1} \tag{20}$$

$$\mathbf{T}(t'_1) = e^{\mathbf{A}_{\kappa_1}\Delta t'_1}\mathbf{T}(t'_0) + \mathbf{A}_{\kappa_1}^{-1}(e^{\mathbf{A}_{\kappa_1}\Delta t'_1} - \mathbf{I})\mathbf{B}_{\kappa_1}. \tag{21}$$

Subtract Eq. (20) from (21) on both sides, and simplify the result by applying $\Delta t'_1 = \Delta t_1$, $t_0 = 0$ and $t'_0 = L$, we get

$$\mathbf{T}(t'_1) - \mathbf{T}(t_1) = e^{\mathbf{A}_{\kappa_1}\Delta t_1}(\mathbf{T}(L) - \mathbf{T}(0)).$$

Following the same trace of the above derivation, we have that

$$\begin{aligned} \mathbf{T}(t'_2) - \mathbf{T}(t_2) &= e^{\mathbf{A}_{\kappa_2}\Delta t_2}e^{\mathbf{A}_{\kappa_1}\Delta t_1}(\mathbf{T}(L) - \mathbf{T}(0)) \\ &\cdots \\ \mathbf{T}(t'_s) - \mathbf{T}(t_s) &= e^{\mathbf{A}_{\kappa_s}\Delta t_s} \ldots e^{\mathbf{A}_{\kappa_1}\Delta t_1}(\mathbf{T}(L) - \mathbf{T}(0)). \end{aligned} \tag{22}$$

Since $t_s = L$, $t'_s = 2L$, and $e^{\mathbf{A}_{\kappa_s}\Delta t_s} \ldots e^{\mathbf{A}_{\kappa_1}\Delta t_1} = K$, Eq. (22) can be rewritten as

$$\mathbf{T}(2L) - \mathbf{T}(L) = \mathbf{K}(\mathbf{T}(L) - \mathbf{T}(0)). \tag{23}$$

In the same way, we can see that

$$\begin{aligned} \mathbf{T}(3L) - \mathbf{T}(2L) &= \mathbf{K}(\mathbf{T}(2L) - \mathbf{T}(L)) \\ \mathbf{T}(4L) - \mathbf{T}(3L) &= \mathbf{K}(\mathbf{T}(3L) - \mathbf{T}(2L)) \\ &\cdots \\ \mathbf{T}(nL) - \mathbf{T}((n-1)L) &= \mathbf{K}(\mathbf{T}((n-1)L) - \mathbf{T}((n-2)L)). \end{aligned}$$

Thus, we can construct that

$$\mathbf{T}(xL) - \mathbf{T}((x-1)L) = \mathbf{K}^{x-1}(\mathbf{T}(L) - \mathbf{T}(0)), \tag{24}$$

where $x = 1, 2, \ldots, n$. Summing up the above n equations, we get

$$\mathbf{T}(nL) = \mathbf{T}(0) + \left(\sum_{x=1}^{n} \mathbf{K}^{x-1}\right)(\mathbf{T}(L) - \mathbf{T}(0)). \tag{25}$$

In the above, $\{\mathbf{K}^{x-1} | x = 1, 2, \ldots, n\}$ forms a matrix geometric sequence. If $(\mathbf{I} - \mathbf{K})$ is invertible, then we have

$$\mathbf{T}(nL) = \mathbf{T}(0) + (\mathbf{I} - \mathbf{K})^{-1}(\mathbf{I} - \mathbf{K}^n)(\mathbf{T}(L) - \mathbf{T}(0)). \tag{26}$$

Now, we consider the temperature variation at an arbitrary time instant when repeating a periodic schedule. Given a periodic voltage schedule $\mathcal{S}(t)$, for any time instant $t = t_q$, where $t_q \in [0, L]$, repeat $\mathcal{S}(t)$ for n times, where $n \geq 1$. Let $\mathbf{T}(nL + t_q)$ represent the corresponding temperature of $\mathbf{T}(t_q)$ in the nth scheduling period, then by following in a similar way to the above derivation, we can get that

$$\mathbf{T}(nL + t_q) = \mathbf{T}(t_q) + \mathbf{K}_q(\mathbf{I} - \mathbf{K})^{-1}(\mathbf{I} - \mathbf{K}^n)(\mathbf{T}(L) - \mathbf{T}(0)), \tag{27}$$

where $\mathbf{K}_q = e^{\mathbf{A}_{\kappa_q}\Delta t_q} \cdot e^{\mathbf{A}_{\kappa_{q-1}}\Delta t_{q-1}} \ldots e^{\mathbf{A}_{\kappa_1}\Delta t_1}$, $q = 1, 2, \ldots, s$.

Now, we are ready to formulate the temperature variation in the system steady state. Consider an arbitrary time instant within the first scheduling period, i.e., $t = t_q$ where $0 \leq t_q \leq L$. The basic idea to get the steady-state temperature corresponding to t_q is to let n go to infinity in Eq. (27). We formally formulate our method in Theorem 1 as below.

Theorem 1 *Given a periodic voltage schedule $\mathcal{S}(t)$, let $\boldsymbol{T}(L)$ and $\boldsymbol{T}(t_q)$ be the temperatures at time instant L and t_q, $t_q \in [0, L]$, respectively. If for each eigenvalue λ_i of $\boldsymbol{K}$, we have $|\lambda_i| < 1$, then the steady-state temperature corresponding to t_q can be formulated as*

$$\boldsymbol{T}_{ss}(t_q) = \boldsymbol{T}(t_q) + \boldsymbol{K}_q(\boldsymbol{I} - \boldsymbol{K})^{-1}(\boldsymbol{T}(L) - \boldsymbol{T}(0)). \tag{28}$$

Proof First, based on Eq. (27), by letting $n \mapsto \infty$, the steady-state temperature of the q^{th} scheduling point in $\mathcal{S}(t)$ can be represented as

$$\mathbf{T}_{ss}(t_q) = \mathbf{T}(t_q) + \mathbf{K}_q(\mathbf{I} - \mathbf{K})^{-1}\left(\mathbf{I} - \lim_{n\mapsto\infty} \mathbf{K}^n\right)(\mathbf{T}(L) - \mathbf{T}(0)). \tag{29}$$

When $n \mapsto \infty$, the matrix sequence $\mathbf{K}^n$ converges if and only if $|\lambda_i| < 1$, for each eigenvalue λ_i of $\mathbf{K}$ [4]. Under this condition, we have $\lim_{n\to\infty} \mathbf{K}^n = 0$. Moreover, if $\forall\lambda_i, |\lambda_i| < 1$ holds, $(\mathbf{I} - \mathbf{K})$ is invertible. Thus, the steady-state temperature of the q^{th} scheduling point in $\mathcal{S}(t)$ can be further formulated as

$$\mathbf{T}_{ss}(t_q) = \mathbf{T}(t_q) + \mathbf{K}_q(\mathbf{I} - \mathbf{K})^{-1}(\mathbf{T}(L) - \mathbf{T}(0))\mathbf{T}(0)). \tag{30}$$

□

It is important to point that as $n \mapsto \infty$, unless the temperature increases and causes the system to break down, the system will eventually achieve its steady state. Thus, the condition of $|\lambda_i| < 1$, for each eigenvalue λ_i of $\mathbf{K}$, should always hold once the system achieves its thermal steady state. Therefore, it is reasonable and practical to make the assumption of the condition given by Theorem 1.

5 Energy calculation on multi-core platforms

With the temperature formulation introduced as above, we are now ready to discuss our method to formulate the energy consumption on multi-core systems considering the interdependence of leakage power and temperature. In what follows, we first present an analytical solution to calculate the energy consumption for one state interval. Then, we formulate the total energy consumption for the entire voltage schedule.

5.1 Energy calculation for one state interval

Consider a state interval, i.e., $[t_{q-1}, t_q]$ with initial temperature of $\mathbf{T}(t_{q-1})$. The energy consumption of all cores within that interval can be simply formulated as

$$\mathbf{E}(t_{q-1}, t_q) = \int_{t_{q-1}}^{t_q} \mathbf{P}(t)\mathrm{d}t. \tag{31}$$

Based on our system power model, given by Eq. (8), we have

$$\mathbf{E}(t_{q-1}, t_q) = \Delta t_q \boldsymbol{\Psi} + \boldsymbol{\Phi} \int_{t_{q-1}}^{t_q} \mathbf{T}(t)\mathrm{d}t. \tag{32}$$

Given a multi-core platform, for any state interval, according to Eqs. (5) and (8), $\boldsymbol{\Psi}$ is a constant. Therefore, to calculate $\mathbf{E}(t_{q-1}, t_q)$, we only need to get $\int_{t_{q-1}}^{t_q} \mathbf{T}(t)\mathrm{d}t$.

Recall that the analytical solution for $\mathbf{T}(t)$ is given by Eq. (18). One intuitive approach is therefore to find $\int_{t_{q-1}}^{t_q} \mathbf{T}(t)\mathrm{d}t$ as follows:

$$\int_{t_{q-1}}^{t_q} \mathbf{T}(t)\mathrm{d}t = \int_{t_{q-1}}^{t_q} \left(e^{t\mathbf{A}}\mathbf{T}(t_{q-1}) + \mathbf{A}^{-1}(e^{t\mathbf{A}} - \mathbf{I})\mathbf{B}\right)\mathrm{d}t \tag{33}$$

$$\int_{t_{q-1}}^{t_q} \mathbf{T}(t)\mathrm{d}t = \int_{t_{q-1}}^{t_q} e^{t\mathbf{A}}\mathrm{d}t\mathbf{T}(t_{q-1}) + \mathbf{A}^{-1}\left(\int_{t_{q-1}}^{t_q} e^{t\mathbf{A}}\mathrm{d}t - t\mathbf{I}\right)\mathbf{B}. \tag{34}$$

The problem of this approach is that we need to find $\int_{t_{q-1}}^{t_q} e^{t\mathbf{A}}\mathrm{d}t$, but unfortunately we are not aware of any existing method or mathematical tools that can be used to solve the problem of exponential matrix integration. Therefore, to replace $\mathbf{T}(t)$ in Eq. (32) with Eq. (18) does not seem to be a promising approach.

In what follows, we present our approach to calculate the energy consumption for any state interval on a multi-core platform. We formally conclude our energy calculation method for an arbitrary state interval in Theorem 2.

Theorem 2 *Given a state interval* $[t_{q-1}, t_q]$*, let* $\boldsymbol{T}_{q-1}$ *be the temperature at time* t_{q-1}*. Then the overall system energy consumption within interval* $[t_{q-1}, t_q]$ *can be formulated as*

$$\boldsymbol{E}(t_{q-1}, t_q) = \Delta t_q \boldsymbol{\Psi} + \boldsymbol{\Phi} \boldsymbol{G}^{-1} \boldsymbol{H}, \tag{35}$$

where $\Delta t = t_q - t_{q-1}$ *and* $\boldsymbol{H} = \Delta t_q(\boldsymbol{\Psi} + \boldsymbol{\delta}) - \boldsymbol{C}\big(\boldsymbol{T}(t_q) - \boldsymbol{T}(t_{q-1})\big)$.

Proof We start our proof from Eq. (32). To make the presentation clear, we rewrite Eq. (32) as follows:

$$\mathbf{E}(t_{q-1}, t_q) = \Delta t_q \boldsymbol{\Psi} + \boldsymbol{\Phi} \int_{t_{q-1}}^{t_q} \mathbf{T}(t) \mathrm{d}t$$

Note that as long as we can get $\int_{t_{q-1}}^{t_q} \mathbf{T}(t)\mathrm{d}t$, we would find the solution for the overall energy consumption within state interval $[t_{q-1}, t_q]$. Let $\mathbf{X} = \int_{t_{q-1}}^{t_q} \mathbf{T}(t)\mathrm{d}t$, then the above equation can be rewritten as

$$\mathbf{E}(t_{q-1}, t_q) = \Delta t_q \boldsymbol{\Psi} + \boldsymbol{\Phi} \mathbf{X}. \tag{36}$$

Recall that the system thermal model can be formulated as [see Eq. (16)]:

$$\mathbf{C} \frac{d\mathbf{T}(t)}{\mathrm{d}t} + \mathbf{G}\mathbf{T}(t) = \boldsymbol{\Psi} + \boldsymbol{\delta}.$$

Since $\mathbf{C}$, $\mathbf{G}$, $\boldsymbol{\Psi}$ and $\boldsymbol{\delta}$ are all constants within interval $[t_{q-1}, t_q]$, by integrating on both sides of the above equation with respect to time t, where $t \in [t_{q-1}, t_q]$, we can get

$$\mathbf{C}\big(\mathbf{T}(t_q) - \mathbf{T}(t_{q-1})\big) + \mathbf{G} \int_{t_{q-1}}^{t_q} \mathbf{T}(t)\mathrm{d}t = \Delta t_q(\boldsymbol{\Psi} + \boldsymbol{\delta}), \tag{37}$$

where $\Delta t_q = t_q - t_{q-1}$. Then we replace $\int_{t_{q-1}}^{t_q} \mathbf{T}(t)\mathrm{d}t$ with $\mathbf{X}$ in the above and derive that

$$\mathbf{C}\Delta\mathbf{T}_q + \mathbf{G}\mathbf{X} = \Delta t_q(\boldsymbol{\Psi} + \boldsymbol{\delta}). \tag{38}$$

Let $\mathbf{H} = \Delta t_q(\boldsymbol{\Psi} + \boldsymbol{\delta}) - \mathbf{C}\big(\mathbf{T}(t_q) - \mathbf{T}(t_{q-1})\big)$. Note that once $\mathbf{T}(t_{q-1})$ is known, $\mathbf{T}(t_q)$ can be directly calculated according to Eq. (19). Consequently, $\mathbf{H}$ can be determined. By simplifying Eq. (38) with $\mathbf{H}$, we can get that

$$\mathbf{G}\mathbf{X} = \mathbf{H}. \tag{39}$$

Assuming $\mathbf{G}$ is nonsingular (in fact, $\mathbf{G}$ is always a nonsingular matrix in practice), $\mathbf{X}$ can thus be solved as

$$\mathbf{X} = \mathbf{G}^{-1}\mathbf{H}. \tag{40}$$

Finally, by applying Eq. (40) into (36), we can get that

$$\mathbf{E}(t_{q-1}, t_q) = \Delta t_q \boldsymbol{\Psi} + \boldsymbol{\Phi}\mathbf{G}^{-1}\mathbf{H}. \tag{41}$$

□

From Theorem 2, we can see that for any state interval $[t_{q-1}, t_q]$, once the temperature at the beginning of that interval, i.e., $\mathbf{T}(t_{q-1})$, is known, the total energy consumption within $[t_{q-1}, t_q]$ can be directly calculated.

As such, given a periodic voltage schedule $\mathcal{S}$ and an initial temperature $\mathbf{T}_0$, we are able to calculate the energy consumption within any state interval in any scheduling period. We conclude our method in Corollary 1.

Corollary 1 *Given a periodic voltage schedule $\mathcal{S}(t)$ consisting of Q state intervals, let $\boldsymbol{T}_0$ be the initial temperature. Then the energy consumption within the qth $(1 \le q \le Q)$ state interval in the nth $(n \ge 1)$ scheduling period, denoted as $\boldsymbol{E}(t_{q-1}+nL, t_q+nL)$, can be calculated as*

$$\boldsymbol{E}(t_{q-1}+nL, t_q+nL) = \Delta t_q \boldsymbol{\Psi}_{\kappa_q} + \boldsymbol{\Phi}_{\kappa_q}\boldsymbol{G}_{\kappa_q}^{-1}\boldsymbol{H}_{\kappa_q}, \tag{42}$$

where $\Delta t = t_q - t_{q-1}$ and $\boldsymbol{H}_{\kappa_q} = \Delta t_q(\boldsymbol{\Psi}_{\kappa_q} + \boldsymbol{\delta}) - \boldsymbol{C}\big(\boldsymbol{T}(t_q+nL) - \boldsymbol{T}(t_{q-1}+nL)\big)$.

Corollary 1 can be directly derived from Theorem 2. With the help of Corollary 1, given any periodic voltage schedule on a multi-core platform, we can easily calculate the energy consumption within any state interval when repeating that schedule.

Correspondingly, given a periodic voltage schedule, we can calculate the energy consumption within any interval in system thermal steady state. We formally conclude our method in Corollary 2.

Corollary 2 *Given a periodic voltage schedule $\mathcal{S}(t)$ consisting of Q state intervals, let $\mathbf{T_0}$ be the initial temperature. Then the energy consumption within the qth $(1 \le q \le Q)$ state interval in the system steady state , denoted as $\boldsymbol{E}_{ss}(t_{q-1}, t_q)$, can be calculated as*

$$\boldsymbol{E}_{ss}(t_{q-1}, t_q) = \Delta t_q \boldsymbol{\Psi}_{\kappa_q} + \boldsymbol{\Phi}_{\kappa_q}\boldsymbol{G}_{\kappa_q}^{-1}\boldsymbol{H}_{ss_{\kappa_q}}, \tag{43}$$

where $\Delta t = t_q - t_{q-1}$ and $\boldsymbol{H}_{ss_{\kappa_q}} = \Delta t_q(\boldsymbol{\Psi}_{\kappa_q} + \boldsymbol{\delta}) - \boldsymbol{C}\big(\boldsymbol{T}_{ss}(t_q) - \boldsymbol{T}_{ss}(t_{q-1})\big)$.

Corollary 2 is directly derived from Corollary 1 by replacing the transient temperatures with steady-state temperatures.

5.2 Energy calculation for one scheduling period

We further derive a method to calculate the overall energy consumption within one scheduling period for a periodic voltage schedule. Consider a periodic voltage schedule $\mathcal{S}(t)$ consisting of Q state intervals; the total system energy consumption in the nth scheduling period can be obtained by summing up the energy consumptions of all state intervals within that scheduling period. We conclude this energy calculation method in Theorem 3.

Theorem 3 *Given a periodic voltage schedule $\mathcal{S}(t)$ consisting of Q state intervals, let $\mathbf{T_0}$ be the initial temperature. Then the overall system energy consumption within the nth scheduling period, denoted as $E_n^{total}(\mathcal{S})$, can be calculated as*

$$E_n^{total}(\mathcal{S}) = \sum_{q=1}^{Q} \sum_{i=1}^{M} E_i(t_{q-1} + nL, t_q + nL), \tag{44}$$

where $E_i(t_{q-1} + nL, t_q + nL)$ is obtained according to Eq. (42).

In Theorem 3, the energy consumption of each core within the qth scheduling interval in the nth scheduling period can be obtained based on Eq. (42). Meanwhile, the corresponding temperature of the qth scheduling interval in the nth scheduling period can be calculated according to Eq. (27).

Given a periodic voltage schedule, from Theorem 44, we can further derive a method to calculate the overall energy consumption within one scheduling period in the system steady state. We conclude our approach in Corollary 3.

Corollary 3 *Given a periodic voltage schedule $\mathcal{S}$ consisting of Q state intervals, let $\mathbf{T_0}$ be the initial temperature. Then the total system energy consumption within one scheduling period in the system steady state, denoted as $E_{ss}^{total}(\mathcal{S})$, can be calculated as*

$$E_{ss}^{total}(\mathcal{S}) = \sum_{q=1}^{Q} \sum_{i=1}^{M} E_{ss_i}(t_{q-1}, t_q), \tag{45}$$

where $E_{ss_i}(t_{q-1}, t_q)$ is obtained according to Eq. (43).

Corollary 3 calculates the steady-state energy consumption within one schedule period by applying our proposed steady-state temperature formulation given in Eq. (28) into the steady-state energy calculation formula given in Eq. (44).

The computational complexity of our proposed energy calculation approach for one scheduling period, either in system steady state or not, mainly comes from the matrix multiplications and inversions. The energy calculation for each state interval has a complexity of $O(M^3)$. Thus, the energy calculation for one schedule period with Q state intervals has a complexity of $O(Q \times M^3)$. In what follows, we use experiments to evaluate the performance of our proposed method.

6 Experimental evaluation

In this section, we validated the proposed energy calculation method with simulations. We compared our proposed method with the traditional *numerical method* to obtain some insights into the effectiveness and efficiency of an energy estimation approach. In what follows, we first introduce the settings for our experiments. We then present and discuss the experimental results.

6.1 Experimental setup

We performed our experimental simulations based on a 3×3 multi-core system. The granularity of the floorplan was restricted to core level. Our core model was based on $65nm$ technology as presented in [15]. We assumed that each core supported 3 active modes with supply voltage ranging from 0.8 to 1.0 V and a step size of 0.1 V. We also set one inactive/sleep mode with supply voltage equal to 0 V.

Specifically, we adopted the same platform parameters as used in work [19] (see Table 1). The thermal parameters, including convection resistance, convection capacitance, etc., were taken from HotSpot-4.02 [10]. The thermal nodes in our thermal model included active layer, interface layer, heat spreader and heat sink. We set the peak temperature constraint as 110 °C, and set the ambient temperature T_{amb} as well as the initial temperature T_0 as 30°C. In our experiment, we chose three active modes (voltage = [0,0.8, 0.9,1.0]) and one sleep mode (voltage = 0); the corresponding parameters can be found in Table 2.

We randomly generated 50 multi-core speed schedules as our test cases. The running mode for each scheduling interval was randomly chosen from [0,0.8, 0.9,1.0]V. The

Table 1 HotSpot parameters and floorplan

Parameter	Value
Total cores	9 (3×3)
Area per core	4 mm^2
Die thickness	0.15 mm
Heat spreader Side	20 mm
Heat sink side	30 mm
Convection resistance	0.1 K/W
Convection capacitance	140 J/K
Ambient temperature	30 °C

Table 2 Power/thermal parameters

$V_{\text{dd}}(V)$	α	β	γ
0.0	0.0	0.0	0.0
0.8	1.4533	0.0760	6.0531
0.9	2.4173	0.0844	5.8008
1.0	4.0533	0.0936	5.8906

total length of the scheduling interval was evenly distributed within [100,200], and the length of each scheduling interval was evenly distributed within [30,50]. For each test case, our proposed method as well as the traditional numerical method with sampling interval lengths varying from 0.5 to 3.0 s was used to calculate the energy consumption. The baseline was obtained by setting the length of the sampling interval to 0.01. When applying the numerical method, we calculated the leakage power consumption based on the accurate circuit level leakage temperature model [15], i.e.,

$$I_{\text{leak}} = I_s \cdot (\mathcal{A} \cdot T^2 \cdot e^{((a \cdot V_{\text{dd}} + b)/T)} + \mathcal{B} \cdot e^{(c \cdot V_{dd} + d)}), \tag{46}$$

where I_s is the leakage current at a certain reference temperature and supply voltage, T is the core temperature, and $\mathcal{A}, \mathcal{B}, a, b, c, d$ are physically determined constants (i.e., fitting parameters). All simulations were conducted on a *Dell Precision T1500 Desktop Workstation* with CPU type of *Intel i5 750 Quad Core* and $4GB$ memory capacity.

6.2 Accuracy analysis of our energy calculation method

We first investigated the performance of our analytical energy calculation method from the perspective of accuracy.

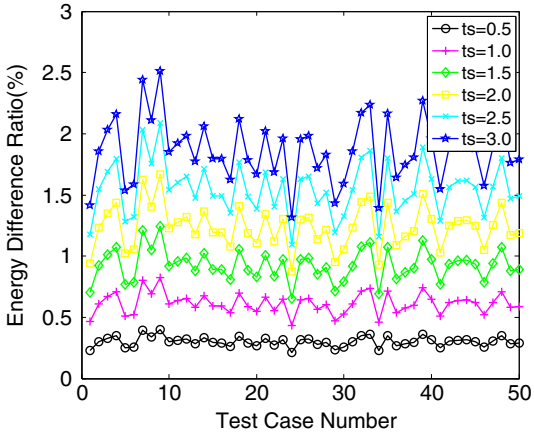
6.2.1 Accuracy analysis

We first evaluated the accuracy of our proposed energy calculation method. Note that, while we analytically developed the energy formulation as shown in Eqs. (42) and (44), its accuracy is contingent upon our leakage/temperature dependency assumption as listed in Eq. (2).

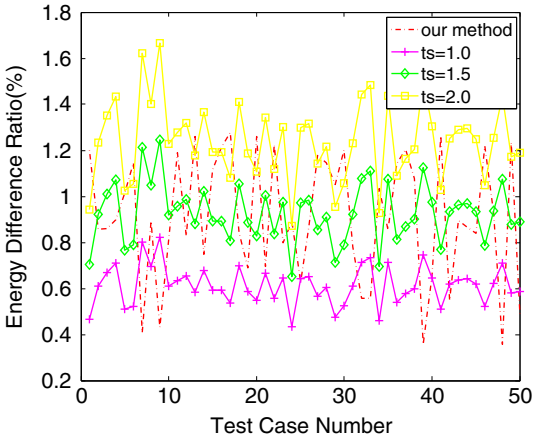
To compare the accuracy of different energy estimation approaches, we need to identify the accurate energy consumption for a given speed schedule. We resorted to the numerical method with a very short sampling interval to achieve this goal. The question is how short should the sampling interval be.

In this experiment, we set the length of sampling interval t_s from 0.5 s to 3.0 s with a step length of 0.5 s and calculated the energy consumption for different schedules. Particularly, we set $t_s = 0.01$ s as the baseline, since we found that the largest relative energy difference between $t_s = 0.01$ s and $t_s = 0.5$ s was smaller than 0.4 %. We then normalized the energy consumption by other approaches to the baseline results. Figure 3a shows the relative differences of energy consumption estimation results using a numerical approach with different sampling intervals, i.e., from $t_s = 0.5$ s to $t_s = 3.0$ s. The relative differences of energy consumption based on our proposed approach and comparable numerical results are presented in Fig. 3b.

From Fig. 3a, it is not surprising to see that the smaller the sampling interval, the smaller does the energy difference ratio become. For example, when t_s is decreased from 3.0 to 0.5, the average energy difference ratio is reduced from 1.7 to 0.4 %. This is because the smaller the sampling interval, the less can the temperature change. Since the numerical method estimates the leakage consumption within an interval assuming



(a) Numerical method



(b) Our proposed method

Fig. 3 Accuracy analysis, compared with the numerical method under $t_s = 0.01$

temperature within a sampling interval does not change, the error of the estimated leakage energy can be kept small if the sampling interval is small enough.

On the other hand, we can see from Fig. 3b that our proposed method performed well from the aspect of accuracy. For example, the largest relative error observed in Fig. 3b is no more than 1.5 %. As shown in Fig. 3b, we can see that our method outperforms the numerical method with $t_s = 2.0$ s for most test cases and is compatible with the method with $t_s = 1.5$ s. The experimental results clearly show that our proposed approach can achieve very good accuracy in estimating the overall energy consumption for a given speed schedule.

6.3 Time efficiency analysis of our energy calculation method

Then we evaluated the computational efficiency of our proposed method. We collected the CPU times for different approaches for all test cases. We then used the CPU times of our method as the baseline results. The normalized results are shown in Fig. 4.

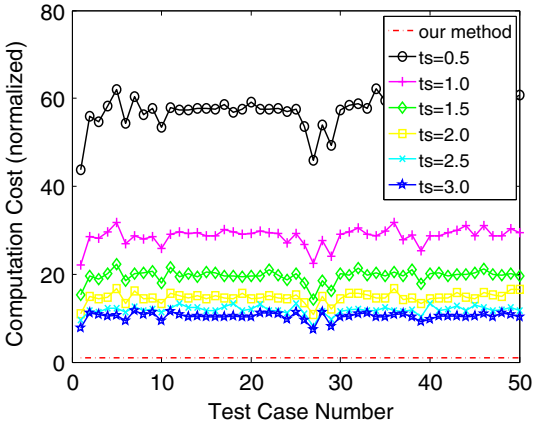


Fig. 4 Time efficiency analysis, normalized with our method

From Fig. 4, we can see that the numerical method with a small sampling interval can have a substantially large computational overhead than our approach. For example, as shown in Fig. 4, our method is more than 50 times (on average) faster than the numerical approach with $t_s = 0.5$, and 10 times (on average) faster than that with $t_s = 3.0$. Compared with the numerical method with $t_s = 1.5$, which is compatible with our method from the perspective of accuracy, our method can achieve an average speedup of 15 times. Note that the computational complexity of our approach is determined only by the number of state intervals in a speed schedule, while the complexity of the numerical approach depends on both the schedule length (L) and sampling interval (t_s). As shown in Fig. 3a, to achieve high accuracy, the sampling interval must be very small and thus very timing consuming. From Fig. 4, we can conclude that the proposed method is much more time efficient than the numerical approach.

7 Conclusions

Energy consumption optimization is a critical design issue in the design of multi-core computing systems. It becomes more challenging in deep sub-micron domains when leakage consumption becomes more and more significant and the interdependency of leakage and temperature becomes substantial. A key to solve this problem is to calculate the energy consumption efficiently and effectively.

In this paper, we proposed a closed-form solution for energy calculation for periodic scheduling on multi-core platforms. We first presented an analytical solution of temperature calculation for periodic multi-core scheduling, which can quickly obtain the temperature dynamics in the system thermal steady state. Then based on our temperature calculation method, we developed a closed-form solution of energy calculation for any scheduling period, particularly in system thermal steady state. Different from the traditional numerical approach, our proposed analytical solution of energy calculation can rapidly and accurately obtain the energy consumption for a periodic speed/voltage schedule. Our experimental results showed that the proposed method can achieve a

speedup of 15 times compared with the numerical method, with a relative error no more than 1.5 %.

By taking the interdependency between leakage and temperature into consideration, our system models (particularly the energy model) become rather general and practical, and thus our proposed technique can be easily extended for different platforms and applications. Moreover, since our proposed energy calculation method is an analytical solution, it can be widely used in most designs and analysis with energy awareness.

References

1. AMD Opteron 6300 Series Processors. http://www.amd.com/en-us/products/server/opteron/6000/6300
2. Bao M, Andrei A, Eles P, Peng Z (2008) Temperature-aware voltage selection for energy optimization. In: Design, Automation and Test in Europe (DATE), pp 1083–1086. doi:10.1109/DATE.2008.4484920
3. Bao M, Andrei A, Eles P, Peng Z (2009) On-line thermal aware dynamic voltage scaling for energy optimization with frequency/temperature dependency consideration. In: Design Automation Conference (DAC), 46th ACM/IEEE, pp 490–495
4. Bell JRS (1998) Mathematical analysis for modeling. CRC Press, Florida
5. Borkar S (2007) Thousand core chips: a technology perspective. In: Design Automation Conference (DAC), 44th ACM, ACM, New York, NY, USA, pp 746–749. doi:10.1145/1278480.1278667
6. Chantem T, Hu XS, Dick R (2009) Online work maximization under a peak temperature constraint. In: ISLPED, pp 105–110
7. Fan M, Chaturvedi V, Sha S, Quan G (2013) An analytical solution for multi-core energy calculation with consideration of leakage and temperature dependency. In: Low Power Electronics and Design (ISLPED), 2013 IEEE International Symposium on, pp 353–358. doi:10.1109/ISLPED.2013.6629322
8. Hanumaiah V, Rao R, Vrudhula S, Chatha KS (2009) Throughput optimal task allocation under thermal constraints for multi-core processors. In: Proceedings of the 46th Annual Design Automation Conference, ACM, New York, NY, USA, DAC '09, pp 776–781. doi:10.1145/1629911.1630112
9. Hanumaiah V, Vrudhula S, Chatha K (2009) Maximizing performance of thermally constrained multi-core processors by dynamic voltage and frequency control pp 310–313
10. Hotspot 4.2 temperature modeling tool. University of Virgina p. http://lava.cs.virginia.edu/HotSpot
11. Huang H, Quan G (2011) Leakage aware energy minimization for real-time systems under the maximum temperature constraint. In: Design, Automation Test in Europe(DATE), pp 1–6. doi:10.1109/DATE.2011.5763083
12. ITRS International Technology Roadmap for Semiconductors (2011 Edition). International SEMATECH, Austin, TX. http://public.itrs.net/
13. Jejurikar R, Gupta R (2005) Dynamic slack reclamation with procrastination scheduling in real-time embedded systems. In: Design Automation Conference (DAC), 42nd IEEE, pp 111–116. doi:10.1109/DAC.2005.193783
14. Lee CH, Shin K (2004) On-line dynamic voltage scaling for hard real-time systems using the edf algorithm. In: Real-Time Systems Symposium (RTSS), 25th IEEE International, pp 319–335. doi:10.1109/REAL.2004.38
15. Liao W, He L, Lepak K (2005) Temperature and supply voltage aware performance and power modeling at microarchitecture level. Computer-Aided Design Integr Circuits Syst IEEE Trans on 24(7):1042–1053. doi:10.1109/TCAD.2005.850860
16. Liu Y, Yang H, Dick R, Wang H, Shang L (2007) Thermal vs energy optimization for dvfs-enabled processors in embedded systems. In: Quality Electronic Design (ISQED), 8th International Symposium on, pp 204–209. doi:10.1109/ISQED.2007.158
17. Lung C, Ho Y, Kwai D, Chang S (2011) Thermal-aware online task allocation for 3d multi-core processor throughput optimization. Design, automation, and test in Europe (DATE). Grenoble, France, pp 1–6
18. Markoff J (2004) Intel's big shift after hitting technical wall. New York Times
19. Quan G, Chaturvedi V (2010) Feasibility analysis for temperature-constraint hard real-time periodic tasks. Ind Inf, IEEE Trans on 6(3):329–339. doi:10.1109/TII.2010.2052057

20. Quan G, Zhang Y (2009) Leakage aware feasibility analysis for temperature-constrained hard real-time periodic tasks. In: Real-Time Systems (ECRTS), 21st Euromicro Conference on, pp 207–216. doi:10.1109/ECRTS.2009.28
21. Rabaey J, Chandrakasan A, Nikolic B (2003) Digital integrated circuits: A design perspective. In: Englewood Cliffs, NJ: Prentice-Hall
22. Sharifi S, Ayoub R, Rosing T (2012) Tempomp: Integrated prediction and management of temperature in heterogeneous mpsocs. In: Design, Automation Test in Europe(DATE), pp 593–598. doi:10.1109/DATE.2012.6176542
23. Ukhov I, Bao M, Eles P, Peng Z (2012) Steady-state dynamic temperature analysis and reliability optimization for embedded multiprocessor systems. In: Design Automation Conference (DAC), 49th ACM/EDAC/IEEE, pp 197–204
24. Yang CY, Chen JJ, Thiele L, Kuo TW (2010) Energy-efficient real-time task scheduling with temperature-dependent leakage. In: DATE, pp 9–14
25. Yao F, Demers A, Shenker S (1995) A scheduling model for reduced cpu energy. In: Foundations of Computer Science(FOCS), 36th Annual Symposium on, pp 374–382. doi:10.1109/SFCS.1995.492493
26. Yeh D, Peh LS, Borkar S, Darringer J, Agarwal A, Hwu W (2008) Thousand-core chips [roundtable]. Design Test Comput IEEE 25(3):272–278. doi:10.1109/MDT.2008.85
27. Yongpan L, Huazhong Y (2010) Temperature-aware leakage estimation using piecewise linear power models. IEICE Trans on Electron 93(12):1679–1691
28. Zhang S, Chatha KS (2010) Thermal aware task sequencing on embedded processors. In: DAC, pp 585–590