Energy-Optimal Dynamic Thermal Management for Green Computing *

Donghwa Shin,
Jihun Kim
and Naehyuck Chang†
Seoul National University, Korea
{dhshin, jhkim, naehyuck}
@elpl.snu.ac.kr

Jinhang Choi,
Sung Woo Chung
Korea University, Korea
{cepiross, swchung}
@korea.ac.kr

Eui-Young Chung
Yonsei University, Korea
eychung@yonsei.ac.kr

ABSTRACT

Existing thermal management systems for microprocessors assume that the thermal resistance of the heat-sink is constant and that the objective of the cooling system is simply to avoid thermal emergencies. But in fact the thermal resistance of the usual forced-convection heat-sink is inversely proportional to the fan speed, and a more rational objective is to minimize the total power consumption of both processor and cooling system. Our new method of dynamic thermal management uses both the fan speed and the voltage/frequency of the microprocessor as control variables. Experiments show that tracking the energy-optimal steady-state temperature can saves up to 17.6% of the overall energy, when compared with a conventional approach that merely avoids overheating.

1. INTRODUCTION

As the power density of microprocessors increases, more elaborate methods of thermal management are required. These cause from the air conditioners necessary in a large data center, to the cooling fans in a sub-notebook computer. The power consumption of active cooling systems is significant, and can usually be adjusted dynamically. For instance, the thermal resistance of a forced-convection heat-sink is determined by the rotational speed of the fan. A typical cooling fan is driven by a brushless DC motor with a feedback speed controller, so that the fan speed can be controlled by software. A higher speed produces a lower thermal resistance, but uses more power.

Modern forced-convection heat-sinks for desktop computers have a thermal resistance from 0.2 to 0.6°C/W and can consume several watts. The number of fans required by multiple-node servers must be more than proportional to the number of nodes, in order to compensate for higher power densities. In an efficient system, it is crucial to reduce the cooling power as far as possible, while avoiding thermal emergencies in the microprocessor. Conventional systems commonly maintain the lowest fan speed that avoids a thermal emergency, and this minimizes the power consumption of the cooling system.

Unfortunately, this method of controlling a cooling fan does not minimize the total system power consumption. Leakage power increases as the scale of semiconductor technology is reduced, and this power is now very significant. Furthermore, it increases exponentially with the die temperature. Thus, an approach to cooling fan management that simply avoids thermal emergencies may result in excessive leakage power consumption in the microprocessor. It is reasonable to supply more power to the cooling fan if this produces a disproportionate drop in leakage power. We assert that there is an important tradeoff between the power consumption of a microprocessor and its cooling fan. This strongly suggests that the power consumption of these two components should be jointly optimized to achieve a globally minimum total power consumption.

Although commercial forced-convection heat-sinks are able to control their thermal resistance by adjusting the speed of the cooling fan, existing dynamic thermal management (DTM) techniques do not consider the cooling fan speed as a control variable, but take the thermal resistance of the heat-sink as a constant. Typical commercial systems only control the speed of the cooling fan while the die temperature is above a threshold. Some commercial DTM systems give priority to controlling either the fan speed or the microprocessor clock frequency, which cannot minimize the total power consumption.

We introduce a new DTM technique in which the power consumption of a microprocessor and its cooling fan are jointly minimized. We formulate and solve an optimization problem in which the temperature-dependent leakage power consumption of the microprocessor and the power consumption of the cooling fan form a convex function. We believe this is the first approach in which a DTM uses the speed of the cooling fan as a control variable. Our energy-optimal DTM has two control variables: the scaling factor used for dynamic supply voltage/frequency scaling (DVFS), and the fan speed.

2. RELATED WORK

Early work in DTM resulted in two widely used scaling techniques, dynamic frequency scaling (DFS) and DVFS, and three micro-architectural techniques, decode throttling, speculation control, and I-cache toggling [1]. More recently, profiling-based predictive DPM has been proposed for multimedia applications [2]. Another kind of thermal management scheme, designed for a multiprocessor environment, reschedules tasks to make use of idle symmetric multiprocessor nodes [3].

Since the effect of leakage current on dynamic voltage scaling [4] has become apparent, power models have included leakage

*This work is supported by the Brain Korea 21 Project; National Research Foundation of Korea (NRF) grant funded by the Korean government (MEST) (grant code: 2009-0060054); The Korea Research Foundation Grant funded by the Korean Government (KRF-2009-013-D00099); and the IC Design Education Center (IDEC). The ICT at Seoul National University provides research facilities for this study.

†Corresponding author

ICCAD'09 November 2–5, 2009, San Jose, California.
Copyright 2009 ACM 978-1-60558-800-1/09/11 ...$10.00.

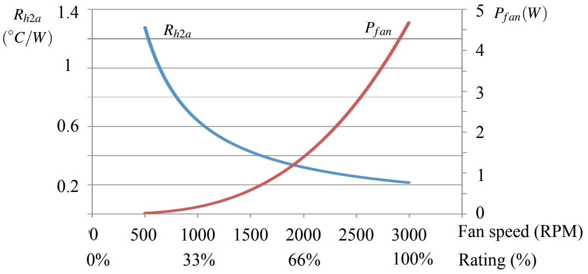


Figure 1: The thermal resistance and power consumption of a forced-convection heat-sink composed of a parallel plate copper fin heat-sink $(70 \times 70 \times\ 50mm^3)$ and a $70mm$ cooling fan.

power. An improved model of leakage current [5] shows its exponential dependence on temperature. Based on this model, another working group proposed an algorithm to minimize temperature-aware leakage in real-time systems [6].

The growth in research on thermal management has increased the need for an accurate model of thermal behavior. HotSpot [7] is a simuator for developing precise but compact thermal models for the popular stacked-layer packaging scheme used in modern VLSI systems. HotSpot has become a *de facto* standard for thermal simulation, and we use this tool to evaluate our algorithm.

None of the work mentioned above deals with the optimality of the total energy consumption of a system. As we have already said, existing DTM schemes primarily focus on avoiding a thermal emergency. The speed of the cooling fan is varied with die temperature, but the thermal resistance of the heat-sink is considered to be constant.

Some recent research has considered combining DTM scheduling with optimizing the throughput of a given set of tasks. By accounting for the different thermal conductivities and heat capacities of the chip and its package, exponentially time-varying speed control of a microprocessor [8] can reduce the loss of performance involved in the earlier constant throttling technique. Another approach [9] addresses the problem of optimizing the performance of a set of periodic tasks using the discrete voltage/frequency states available on actual processors. Both of these approaches retain the constraint that the thermal threshold should not be violated. They optimize the throughput in a multiple-task environment, but they do not attempt overall energy minimization.

3. POWER AND THERMAL MODELS

3.1 Modeling the thermal resistance and power consumption of a heat-sink

One of the most common types of cooling system is a forced-convection heat-sink. It consists of a heat-sink made of a material of low thermal resistance and a cooling fan that circulates ambient air over and through the heat-sink. We will exclude liquid cooling systems from discussion, but they could be accommodated in a similar optimization framework because their power consumption and thermal resistance have a similar relation.

A forced-convection heat-sink is a heat exchanger that transfers heat from a microprocessor to ambient air [10]. The temperature of the microprocessor is determined by the amount of heat transferred from the device to the heat exchanger, which is in turn determined by the thermal resistance of the latter. The thermal resistance of a forced-convection heat-sink varies with the amount of convection, which is itself determined by the speed of the cooling fan. We will now describe a model of the power consumption of a forced-convection heat-sink and thermal resistance. We will assume that a typical commercial fan speed control scheme is in use.

We model the thermal resistance R_{h2a} of a heat-sink as a function of the power of the fan P_{fan}. Out of several existing models of a forced-convection heat-sink [11, 12, 10], we select the thermal exchanger [10] which is given by

$$R_{h2a} = \left(mc_p \left(1 - e^{\left(-\frac{hA_e}{mc_p} \right)} \right) \right)^{-1}, \tag{1}$$

where $m = \rho \upsilon_f / \Delta$ is the mass flow-rate of the air; υ_f is the velocity of the air; ρ is the density of the air; Δ is the cross–sectional area of the air channel; c_p is the specific heat of the air; A_e is the effective area of the heat-sink; $h = kN_u/D_h$ is the heat transfer coefficient of the heat-sink, in which the Nusselt number N_u can in turn be approximated as a function of the Reynolds number; $R_e = \upsilon_f D_h / \nu$ is the Reynolds number; D_h is the hydraulic diameter of the air channel; ν is the viscosity of the air; and k is the thermal conductivity of the heat-sink material.

Finally, we are able to represent the thermal resistance as a function of υ_f with physical coefficients $h_1, \cdots, h_4$ as follows:

$$R_{h2a} = \left(h_1 \upsilon_f \left(1 - e^{\frac{h_2 \upsilon_f^{h_3} + h_4}{h_1 \upsilon_f}} \right) \right)^{-1}. \tag{2}$$

For a fixed air channel, the flow-rate and the velocity of the air are determined by the speed of the fan. By conservation of energy, the energy consumed in rotating the fan is the same as the energy required to deliver the air:

$$P_{fan} \propto \upsilon_f{}^3. \tag{3}$$

The efficiency of air delivery is determined by factors which include the shape of the channel and friction. By substituting (3) into (2), the thermal resistance of a forced-convection heat-sink can be expressed as a function of its power consumption. For simplicity, we will use (3) alone which allows us to manipulate P_{fan} instead of the fan speed in the formulars which follow.

Fig.1 shows how the thermal resistance and power consumption of the forced-convection heat-sink change significantly with the fan speed. While old-fashioned cooling fans operate at a constant speed, a modern forced-convection heat-sink has an encoder that communicates the speed of the fan to the microprocessor that it is cooling. This microprocessor will be equipped with temperature sensors, and can control the fan speed using pulse width modulation (PWM).

3.2 Modeling the power consumption of a microprocessor

In this paper, we model the power consumption of a microprocessor as a function of the following known parameters: the effective switching capacitance C_e, the supply voltage V_{dd}, the operating clock frequency f, and the technology constant K_n.

The power consumption of a CPU can be expressed as:

$$P_{cpu} = P_d + P_s + P_0, \tag{4}$$

where P_d, P_s and P_0 respectively are the dynamic, static, and always-on power consumption. The dynamic power consumption is given by

$$P_d = \frac{1}{2} C_e V_{dd}^2 f. \tag{5}$$

Although a more detailed model might be formulated, we consider it sufficient to include the two major consumers of leakage power in the static power model. These are the subthreshold leakage and the gate leakage power. The static power consumption is also dependent on the die temperature T_d, and can be expressed as follows:

$$P_s(T_d) = V_{dd} \left(K_1 T_d^2 e^{\frac{K_2 V_{dd} + K_3}{T_d}} + K_4 e^{(K_5 V_{dd} + K_6)} \right). \tag{6}$$

We expand the right-hand side of this equation as a Taylor series and retain its linear terms:

$$\begin{aligned} P_s(T_d) &= \sum_{n=0}^{\infty} \left(\frac{1}{n!} \right) \frac{d^n P_s(T_r)}{dT_d^n} (T_d - T_r)^n \\ &\approx P_s(T_r) + \frac{dP_s(T_r)}{dT_d} (T_d - T_r), \end{aligned} \tag{7}$$

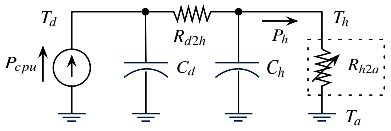


Figure 2: RC-thermal circuit model.

where T_r is a reference temperature, which is usually some average value within the operational temperature range. Note that an approximation of this form would allow us to accommodate additional leakage power sources, if the gain in accuracy were likely to be significant. However, within an ordinary temperature range of 25°C to 120°C we may expect an error of less then 5% with a simple linear model [13].

While this model provides relatively accurate power estimation, it does not reflect local hotspots. Spatial variations in temperature are beyond the scope of this paper and an approach to thermal management which took them into account would need to be the subject of further research.

3.3 Combined thermal and power model with an adjustable thermal resistance

We use a typical RC-thermal model [14, 15], as shown in Fig.2, to analyze the thermal dynamics of a microprocessor and its cooling system. T_d is the die temperature; C_d is the thermal capacitance of the die; R_{d2h} is the thermal resistance from the die to the package combined with its heat-sink; C_h is the thermal capacitance of the package combined with its heat-sink; P_h is the heat dissipated by the heat-sink; T_h is the temperature of the heat-sink; and T_a is the ambient temperature. Since we are able to adjust the fan speed, our model has the distinct feature that the thermal resistance R_{h2a} is a variable and not a constant. This makes the problem statement and the solution completely different from previous DTM formulations. Both T_d and T_h can be determined from the following equations:

$$P_{cpu} = C_d \frac{dT_d}{dt} + \frac{T_d - T_h}{R_{d2h}}, \tag{8}$$

$$\frac{T_d - T_h}{R_{d2h}} = C_h \frac{dT_h}{dt} + \frac{(T_h - T_a)}{R_{h2a}}. \tag{9}$$

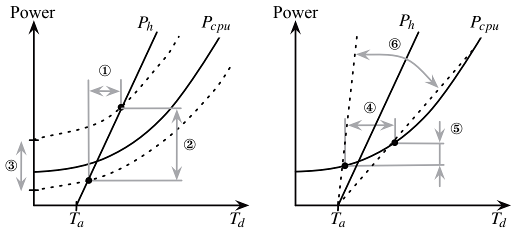
Fig.3(a) shows a conventional thermal management system in which the thermal resistance of the heat-sink, with or without a cooling fan, is constant. In this case, the thermal equilibrium die temperature can be obtained as follows:

$$P_{cpu} = P_h = \frac{T_d - T_a}{R_{d2h} + R_{h2a}}. \tag{10}$$

If the dynamic power consumption of the microprocessor P_d increases so that the lower dashed curve in Fig.3(a) is replaced by the upper dashed curve (marked ③), then both the die temperature T_d and the CPU power P_{cpu} at thermal equilibrium (marked ① and ②) increase. Note that the extent of the increase in P_{cpu} is larger than the increase in P_d. This is due to the way in which the temperature-dependent leakage power varies with temperature.

We have already made it clear that we use the thermal resistance of a force-convection heat-sink as a control variable, which can be changed by adjusting the fan speed, as shown in (2) and Fig.3(b). A change in fan speed alters the slope of the P_h curve (⑥ in Fig.3(b)). If the thermal resistance $R_{d2h} + R_{h2a}$ were to be zero, which cannot of course happen in reality, then T_d were equal T_a. A low fan speed increases the thermal resistance of the cooling system, producing a high thermal equilibrium T_d (the dashed line on the right of Fig.3(b)). Although the die temperature may be lower than the thermal emergency temperature, it may still incur a large temperature-dependent leakage power. A higher fan speed reduces T_d (④ of Fig.3(b)) and thus the leakage power (⑤ of Fig.3(b)). If the extent of the reduction in the temperature-dependent leakage power is larger than the additional power used by the fan, the total power consumption is reduced.

We can now formulate a total power model which combines temperature-dependent leakage and thermal resistance in a thermal equilibrium:



(a) Variation in T_d ① and P_{cpu} ② with dynamic power ③.

(b) Variation in T_d ④and P_{cpu} ⑤ with thermal resistance R_{h2a}⑥.

Figure 3: The effect of a variable thermal resistance, achieved by controlling the fan speed, on the thermal equilibrium die temperature.

Table 1: Comparison of (11) with the HotSpot simulation.

V_{dd} (V)	f (GHz)	P_{total} (W) from HotSpot 40°C	60°C	80°C	100°C
1.35	3.00	N/A	104.67	109.81	115.17
1.30	2.67	78.37	82.46	86.55	N/A
1.25	2.25	59.31	62.38	65.51	N/A
1.20	1.87	46.24	48.74	N/A	N/A
V_{dd} (V)	f (GHz)	P_{total} (W) from (11) 40°C	60°C	80°C	100°C
1.35	3.00	98.40	103.48	108.56	113.63
1.30	2.67	78.10	82.10	86.10	90.10
1.25	2.25	60.80	63.87	66.94	70.01
1.20	1.87	44.44	46.62	48.80	50.97

$$\begin{aligned} P_{total} &= P_{cpu} + P_{fan}, \\ &= P_d + (\alpha \cdot \frac{(R_{h2a} + R_{d2h})(\beta + P_d + P_0) + T_a}{1 - \alpha(R_{h2a} + R_{d2h})} + \beta) \\ &\quad + P_0 + P_{fan}, \end{aligned} \tag{11}$$

where the temperature-dependent leakage power is linearized so that $\alpha = dP_s(T_r)/dT_d$ and $\beta = P_s(T_r) - T_r dP_s(T_r)/dT_d$.

We compared the result of estimating the power consumption using our analytical model with the result of HotSpot simulation. We modified HotSpot [16] so that the thermal resistance of the heat-sink can be changed during run-time to accommodate an adjustable forced-convection heat-sink, and integrated it with Wattch [17, 18]. We simulated the Intel Xeon E7330 quad-core processorrunning the gcc benchmark from SPEC2000 [19], since gcc is known to cause large variations in temperature over time [20]. We used the performance monitoring unit on the microprocessor to obtain activity counts for each functional block of the microprocessor [20] while executing SPEC2000. Wattch estimates the power consumption of a microprocessor using these activity counts, and HotSpot generates a temperature profile using the power consumption values from Wattch. Simulation results with discrete-level DVFS are shown in Table 1. To compare these results with (11), we extracted the parameters of the power model from Wattch and HotSpot, and calculated the total power consumption at four different die temperatures with the same supply voltage and frequency. As shown in Table 1, the discrepancy between simulation results and analytic prediction is less than 5%.

4. JOINT POWER AND THERMAL OPTIMIZATION

We will now address the joint optimization of cooling power and microprocessor power. It is relatively easy to derive an optimal cooling fan speed for continuous task execution with a fixed supply voltage and frequency. As illustrated by Fig.4, both P_{cpu} and P_{fan} are convex functions of the fan speed. Thus the total power consumption P_{total} is convex, and so the optimal fan speed can be calculated by differentiating (11). We will deal with more practical situations in the following sections.

4.1 A general (not real-time) task with a continuous voltage and frequency domain

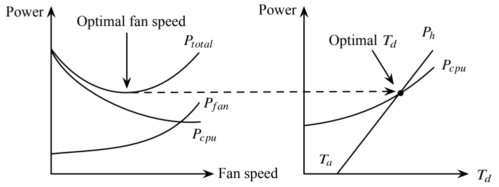


Figure 4: Optimal cooling fan speed in thermal equilibrium.

We will now consider energy-optimal cooling by controlling fan speed together with supply voltage/frequency scaling for a given batch workload, W_b. The microprocessor operates at a reduced clock frequency determined by the scaling factor, such that $s = f/f_{max}$, where f_{max} is the maximum possible clock frequency. The supply voltage V_{dd} is determined by the alpha power law, which determines the minimum possible voltage that guarantees stable operation of the microprocessor at the frequency f.

Unlike previous DTM techniques, we have two control parameters that affect T_d: These are P_{fan} and s. Therefore, we have multiple feasible solutions which achieve the desired value of T_d. Among these feasible solutions, we wish to find the energy-optimal pair (P_{fan}, s). Note that either element of this pair (P_{fan}, s) may be located outside the feasible range: the optimal solution will then be found at the boundary of the feasible range of either variable, or both.

PROBLEM 1. *Energy-optimal cooling fan power and scaling factor for a given batch workload: minimize the energy consumption including the cooling power per cycle, which is given as*

$$E = (P_{cpu} + P_{fan})/f. \tag{12}$$

by controlling the fan speed together with supply voltage/frequency scaling. ■

The total energy is a convex function of both P_{fan} and s, as long as s is continuous, so the optimal solution pair is determined as follows:

$$(P_{fan}, s) \in \left((P_{fan}, s) \middle| \frac{\partial E}{\partial P_{fan}} = 0, \frac{\partial E}{\partial s} = 0 \right). \tag{13}$$

As an example, we derived the energy-optimal P_{fan} and s for an Intel Xeon Quadcore E7330 processorassembled with a parallel-plate finned copper heat-sink (70mm × 70mm × 50mm) and a 70mm cooling fan. Fig.5 shows how the total energy consumption of the microprocessor and cooling fan for a given workload varies with fan power and scaling factor. In this case the optimal solution which minimizes the total energy consumption is found within the feasible range of the control variables.

In practice, s has several discrete levels, so that $S = (s_1, ..., s_n)$. But, it is still quite easy to obtain the optimal feasible solution by calculating $\left((P_{fan}, s_i) \middle| \frac{\partial E}{\partial P_{fan}} = 0, s_i \in S \right)$ and then selecting the pair (P_{fan}, s) which minimizes the total energy consumption. We will describe experiments on discrete DVFS in Section 5.

4.2 A stationary periodic task with a continuous voltage and frequency domain

We will now tackle the more realistic case of joint optimization of fan speed and scaling factor. We start by considering the effect of the initial and final temperatures for a sequence of a scheduled tasks, in which the final temperature of one task becomes the initial temperature of the next task. Unfortunately, we need to make some assumptions because we cannot predict what task will be scheduled after a given task sequence.

In one previous approach, the initial temperature is assumed to be at an arbitrary value between the ambient temperature and the thermal threshold temperature, and the final temperature is forced to be lower than the initial temperature [9]. However, as shown in Fig.6, this may not minimize the energy consumption if the sequence of tasks sufficiently long. The result may either be overheating (Fig.6(a)) or overcooling (Fig.6(d)).

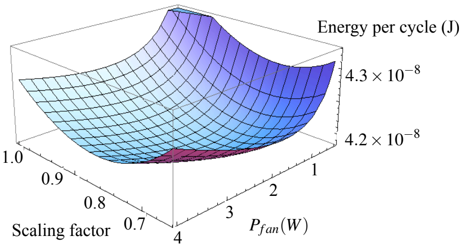


Figure 5: Total energy consumption for a batch workload on an Intel E7330 processor with voltage/frequency scaling and a cooling fan with speed control.

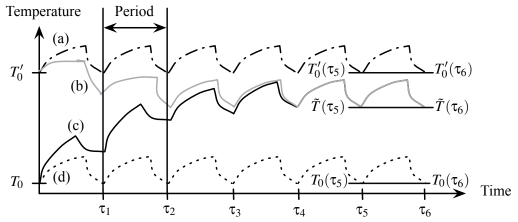


Figure 6: Effect of initial temperature on a periodic task.

Figs. 6(b) and 6(c) shows that the energy-optimal temperature at the start of period for a stationary periodic task converges to some T_s when the thermal resistance is fixed at certain value. The fan power balances the temperature-dependent leakage power at a temperature determined by the efficiency of the cooling system. Therefore, it is crucial to find the energy-optimal T_s in this case. Thus we use the following assumptions:

ASSUMPTION 1. *Uniform DVFS scheduling: a task $\mathcal{T}$ is a tuple such that $\mathcal{T} = (W_p, D)$, where W_p is the workload and D is the deadline. We assume that $\mathcal{T}$ is a periodic real-time task, where W_p is a constant which is known in advance.* ■

ASSUMPTION 2. *Slow fan dynamics: the fan is too sluggish to update its speed promptly at each period.* ■

PROBLEM 2. *Finding the energy-optimal steady-state pair (P_{fan}, s): for given values of T_a and $\mathcal{T}$, determine the energy-optimal values of P_{fan}, s, and T_s under the constraint of the thermal threshold.* ■

The energy-optimal steady-state temperature at the end of a period is obviously the same as its energy-optimal initial temperature: this follows from the definition of a steady state, which is $T_s = \tilde{T}(\tau_i) = \tilde{T}(\tau_{i+1})$. To solve Problem 2, we need to determine T_s for each pair (P_{fan}, s). From (8) and (9), we can represent T_d as a function of t, T_0, $\frac{dT_0}{dt}$, and P_{cpu}. The steady-state temperature T_s can then be found by solving the equations

$$T_{peak} = T_d(t_e, T_s, \frac{dT_d(D)}{dt}, P_{cpu}), \tag{14}$$

and

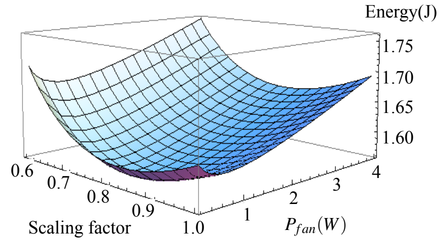


Figure 7: Total energy consumption for a periodic real-time task running on a Xeon E7330 processor with DVFS and a cooling fan with speed control.

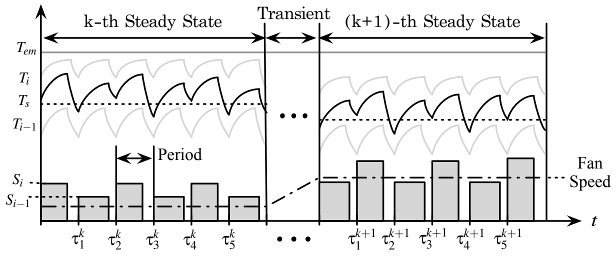

k-th Steady State
Transient
(k+1)-th Steady State
T_{em}
T_i
T_s
T_{i-1}
Period
S_i
S_{i-1}
Fan Speed
t
τ_1^k τ_2^k τ_3^k τ_4^k τ_5^k
τ_1^{k+1} τ_2^{k+1} τ_3^{k+1} τ_4^{k+1} τ_5^{k+1}

Figure 8: Energy-optimal temperature-tracking DTM.

$$T_s = T_d(D - t_e, T_{peak}, \frac{dT_d(t_e)}{dt}, P_{cpu}), \tag{15}$$

under the constraint that the peak temperature T_{peak} has to be lower than the thermal emergency temperature T_{em}. The execution time t_e is equal to $W_p/(f_{max}s)$. Finally, we can find the optimal pair (P_{fan}, s) by determining the energy consumption of each (P_{fan}, s) with its corresponding T_s, which is given by

$$E = \int_0^{t_e} P_{cpu}(T_s, T_a, s, \upsilon_f)dt + P_{fan}D. \tag{16}$$

As an example, we solved Problem 2 from the previous section with $\mathcal{T}(7 \times 10^7\text{cycles, 40ms})$. The result is shown in Fig.7.

4.3 An online stationary periodic task with a discrete voltage and frequency domain

We need to adopt our energy-efficient control policy to the discrete levels of supply voltage/frequency provided by modern DVFS technology. From the continuous solution of Problem 2 and Assumption 2, we can track the temperature that minimizes the total energy consumption. However, discrete DVFS cannot always provide the optimal solution of Problem 2, and therefore we propose a control-theoretic approach to track the energy-optimal temperature as closely as possible. The detailed control policy is as follows:

1: For given $\mathcal{T}$ and T_a, obtain the energy-optimal T_s and the corresponding value of P_{fan} using (14) to (16).
2: Among the discrete levels of s, select the two adjacent values which stabilize the temperature most closely to T_s; one of them converges above T_s (s_i in Fig.8), and the other converges below T_s (s_{i-1} in Fig.8).
3: Based on Assumption 2, fix the fan speed to achieve P_{fan}.
4: Operate the microprocessor at s_i until the system begins to overheat.
5: As soon as the system detects that the temperature is too high, adjust the DVFS level to maximum s_{oh}, such that $(s_{oh}|T_{peak}^{oh} < T_{em}, s_{oh} \in S_i)$ at the beginning of the next period, where $S_i = (s_1, \cdots, s_{i-1})$. The variable T_{peak}^{oh} represents T_{peak} when the next cycle is operated at s_{oh}.
6: As soon as the system detects that the temperature is too low, adjust the DVFS level to s_i at the beginning of the next period.
7: Repeat procedures 5 and 6 until the task finishes.

Fig.8 is an example of the profile produced by this policy.

To develop a solution for more realistic problem, We need to consider not only discrete DVFS but also the complex tasks or a temporal thermal dynamics of heat-sink. Those are beyond the scope of this paper and an approach to thermal management which took them into account would need to be the subject of further research.

5. EXPERIMENT

5.1 Total-energy-optimal (P_{fan}, s) for discrete DVFS

Table 2: Values of optimal P_{fan} to minimize total power consumption.

s	V_{dd} (V)	f (GHz)	Optimal		P_{total} (W) (1000 RPM)	P_{total} (W) (Optimal)
			P_{fan} (W)	RPM		
s_4	1.35	3.00	2.19	2699	120.15	109.01
s_3	1.30	2.63	1.58	2499	92.33	86.07
s_2	1.25	2.25	1.07	2310	68.92	65.79
s_1	1.20	1.88	0.67	2181	49.22	47.94

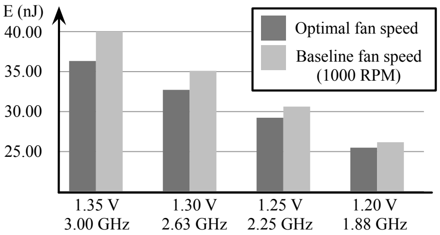
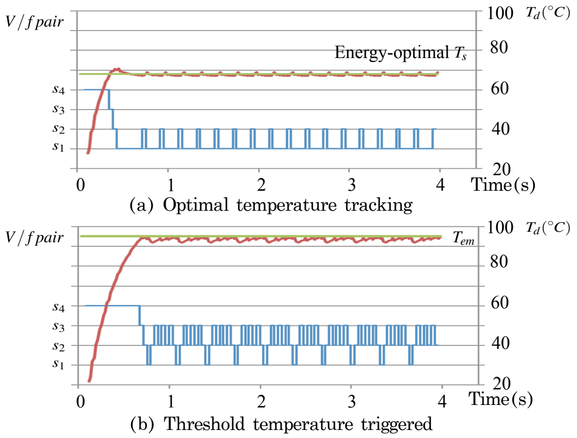


Figure 9: Total energy consumption per clock cycle against (P_{fan}, s) in a discrete DVFS.

We analytically derived the total-energy-optimal (P_{fan}, s) from (11) and (12) for discrete DVFS, as described in Section 4.1. As in Section 3, the analytical model was then checked against a HotSpot simulation. Since most practical DVFS systems support fewer than 16 scaling factors, the optimal (P_{fan}, s) can sensibly be found by an exhaustive search. We calculated the energy-optimal cooling power from (11) and (12) for given scaling factors within the feasible range. The results are summarized in Table 2, which compares the total power consumption with the fan running at its baseline speed (1000 RPM) and at the optimized speed. In determining the baseline speed we need to consider the operating range of the processor and the physical constraint of the fan. Usually, the fan is running as slow as possible within the feasible range to reduce noise and vibration. We set the baseline speed at 1000 RPM considering those factors.

Fig.9 illustrates the variation in energy consumption over each clock cycle as (P_{fan}, s) changes. It turns out that the total energy requirement can be reduced by 9.3% if we use the faster fan speed, rather than the baseline speed, even though the latter is adequate to keep T_d below the thermal threshold. At the faster speed the die temperature drops by 9°C to 26°C .

To illustrate the difference between our approach and conventional DTM, we performed a further simulation based on (16). Figs. 10(a) and 10(b) show how our method maintains the die temperature at a lower value than the thermal threshold target used by a conventional DTM policy. In this case an operating temperature of 68°C minimizes the total energy consumption, and running the system at this temperature uses 8.2% less energy than



(a) Optimal temperature tracking

(b) Threshold temperature triggered

Figure 10: Optimal temperature tracking scheduling and threshold temperature triggered scheduling.

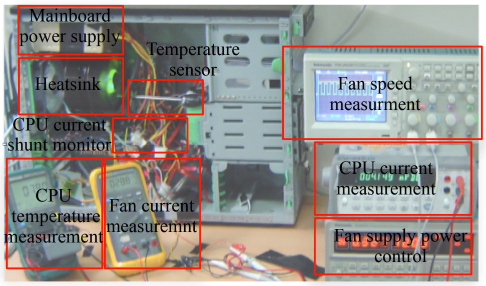


Figure 11: Measurement setup.

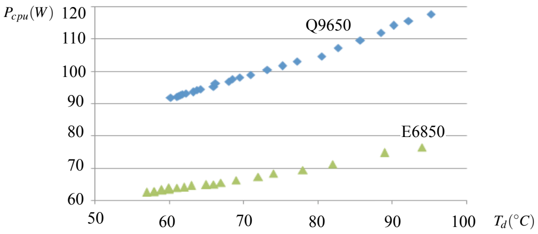


Figure 12: Power variation with temperature.

running it at the threshold temperature of 95°C.

5.2 Real device measurements

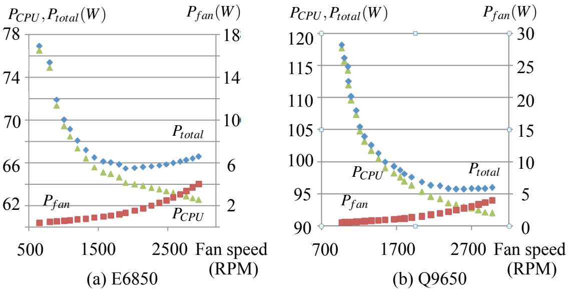
While our analyses suggest that energy-optimal DTM can save a significant amount of overall power, we wished to confirm that the idea is applicable to real systems. Therefore we measured the power consumption of two real processors, the Intel E6850 and Q9650, at different temperatures to confirm the temperature-dependent leakage power and the cooling power. The two processors were assembled with a Zalman CNPS-9700 NT heat-sink in a PC. We varied the fan speed to change the running temperature of each microprocessor. The measurement setup is shown in Fig.11. We used high-precision equipment including an Agilent A34401 multimeter, a Tektronix TDS2024B oscilloscope, a TX3 multimeter, a PS2521G power supply, a Fluke 87III multimeter, and a K-type temperature sensor to measure the fan power and speed, the ambient temperature, and the microprocessor power supply current. We determined the fan speed from the encoder pulse output of the fan motor, while the fan supply current is measured. We use the Prime95, which is a stress-test tool based on fast fourier transforms as the microprocessor workload. We read the die temperature directly from the on-chip thermal sensor in the microprocessor.

As shown in Fig.12, we observed that the power consumption of the E6850 and Q9650 microprocessors with the Prime95 workload increased by up to 18% and 22% as the die temperature increased. This demonstrates that the curve of total power consumption against fan speed is convex, as shown in Figs. 13 (a) and (b).

To estimate the amount of energy that could be saved using the proposed control-theoretic approach, we applied the parameters from these results to the example problem in Sect. 5.1. We extracted the coefficients of the power model and the heat-sink model from the measurements shown in Figs. 12 and Figs. 13. We used the same task parameters presented in Sect. 5.1 with value of (P_{fan}, s) for the real processors. We were thus able to predict a total energy saving of 6.5% and 17.6% for the E6850 and Q9650 respectively, when compared with the energy requirement at the baseline fan speed.

6. CONCLUSIONS

Conventional thermal management techniques aim at avoiding a thermal emergency while maximizing some performance metric, typically throughput. This approach to thermal management does not try to minimize total power consumption, which is rapidly becoming a much more urgent objective.



(a) E6850

(b) Q9650

Figure 13: E6850 and Q9650 power versus fan speed.

We have introduced the idea of using the thermal resistance of a forced-convection heat-sink as a control variable, to be used in the same way as the voltage and frequency of a microprocessor. We have proposed a new thermal management method that explicitly tracks the energy-optimal temperature as closely as possible with a given workload, making the best trade-off between cooling power and temperature-dependent leakage power. Experimental results show that using the optimal fan speed of the fan can reduce the total power consumption by up to 17.6%, and that scheduling a typical set of tasks based on the optimal steady-state temperature can achieve an overall 8.2% reduction in energy consumption compared with conventional DTM.

7. REFERENCES

[1] D. Brooks and et. al., “Dynamic thermal management for highperformance microprocessors,” in *HPCA’01*, pp. 171–182, 2001.

[2] J. Srinivasan and et. al., “Predictive dynamic thermal management for multimedia applications,” in *ICS’03*, pp. 109–120, 2003.

[3] A. Merkel and et. al., “Event-driven thermal management in SMP systems,” in *TACS’05*, pp. 1659–1664, 2005.

[4] R. Jejurikar and et. al., “Leakage aware dynamic voltage scaling for real-time embedded systems,” in *DAC’04*, pp. 275–280, 2004.

[5] W. Liao and et. al., “Temperature and supply voltage aware performance and power modeling at microarchitecture level,” *IEEE Trans. on CAD*, vol. 24, no. 7, 2005.

[6] L. Yuan and et. al., “Temperature-aware leakage minimization techniques for real-time systems,” in *ICCAD’06*, pp. 761–764, 2006.

[7] W. Huang and et. al., “HotSpot: a compact thermal modeling methodology for early-stage VLSI design,” *IEEE Trans. on VLSI*, vol. 14, pp. 501–513, 2006.

[8] R. Rao and et. al., “Performance optimal processor throttling under thermal constraints,” in *CASES’07*, pp. 257–266, 2007.

[9] S. Zhang and et. al., “Approximation algorithm for the temperature aware scheduling problem,” in *ICCAD’07*, pp. 281–288, 2007.

[10] R. J. Moffat, “Modeling air-cooled heat sinks as heat exchangers,” in *Semi-Therm’07*, pp. 200–207, 2007.

[11] P. Teertstra and et. al., “Analytical forced convection modeling of plate fin heat sink,” in *Semi-Therm’99*, pp. 34–41, 1999.

[12] K. Azar and et. al., “How much heat can be extracted from a heat sink,” *Electronics Cooling*, 2003.

[13] Y. Liu and et. al., “Accurate temperature-dependent integrated circuit leakage power estimation is easy,” in *DATE’07*, pp. 1526–1531, 2007.

[14] K. Skadron and et. al., “Control-theoretic techniques and thermal-RC modeling for accurate and localized dynamic thermal management,” in *HPCA’02*, pp. 17–28, 2002.

[15] M. Pedram and et. al., “Thermal modeling, analysis, and management in VLSI circuits: principles and methods,” *Proc. of the IEEE*, vol. 94, pp. 1487–1501, 2006.

[16] W. Huang and et. al., “Accurate, pre-RTL temperature-aware design using a parameterized, geometric thermal model,” *IEEE Trans. on Computers*, vol. 57, no. 9, pp. 1277–1288, 2008.

[17] D. Brooks and et. al., “Wattch: a framework for architectural-level power analysis and optimizations,” in *ISCA’00*, pp. 83–94, 2000.

[18] C. Isci and et. al., “Runtime power monitoring in high-end processors: methodology and empirical data,” in *MICRO’03*, pp. 93–104, 2003.

[19] *Standard Performance Evaluation Corporation, CPU2000.* http://www.spec.org/osg/cpu2000.

[20] K. Lee and et. al., “Using performance counters for runtime temperature sensing in high-performance processors,” in *IPDPS’05*, pp. 232–239, 2005.