

# Thermal-Aware Lifetime Reliability in Multicore Systems

Shengquan Wang  
Department of Computer and  
Information Science  
University of Michigan-Dearborn  
Dearborn, USA  
shqwang@umd.umich.edu

Jian-Jia Chen  
Computer Engineering and  
Networks Lab.  
Swiss Federal Institute of Technology  
(ETH) Zurich, Switzerland  
jchen@tik.ee.ethz.ch

**Abstract**—As the power density of modern electronic circuits increases dramatically, systems are prone to overheating. High temperatures not only raise packaging costs, degrade system performance, and increase leakage power consumption, but also reduce the system reliability. Due to many limits in single core design including the performance and the power density, the microprocessor industry has switched their attentions to multicore design to enable the scaling of performance. Thermal effects on multicore systems are still prominent issues. One typical thermal effect is the thermal-aware lifetime reliability, which has become a serious concern. In this paper, we address the issue on how to maximize the lifetime of multicore systems while maintaining a given aggregate processor speed. By applying sequential quadratic programming, we present how to derive the ideal speed for each core to maximize the system lifetime. We perform experiments on several multi-core platforms, which show that the proposed method can significantly outperform the existing approaches by minimizing the peak temperature of the system.

**Index Terms**—thermal-aware issues, reliability, multicore systems

## I. INTRODUCTION

Today, demands for high computation capabilities from processors keep growing. Meanwhile, semiconductor manufacturing technologies keep scaling processors to smaller feature sizes. All these factors results in high power density for modern electronic circuits. As the power density increases dramatically, systems are prone to overheating. High temperatures not only raise packaging costs, degrade system performance, and increase leakage power consumption, but also reduce the system reliability. Thermal management has become a prominent issue in system design.

Due to many limits in single core design including the performance and the power density, the microprocessor industry has switched their attentions to multicore design to enable the scaling of performance. However, multicore processors present significant new challenges to processor design. Thermal effects on multicore systems are still prominent issues. One typical thermal effect is the thermal-aware lifetime reliability, which has become a serious concern [1], [2]. In multicore systems, some cores might age much faster and die earlier than the others. If the system lifetime is determined by the shortest lifetime among the cores, then the shortest-lifetime core becomes the reliability bottleneck for multicore systems, in particular for embedded systems [3]. In this paper, we will address how to overcome such bottleneck.

Techniques for thermal management have been explored both at design time through appropriate packaging and active heat dissipation mechanisms, and at run time through various forms of Dynamic Thermal Management (DTM) such as Dynamic Voltage Scaling (DVS), in either single-core or multicore systems. Recent estimates have placed the packaging cost at \$1 to \$3 per watt of heat dissipated [4]. The techniques to reduce the packaging cost of cooling systems (e.g., the amount of cooling hardware in the system) or reduce the temperature in architectural levels have been studied in [4]–[6]. As an alternative solution, the DTM [4]–[6] has been proposed to control the temperature at run time by adjusting the system power consumption. Many modern computer architectures provide system designers with such flexibility. There have been several studies, e.g., [7], [8], for performance improvement while meeting the thermal constraints. DTM techniques could be used to address thermal-aware lifetime reliability issues.

In the literature of thermal management, only few papers have taken reliability explicitly into account. The Reliability-Aware Microprocessor (RAMP) provides a reliability model at the architecture level [2], which addresses the effects of application behavior on reliability. Previous work [9] also showed how to integrate optimization of power management with a reliability constraint and developed a simulator for analysis of power-reliability tradeoffs in system-on-chips. In [3], an analytical model is used to estimate the lifetime reliability of multicore platforms when executing periodic tasks based on the simulated annealing technique. In [10], a novel multicore simulation framework is used to simulate thermal dynamics over far longer time periods to show how job scheduling and power management policies affect system lifetime. In [2], [3], [9], [10], run-time simulation techniques were used without any analytical multicore thermal model. Therefore no tractable analysis is available for offline study. In [11], DTM techniques were studied on multicore systems with an analytical multicore thermal model, which aimed to minimize the peak temperature among all cores. Since the reliability of a core depends on not only the temperature, but also the current and the structure of the interconnect, the minimization of peak temperature might not result in an optimal system lifetime, which will be confirmed by our motivational example in Section III and the performance evaluation in Section V.

In this paper, we adopt a coarse-grained multicore thermal model for multicore systems in order to make tractable thermal-aware lifetime reliability analysis available for offline

study. As heat can transfer among cores and heat sinks, the cooling and heating phenomena is modeled by applying the Fourier's cooling model in the literature [8], [12]–[14], in which the thermal parameters can be calculated by the RC thermal model. In such model, the heat interference among different cores is considered. Although heat transfer is a dynamic process, it is not difficult to see that the temperature on a core is non-decreasing if the execution speeds on all cores are fixed. Moreover, it will end up with a steady state, in which the temperatures on all cores become steady. Assuming the unsteady state is relatively transient we can focus on the steady state. We show how to maximize the lifetime of multicore systems while maintaining a given aggregate processor speed, where the system lifetime is defined as the minimum of the lifetime among all cores. We adopt a reliability model for cores in order to perform the lifetime analysis. By applying sequential quadratic programming, we present how to derive the ideal speed for each core to maximize the system lifetime. The evaluation results show that the proposed method can significantly outperform the existing approaches by minimizing the peak temperature of the system.

The rest of this paper is organized as follows: Section II shows the system model and problem definition. Section III define the problem and motivate our proposed approach with an example. Section IV presents how to derive the ideal speeds of cores to maximize the system lifetime while maintaining a given aggregate processor speed. Section V presents performance evaluation over simulated multicore platforms. We will conclude this paper in Section VI.

## II. SYSTEM MODEL

### A. Power Consumption Model

We explore thermal-aware multicore systems, where each core has an independent DVS capabilities (referred to as DVS cores). As shown in the literature [12], [15], the power consumption  $P_j$  on Core  $j$  is contributed by:

- *The dynamic power consumption  $P_{dyn,j}$*  mainly resulting from the charging and discharging of gates on the circuits, which can be modeled by  $P_{dyn,j} = \alpha s_j^\gamma$ , where  $s_j$  is the execution speed/frequency of Core  $j$  and both  $\gamma$  ( $\leq 3$ ) and  $\alpha$  are constant.
- *The static power consumption  $P_{sta,j}$*  mainly resulting from the leakage current. The static power consumption function is a constant  $\Omega$  when the leakage power consumption is irrelevant to the temperature [16], [17]. When the leakage power consumption is related to the temperature, it is a super linear function of the temperature [18]. As shown in [12], [19], the static power consumption could be approximately modeled by an approximated linear function of the temperature with roughly 5% error. Hence, the static power consumption in this paper is as follows:  $P_{sta,j} = \delta T_j + \Omega$ , where  $T_j$  is the absolute temperature on Core  $j$  and both  $\delta$  and  $\Omega$  are non-negative constants.

As a result, the following formula is used as the overall power consumption  $P_j$  on Core  $j$  of speed  $s_j$  with temperature  $T_j$ :

$$P_j = P_{dyn,j} + P_{sta,j} = \alpha s_j^\gamma + \Omega + \delta T_j. \quad (1)$$

In this paper, we would use (1) for the power consumption for Core  $j$ .

### B. Thermal Model

We consider a multicore system, in which each core is a discrete thermal element. In the system, there is a set of heat sinks and heat spreaders, on top of the cores. We combine a heat sink and a heat spreader into a single unit as used in HotSpot model [20]. We still use heat sink to represent the combined unit. Those heat sinks generate no power, and are used only for heat dissipation. Heating or cooling is a complicated dynamic process depending on the physical system. We could approximately model this process by applying Fourier's Law in which the thermal coefficients can be obtained by using the RC thermal model, such as the approaches in [8], [12]–[14]. The thermal model adopted in this paper is similar to the approaches in [8], [12]–[14].

We define  $\mathcal{M}$  and  $\mathcal{H}$  as the set of the cores and sinks in the multicore system respectively. We assign ID for each core and sink by denoting  $\mathcal{M} = \{1, 2, \dots, n_c\}$  and  $\mathcal{H} = \{n_c + 1, n_c + 2, \dots, n_c + n_s\}$ . We define  $C_j$  as the thermal capacitance and  $R_{j,\ell}$  as the thermal conductance for any  $j, \ell \in \mathcal{M} \cup \mathcal{H}$ . We also define  $T_j(t)$  as the temperature at time  $t$  for any  $j \in \mathcal{M} \cup \mathcal{H}$ . We denote  $T_0$  the ambient temperature, which is assumed to be constant. We also define  $R_{j,0}$  as the thermal conductance between any core/sink and the ambient. We assume  $R_{j,0} = 0$  for any  $j \in \mathcal{M}$ .

We also define  $P_j(t)$  as the power consumption for  $j \in \mathcal{M} \cup \mathcal{H}$  at time  $t$ , where  $P_j(t) = 0$  for  $j \in \mathcal{H}$ . Informally, the rate of change in the temperature on a core is proportional to the power consumption times the quantity of the heating coefficient minus the cooling coefficients times the quantity of the temperature gradients among the core, its neighboring cores, and its heat sinks. The heating/cooling process by Fourier's Law can be formulated as

$$C_j \frac{dT_j(t)}{dt} = P_j(t) - R_{j,0}[T_j(t) - T_0] - \sum_{\ell \in \mathcal{M} \cup \mathcal{H}} R_{j,\ell}[T_j(t) - T_\ell(t)]. \quad (2)$$

We assume Core  $j$  runs at the constant speed  $s_j$  which is independent of time  $t$ . We can simplify (2) as

$$CT' = [B + D] - AT \quad (3)$$

where

$$C = \text{diag}(C_1, \dots, C_{n_c}, C_{n_c+1}, \dots, C_{n_c+n_s}), \quad (4)$$

$$T = (T_1, \dots, T_{n_c}, T_{n_c+1}, \dots, T_{n_c+n_s})^T, \quad (5)$$

$$B = (\Omega, \dots, \Omega, R_{n_c+1,0}T_0, \dots, R_{n_c+n_s,0}T_0)^T, \quad (6)$$

$$D = \alpha \cdot (s_1^\gamma, \dots, s_{n_c}^\gamma, 0, \dots, 0)^T, \quad (7)$$

$$A_{j,\ell} = \begin{cases} -\delta + R_{j,0} + \sum_{\ell \in \mathcal{M} \cup \mathcal{H}} R_{j,\ell}, & \text{if } \ell = j \\ -R_{j,\ell}, & \text{otherwise} \end{cases} \quad (8)$$

and (3) is multi-dimensional first-order linear differential equations with constant coefficients.

### C. Thermal-Aware Lifetime Reliability Model

Many lifetime reliability models have been proposed in the literature. This paper focuses on reliability issues due to thermal effects, in which mechanical failures and faults are not considered in this paper. For thermal-aware reliability, an electro-migration interconnection lifetime reliability model was proposed in [21], [22]. Srinivasan et al. [23] presented an application-aware architecture-level model to evaluate processors' lifetime reliability. Two analytical frameworks for the lifetime reliability of multicore systems were introduced in [24]: a cycle-accurate simulation methodology and a statistical one. A model for the lifetime reliability of homogeneous many-core systems was proposed in [25].

In this paper, we adopt the reliability model defined in [21], [22]. Our method presented later in this paper can be applied to other reliability models with some corresponding changes in the approach and analysis. By adopting the reliability model in [21], [22], we model interconnect time to failure as a resource consumed by the system over time. Specifically, we define

$$r_j(t) = I_j(t) \frac{e^{-\frac{Q_j}{\kappa_j T_j(t)}}}{\kappa_j T_j(t)} \quad (9)$$

as the consumption rate for Core  $j$ , and it represents the void growth rate in the interconnect. In (9),  $Q_j$  is the activation energy,  $\kappa_j T_j(t)$  is the thermal energy, and  $I_j(t)$  is the current density, which satisfies  $I_j(t) = \sigma_j s_j^2(t)$  for a constant  $\sigma_j$ . Hence, we have

$$r_j(t) = \sigma_j s_j^2(t) \frac{e^{-\frac{Q_j}{\kappa_j T_j(t)}}}{\kappa_j T_j(t)}. \quad (10)$$

This model captures the effect of temperature and current on electromigration lifetime. If we define  $LT_j$  as the lifetime (time to failure) of Core  $j$ , then we have

$$\int_0^{LT_j} r_j(t) dt = F_j, \quad (11)$$

where  $F_j$  is a constant determined by the structure of the interconnect. If both  $s_j$  and  $T_j$  reach constant, we have a constant consumption rate  $r_j$ . The lifetime of a core includes an unsteady state in the beginning and a steady state. Since the transient state is relatively transient, then we approximate the lifetime of a core by the lifetime in its steady state as

$$LT_j = \frac{F_j}{r_j}. \quad (12)$$

We define the system lifetime as the shortest lifetime among the cores, i.e.,

$$LT_{sys} = \min_{j \in \mathcal{M}} \{LT_j\}, \quad (13)$$

where  $LT_j$  is defined in (12).

### III. PROBLEM DEFINITION AND MOTIVATIONAL EXAMPLE

In this paper, we aim to maximize the system lifetime while maintaining a minimum system performance requirement for applications, denoted as  $s_{min}$ , a predefined minimum aggregate processor speed. Suppose that Core  $j$  is assigned with

a constant speed  $s_j$  for its execution during the lifetime of the core. Without loss of generality, we assume that the initial temperature is equal to the ambient temperature. If each core runs at its constant speed, it is clear that the temperature is non-decreasing on each core. Moreover, it will end up with a steady state, in which the temperatures on all cores become steady. Therefore, the peak temperature of Core  $j$  is no more than the temperature  $T_j^*$ , which is the solution to Equation  $\frac{dT_j}{dt} = 0$  for  $j \in \mathcal{M}$ . Similarly, we can obtain the peak temperature  $T_j^*$  of Sink  $j$  for  $j \in \mathcal{H}$ . By (3), we have  $\mathbf{A}\mathbf{T}^* = \mathbf{B} + \mathbf{D}$ . Hence the peak temperature for all cores and sinks can be derived by the following equation

$$\mathbf{T}^* = \mathbf{A}^{-1}[\mathbf{B} + \mathbf{D}], \quad (14)$$

where  $\mathbf{A}^{-1}$  is the inverse of matrix  $\mathbf{A}$ . Since matrix  $\mathbf{A}$  is only related to the hardware implementation of the multicore platform, we can calculate its inverse  $\mathbf{A}^{-1}$  as well as  $\mathbf{B} + \mathbf{D}$ . Hence, after assigning a constant execution speed of each core, the peak temperature can be easily obtained with the above formula.

We now provide an example to show why speed scaling matters for maximizing the system lifetime. Consider a system with 4 cores and 2 sinks with diagonal matrix  $\mathbf{C} = \text{diag}(0.00473, 0.00473, 0.00473, 0.00473, 0.0639, 0.0639)$  and matrix  $\mathbf{A}$  defined as follows:

$$\begin{pmatrix} 1.70 & -0.25 & 0 & 0 & -0.15 & -1.20 \\ -0.25 & 1.00 & 0 & 0 & -0.05 & -0.60 \\ 0 & 0 & 1.35 & -0.50 & -0.15 & -0.60 \\ 0 & 0 & -0.50 & 1.85 & -0.05 & -1.20 \\ -0.15 & -0.05 & -0.15 & -0.05 & 5.03 & -1.00 \\ -1.20 & -0.60 & -0.60 & -1.20 & -1.00 & 10.00 \end{pmatrix}$$

Suppose that vector  $\mathbf{B}$  is  $[4.73, 4.73, 4.73, 4.73, 0.639, 0.639]^T$ , where  $\mathbf{C}^{-1}[\mathbf{B} + \mathbf{D}]$  is  $[1000, 1000, 1000, 1000, 10, 10]^T$ .<sup>1</sup> The power consumption of a core at 1GHz is 40 ( $\alpha = 40$ ), and  $\gamma = 3$ . We assume that all the reliability-related parameters are the same among the cores, where  $Q_j = 0.84$  and  $\kappa_j = 8.62 \times 10^{-5}$  (see [26, Page 5]) for all Cores  $j$ , and  $\sigma_j = \sigma$  and  $F_j = F$  for all Cores  $j$  could be normalized.

For comparison, we consider the following two baseline approaches:

- **Power-Balance (BalPower)**: The algorithm assigns each core with the same speed  $s = \frac{s_{min}}{n_c}$ , where  $s_{min}$  is a predefined minimum aggregate processor speed.
- **Peak-Temperature-Optimization (OptTemp)**: The algorithm applies the optimization routine proposed in [11] to minimize the peak temperature among all cores while maintaining the predefined minimum aggregate processor speed  $s_{min}$ .

<sup>1</sup>The matrices  $\mathbf{A}$ ,  $\mathbf{B} + \mathbf{D}$ , and  $\mathbf{C}$  reveal that these 4 cores are with homogeneous material ( $C_1 = C_2 = C_3 = C_4$ ), the sinks are also with the same thermal capacitance, Core 1 only has heat transfer with Core 2 and the sinks, Core 3 only has heat transfer with Core 4 and the sinks, the sinks are with the same thermal conductance with  $R_{5,0}T_0 = R_{6,0}T_0 = 0.639$ , and  $\Omega = 4.73$ .

TABLE I  
RESULTS FROM THE EXAMPLE FOR DIFFERENT SPEED ASSIGNMENTS.

Cores	BalPower			OptTemp			OptLife		
	Speed (GHz)	Temperature (K)	Lifetime ( $\frac{E}{\sigma} \times 10^{10}$ )	Speed (GHz)	Temperature (K)	Lifetime ( $\frac{E}{\sigma} \times 10^{10}$ )	Speed (GHz)	Temperature (K)	Lifetime ( $\frac{E}{\sigma} \times 10^{10}$ )
1	1	356.7	2.24	1.11	364.7	1.02	1.08	362.1	1.29
2	1	375.2	0.61	0.88	364.7	1.64	0.91	367.0	1.29
3	1	368.2	0.98	0.93	364.7	1.46	0.95	365.7	1.29
4	1	359.7	1.80	1.08	364.7	1.09	1.06	362.9	1.29

Suppose that these four cores are requested to provide 4GHz computation frequency requirement. If we try to minimize the impact on the current  $I_j(t) \propto s_j^2(t)$ , we would try to balance the current to achieve the computation requirement. As a result, for the BalPower approach, we will assign each core to run at 1GHz. The resulting temperature and lifetime of each core of the above speed assignment are presented in Table I. The weakness of the BalPower approach is that it might increase the peak temperature of the system, and, hence, the system lifetime is reduced because of the overheated core, which is Core 2 in this example. On the other hand, we could also try to reduce the peak temperature by applying the OptTemp approach proposed in [11]. In this example, the minimum peak temperature among these 4 cores is 364.7K, in which the corresponding speed assignment and lifetime of the cores are presented in Table I. For such a case, the core with the highest current, which is Core 1 in this example, might sacrifice the system lifetime. Therefore, to maximize the system lifetime, we have to balance the temperature and the current on a core. In Section IV, we will present how to maximize the system lifetime (the algorithm is called OptLife), whereas the corresponding solution of the example is presented in Table I. As shown in Table I, the approach in this paper improves the system lifetime by  $\frac{1.29}{0.61} - 1 = 1.11$  ( $\frac{1.29}{1.02} - 1 = 0.26$ , respectively) in comparison with BalPower and OptTemp approaches.

#### IV. DERIVING IDEAL SPEEDS

This section presents how to derive the ideal speed of each core so that the system lifetime is maximized while maintaining a predefined minimum aggregate processor speed.

The optimization of the system lifetime subject to a predefined minimum aggregate processor speed can be written as

$$\begin{aligned}
 & \text{maximize } LT_{sys} = \min_{j \in \mathcal{M}} \left\{ LT_j = \frac{F_j}{r_j} \right\} \\
 & \text{subject to } \sum_{j \in \mathcal{M}} s_j \geq s_{min}, \\
 & \quad s_j \geq 0, j \in \mathcal{M}. \tag{15}
 \end{aligned}$$

Obviously, an optimal solution to (15) will set  $s_j$  to zero where  $j \in \mathcal{H}$ . Thus, we do not specify the constraints of the sinks in the above system. Moreover, (15) is equivalent to the following programming:

$$\begin{aligned}
 & \text{minimize } \max_{j \in \mathcal{M}} \left\{ \frac{r_j}{F_j} \right\} \\
 & \text{subject to } \sum_{j \in \mathcal{M}} s_j \geq s_{min}, \\
 & \quad s_j \geq 0, j \in \mathcal{M}. \tag{16}
 \end{aligned}$$

For the rest of this section, we will present how to extend the sequential quadratic programming [27] to solve the above non-linear programming.

Clearly, the value  $\max_{j \in \mathcal{M}} \left\{ \frac{r_j}{F_j} \right\}$  is no less than 0. Starting with the initial guess  $\rho_0$  on  $\max_{j \in \mathcal{M}} \left\{ \frac{r_j}{F_j} \right\}$ , we try to approach the optimal solution step by step. That is, for the  $i$ -th iteration, based on  $\rho_{i-1}$ , we derive another value  $\rho_i$  with  $\rho_i > \rho_{i-1}$  such that  $\rho_i$  is getting more close to the optimal solution of (16). Specifically, at the  $i$ -th step, we first solve the following unconstrained non-linear programming by applying the sequential quadratic programming method:

$$\begin{aligned}
 & \text{minimize } \sum_{j \in \mathcal{M}} [\max\{0, \frac{r_j}{F_j} - \rho_{i-1}\}]^2 \\
 & \quad + \epsilon_1 [s_{min} - \sum_{j \in \mathcal{M}} s_j]^2, \tag{17}
 \end{aligned}$$

where  $r_j = s_j^2 \frac{e^{\kappa_j T_j^*}}{\kappa_j T_j^*}$  and  $T_j^*$  is the peak temperature on core  $j$  by setting the corresponding speeds in (14), and  $\epsilon_1$  is a positive constant related to the rate of convergence from  $\rho_{i-1}$  to  $\rho_i$ . In general, the constant  $\epsilon_1$  should be set to large numbers for deriving precise results. Suppose that the optimal solution of (17) is  $\Delta_i$ . Then, we can set  $\rho_i$  as  $\rho_{i-1} + \left(\frac{\Delta_i}{n_c}\right)^{\frac{1}{2}}$ . The above procedure repeats until  $\left(\frac{\Delta_i}{n_c}\right)^{\frac{1}{2}}$  is very small. As shown in [27], the resulting speed assignment with the converged  $\rho_i$  is very close to the optimal solution. The detailed algorithm called OptLife is presented in Algorithm 1.

---

#### Algorithm 1 OptLife

---

**Input:**  $A^{-1}, B, s_{min}, \alpha;$

**Output:**  $s_1, s_2, \dots, s_{n_c};$

- 1:  $\rho_0 \leftarrow 0, i \leftarrow 1;$
  - 2: **while true do**
  - 3: find the optimal value  $\Delta_i$  of (17), where  $T_j^*$  is the peak temperature on core  $j$  by setting the corresponding speeds in (14);
  - 4: **if**  $\sqrt{\frac{\Delta_i}{n_c}}$  is less than the threshold of precision **then**
  - 5: return  $\zeta \cdot (s_1, s_2, \dots, s_{n_c})$ , where  $\zeta \leftarrow \frac{s_{min}}{\sum_{j \in \mathcal{M}} s_j};$
  - 6: **else**
  - 7:  $\rho_i \leftarrow \rho_{i-1} + \sqrt{\frac{\Delta_i}{n_c}};$
  - 8:  $i \leftarrow i + 1;$
  - 9: **end if**
  - 10: **end while**
- 

#### V. PERFORMANCE EVALUATION

In Section III, we motivated our proposed approach, OptLife in Algorithm 1 and compared it with Algorithms BalPower and Algorithms OptTemp defined in [11]. In

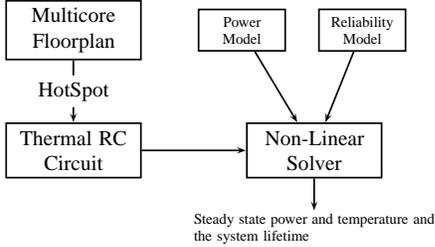


Fig. 1. Numerical computation of steady state power and temperature by using Hotspot and reliability model in (10).

this section, we will conduct more comprehensive experimental study to compare their performance. Note that DTM approaches, e.g., [7], [8], to maximize the performance under the thermal constraints could also be applied. However, such approaches require correct settings on the thermal constraints to maximize the lifetime reliability. As a result, we do not compare with such approaches.

#### A. Experiment Setup

Figure 1 describes the method used to obtain the steady state power and temperature and system lifetime for a given minimum processor speed. The floor plan of the multicore system is first fed into HotSpot simulator [28]. The output of HotSpot is the thermal RC circuit, which together with the dynamic/static power consumption models is then feeded into the non-linear solver to output the speed assignment. The non-linear solver applies either Algorithm BalPower, Algorithm OptLife, or Algorithm OptTemp, to derive the speed assignment for a given input and then the corresponding system lifetime in the steady state.

We simulated two types of chips, says *fixed-core structure* and *fixed-chip structure*. For the fixed-core structure case, we take the single-core Alpha 21264 EV6 floorplan from [28], scale its dimensions to a fixed size. We simulate chips with 4 cores and 8 cores for the fixed-core structure study. For the  $2 \times 2$  chip with 4 cores in such a structure, we dimension 4 cores on the chip. For a  $4 \times 2$  with 8 cores in such a structure, we dimension two  $2 \times 2$  blocks by doubling the chip width. For the fixed-chip structure, we fix the chip size, in which the sizes of the cores are scaled to fit into the chip size. For a given floorplan, Hotspot [28] computes the equivalent thermal RC circuit. The corresponding thermal parameters are determined based on the layout. The ambient temperature in the simulations is assumed as 30 °C, in which all the other thermal parameters are based on HotSpot default settings. We use the default settings of spreaders and heat sink in HotSpot for cooling.

To evaluate the impact on the power consumption functions, we use different power consumption settings in our simulations. We fix the dynamic power consumption as  $\alpha$ Watt at 1GHz. The power consumption function is  $\alpha(\frac{s_j}{1GHz})^3 + 10 + 0.02T_j$  Watt in our simulations, where  $s_j$  is the speed of Core  $j$  and  $T_j$  is the temperature of Core  $j$ . By adopting the reliability parameters in [22] and [26, Page 5], for Core  $j$ , we use  $Q_j = 0.84\text{eV}$  and  $\kappa_j = 8.62 \times 10^{-5}\text{eV/atom/K}$

in our simulations, where  $\sigma_j$  and  $F_j$  are both assumed to be constants  $\sigma$  and  $F$ , respectively. As we are interested in the comparison between the evaluated algorithms, the constants  $F$  and  $\sigma$  could be any fixed numbers, and the results are the same.

By varying different parameters, we evaluate the performance of Algorithms BalPower, OptTemp, and OptLife. To demonstrate the performance between these algorithms, we choose Algorithm BalPower as the baseline, in which the *normalized lifetime* of an algorithm for an input instance is defined as the system lifetime of the solution of the algorithm divided by the system lifetime of the solution of Algorithm BalPower. We compare the normalized lifetime for Algorithm OptTemp and Algorithm OptLife. Moreover, as Algorithm OptTemp is, most of time, better than Algorithm BalPower (we will show later), to see how much improvement of Algorithm OptLife in comparison with Algorithm OptTemp, we use *Improvement Ratio* defined as the system lifetime of the solution of Algorithm OptLife divided by that of Algorithm OptTemp.

#### B. Evaluation Results

For fixed-core structures, Figure 2 presents the simulation results by varying the minimum aggregate processor speed  $s_{min}$  with  $\alpha = 30$ . Figure 2(a) (Figure 2(b), respectively) is for 4 (8, respectively) cores. As we use the same settings on heat sinks, when there are more cores in the chip, it is, in general, more difficult to dissipate heat. Moreover, when the minimum aggregate processor speed requirement is larger, applying Algorithm BalPower leads to speed assignments with very unbalanced temperatures among these cores. Therefore, minimizing peak temperature by applying Algorithm OptTemp when  $s_{min}$  is larger reduces the peak temperature significantly, and, hence increases the system lifetime as well as the normalized lifetime. However, it is noticed that Algorithm OptTemp is not always better than Algorithm BalPower. When  $s_{min} \leq 4$  in Figure 2(a), as the temperature variance among these 4 cores is not significant, the current density  $I_j()$  plays a more important rule, and, hence, Algorithm BalPower derives better solutions. Algorithm OptLife, proposed in this paper, outperforms Algorithm BalPower and Algorithm OptTemp for all cases. The normalized lifetime of Algorithm OptLife in Figure 2 ranges from 1.05 to 1.2 for 4 cores and from 2.1 to 5.2 for 8 cores. Moreover, as  $s_{min}$  increases, the normalized lifetime of Algorithm OptLife also increases, in which the reason is the same as that for Algorithm OptTemp. However, when  $s_{min}$  is larger, the system lifetime maximization is highly related to the peak temperature. Therefore, Algorithm OptTemp has benefit when  $s_{min}$  is large, in which improvement ratio of Algorithm OptLife decreases when  $s_{min}$  increases. In Figure 2, Algorithm OptLife improves the system lifetime of Algorithm OptTemp by 8% to 15% for 4 cores and by 45% to 55% for 8 cores.

For fixed-chip structures, Figure 3 presents the simulation results by varying the minimum aggregate processor speed  $s_{min}$  with  $\alpha = 30$ . Note that the core size is smaller than the

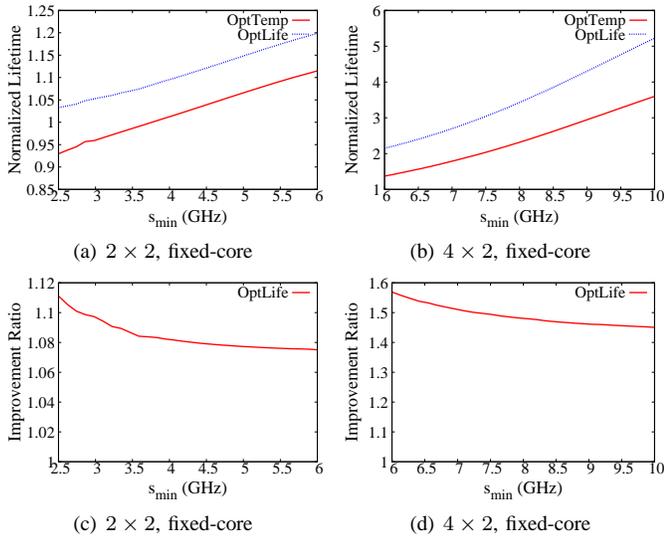


Fig. 2. Simulation results by varying  $s_{min}$  for fixed-core structures.

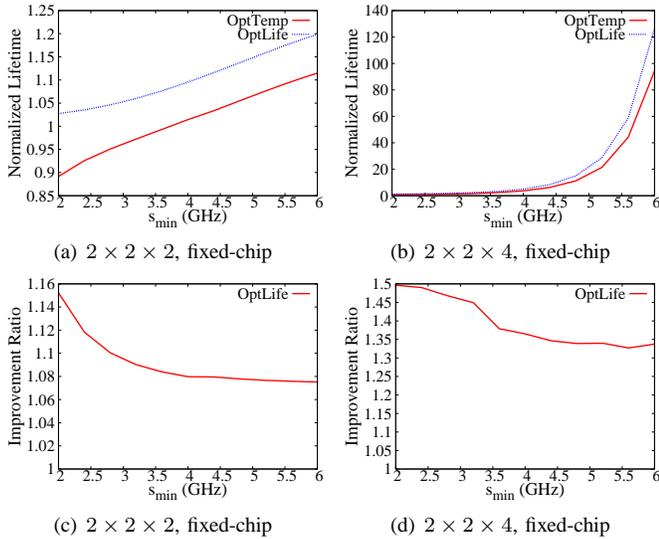


Fig. 3. Simulation results by varying  $s_{min}$  for fixed-chip structures.

studied case in Figure 2, which implies that these cores are easier to be heated up. However, as  $\frac{s_{min}}{n_c}$  is only 50% of the setting in Figure 2, the power consumption of each core is much less. In general, the results in Figure 3 are quite similar to those in Figure 2, since, again, the performance is highly dependent of the power consumption. In our simulations, there is no much difference between the fixed-chip structures and the fixed-core structures. For the rest of this subsection, we will then only present the results for fixed-core structures only.

For fixed-core structures, Figure 4 shows the simulation results by varying  $\alpha$  for different power consumption settings with  $s_{min} = 4\text{GHz}$  ( $s_{min} = 8\text{GHz}$ , respectively) for 4 (8, respectively). Similar to the analysis in the previous paragraph for Figure 2, as  $\alpha$  increases, the cores generate more heat, and hence, the normalized lifetime increases for both Algorithm OptTemp and Algorithm OptLife. The results for improvement ratio are similar to that in Figure 2(c) and Figure 2(d) for 4 cores and 8 cores, respectively, and, are omitted here.

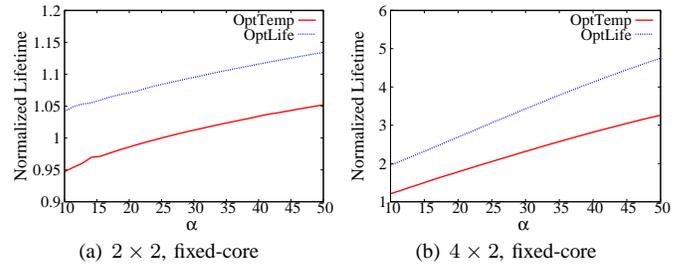


Fig. 4. Simulation results by varying  $\alpha$  for different power consumptions.

## VI. CONCLUSION

In this paper, we investigated the thermal-aware lifetime reliability issues in multicore systems, where the system lifetime is determined by the shortest lifetime among the cores. In multicore systems, some cores might age much faster and die earlier than the others, which becomes the reliability bottleneck, in particular for embedded systems. We have shown that either balancing power consumption or minimizing the peak temperature among cores is not sufficient to optimize the system lifetime. For this, we proposed an algorithm to derive an ideal speed assignment in order to maximize the system lifetime of multicore systems while maintaining a given aggregate processor speed. We performed comprehensive experiments on several multi-core platforms, which show that the proposed method can significantly outperform the existing approaches by either balancing power consumption or minimizing the peak temperature among cores. In our experimental results, our approach improves the system lifetime of existing approach for minimizing the peak temperature by 8% to 55%, and improves the naive approach for balancing power consumption among cores very significantly.

## ACKNOWLEDGMENT

This work is sponsored in part by NSF CAREER Grant No. CNS-0746906 and the European Community's Seventh Framework Programme FP7/2007-2013 project Predator (Grant 216008).

## REFERENCES

- [1] S. Borkar, "Designing reliable systems from unreliable components: the challenges of transistor variability and degradation," *IEEE Micro*, vol. 25, no. 6, pp. 10–16, 2005.
- [2] J. Srinivasan, S. Adve, P. Bose, and J. Rivers, "The case for lifetime reliability-aware microprocessors," in *Proceedings of International Symposium on Computer Architecture*, 2004.
- [3] L. Huang, F. Yuan, and Q. Xu, "Lifetime Reliability-Aware Task Allocation and Scheduling for MPSoC Platforms," in *Proceedings of DATE: Design, Automation and Test in Europe*, 2009.
- [4] K. Skadron, M. R. Stan, W. Huang, S. Velusamy, K. Sankaranarayanan, and D. Tarjan, "Temperature-aware microarchitecture," in *International Symposium on Computer Architecture*, 2003.
- [5] 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.
- [6] D. Brooks and M. Martonosi, "Dynamic thermal management for high-performance microprocessors," in *International Symposium on High-Performance Computer Architecture*, 2001.
- [7] M. Kadin and S. Reda, "Frequency planning for multi-core processors under thermal constraints," in *ISLPED*, 2008, pp. 213–216.

- [8] 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.
- [9] T. Rosing, K. Mihic, and G. De Micheli, "Power and reliability management of SoCs," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 15, no. 4, p. 391, 2007.
- [10] A. K. Coskun, R. Strong, D. Tullsen, and T. Rosing, "Evaluating the Impact of Job Scheduling and Power Management on Processor Lifetime for Chip Multiprocessors," in *Proceedings of ACM SIGMETRICS/Performance*, 2009.
- [11] J. Chen, S. Wang, and L. Thiele, "Proactive Speed Scheduling for Real-Time Tasks under Thermal Constraints," in *Proceedings of IEEE Real-Time Technology and Applications Symposium*, 2009.
- [12] T. Chantem, R. P. Dick, and X. S. Hu, "Temperature-aware scheduling and assignment for hard real-time applications on MPSoCs," in *Proceedings of DATE: Design, Automation and Test in Europe*, 2008.
- [13] 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.
- [14] S. Murali, A. Mutapcic, D. Atienza, R. Gupta, S. Boyd, L. Benini, and G. D. Micheli, "Temperature control of high-performance multi-core platforms using convex optimization," in *Proceedings of DATE: Design, Automation and Test in Europe*, 2008.
- [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] R. Xu, D. Zhu, C. Rusu, R. Melhem, and D. Mosse, "Energy efficient policies for embedded clusters," in *ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems*, 2005.
- [17] 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.
- [18] 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*, vol. 24, no. 7, pp. 1042–1053, 2005.
- [19] Y. Liu, R. P. Dick, L. Shang, and H. Yang, "Accurate temperature-dependent integrated circuit leakage power estimation is easy," in *Proceedings of DATE: Design, Automation and Test in Europe*, 2007.
- [20] M. Stan, K. Skadron, M. Barcella, W. Huang, K. Sankaranarayanan, and S. Velusamy, "Hotspot: A dynamic compact thermal model at the processor-architecture level," *Microelectronics Journal*, vol. 34, no. 12, pp. 1153–1165, 2003.
- [21] Z. Lu, J. Lach, M. Stan, and K. Skadron, "Improved thermal management with reliability banking," *IEEE Micro*, vol. 25, no. 6, pp. 40–49, 2005.
- [22] Z. Lu, W. Huang, M. Stan, K. Skadron, and J. Lach, "Interconnect lifetime prediction for reliability-aware systems," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 15, no. 2, pp. 159–172, 2007.
- [23] J. Shin, V. Zyuban, Z. Hu, J. Rivers, and P. Bose, "A framework for architecture-level lifetime reliability modeling," in *Proceedings of the 37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, 2007.
- [24] A. Coskun, T. Rosing, K. Mihic, G. De Micheli, and Y. Leblebici, "Analysis and optimization of MPSoC reliability," *Journal of Low Power Electronics*, vol. 2, no. 1, p. 56, 2006.
- [25] L. Huang and Q. Xu, "On Modeling the Lifetime Reliability of Homogeneous Manycore Systems," in *Proceedings of the 2008 14th IEEE Pacific Rim International Symposium on Dependable Computing*, 2008, pp. 87–94.
- [26] J. Black, "Mass transport of aluminum by momentum exchange with conducting electrons," in *Proceedings of 43rd Annual IEEE International Reliability Physics Symposium*, 2005.
- [27] S. R. K. Dutta and M. Vidyasagar, "New algorithms for constrained minimax optimization," *Journal Mathematical Programming*, vol. 13, no. 1, pp. 140–155, 1977.
- [28] W. Huang, S. Ghosh, S. Velusamy, K. Sankaranarayanan, K. Skadron, and M. Stan, "HotSpot: A compact thermal modeling methodology for early-stage VLSI design," *IEEE Transactions on Very Large Scale Integration Systems*, vol. 14, no. 5, p. 501, 2006.