Power-Aware Real-Time Scheduling: Models, Open Problems, and Practical Considerations

Nathan Fisher*

1 Introduction

Power-related issues have received considerable research attention from the real-time community in the past decade. In this abstract, we introduce a recent model and set of assumptions made in the recent real-time literature on energy and thermal issues; suggest two high-level open problems for power-aware real-time scheduling; and discuss practical considerations and constraints for proposed solutions.

2 Models

2.1 Workload Model

Throughout this abstract, we will assume that the real-time workload is specified by the widely-used *sporadic task model* [2]. A sporadic task $\tau_i = (C_i, T_i, D_i)$ is characterized by a *worst-case execution requirement* C_i, a *(relative) deadline* D_i, and a *minimum inter-arrival separation* T_i, which is, for historical reasons, also referred to as the *period* of the task. Such a sporadic task generates a potentially infinite sequence of jobs, with successive job-arrivals separated by at least T_i time units. Each job has a worst-case execution requirement equal to C_i and a deadline that occurs D_i time units after its arrival time. A job-arrival sequence is said to be *legal* if each task's minimum inter-arrival constraint on successive job arrivals is satisfied and no job executes for more than its worst-case execution requirement. Furthermore, we will assume that jobs are arbitrarily preemptible without penalty. A *sporadic task system* $\tau \stackrel{\text{def}}{=} \{\tau_1, \ldots, \tau_n\}$ is a collection of n such sporadic tasks. For a given scheduling algorithm $\mathcal{A}$, we say that τ is $\mathcal{A}$*-schedulable*, if τ always meets it deadline when scheduled according to algorithm $\mathcal{A}$ (under any legal job arrival sequence).

2.2 Power Model

A large number of real-time papers on power-aware scheduling focus almost exclusively on the CPU. A common assumption is that the CPU has *dynamic-voltage scaling* (DVS) capabilities. A DVS processor may dynamically increase or decrease the voltage to change the power-consumption or heat-generation of the system. In this section, we briefly review a common power model assumed in many real-time systems research papers; the notation and model formulation used in this section very closely matches the model used in a recent paper by Wang et al. [5].

* fishern@cs.wayne.edu. Department of Computer Science, Wayne State University, Detroit, MI USA.

Dagstuhl Seminar Proceedings 10071
Scheduling
http://drops.dagstuhl.de/opus/volltexte/2010/2539

§**CPU Power-Consumption.** A recent power model (e.g., see [3, 5]) divides power consumption of a DVS processor into two parts: *speed-dependent* and *speed-independent* portions. The speed-dependent power consumption (as a function of time) arises from the charging and discharging of gates and can be quantified on a CMOS circuit by

$$P_{\mathrm{dep}}(s,t) \stackrel{\text{def}}{=} C_{\mathrm{ef}} V_{\mathrm{dd}}^2(t) s(t) \tag{1}$$

where $s(t)$ and $V_{\mathrm{dd}}(t)$ are the speed and supplied voltage (respectively) for the processor at time t. The term C_{ef} is a positive constant describing the effective switch capacitance of the circuit. Furthermore, it is typically assumed that $s(t) = \kappa_v \frac{(V_{\mathrm{dd}}(t) - V_{\mathrm{thresh}})^2}{V_{\mathrm{dd}}(t)}$ where κ_v is a (positive) hardware-design-specific constant and V_{thresh} is the voltage threshold. The above equation can be simplified to

$$P_{\mathrm{dep}}(s,t) = h s^{\gamma}(t), \tag{2}$$

where h and γ are positive constants with $\gamma \leq 3$.

The speed-independent power consumption is due to leakage current which can be dependent upon the temperature of the CPU [1]. If $\Theta(t)$ is the absolute temperature of the CPU at time t, one proposed model of speed-independent power consumption is

$$P_{\mathrm{ind}}(\Theta,t) = \delta\Theta(t) + \rho \tag{3}$$

where δ and ρ are both processor-specific constants. Thus, the total power consumption of the CPU may be modeled by

$$P(s,\Theta,t) = P_{\mathrm{dep}}(s,t) + P_{\mathrm{ind}}(\Theta,t) = h s^{\gamma}(t) + \delta\Theta(t) + \rho. \tag{4}$$

The energy consumed over an interval $[t_1, t_2]$ is thus equal to $\int_{t_1}^{t_2} P(s,\Theta,t)dt$.

§**CPU Cooling.** A majority of research on real-time thermal-aware systems assumes that the cooling process can be approximately modeled by Fourier's Law. Given a fixed ambient temperature Θ_a, the rate of change may be characterized by

$$\Theta'(t) = \hat{\alpha} P(s,\Theta,t) - \hat{\beta}(\Theta(t) - \Theta_a) \tag{5}$$

where $\hat{\alpha}$ and $\hat{\beta}$ are the coefficients for heating and cooling, respectively. The above equation can be reformulated to

$$\Theta'(t) = \alpha s^{\gamma}(t) - \beta\Theta(t) + \sigma \tag{6}$$

where α, β, and σ are positive constants dependent upon $\hat{\alpha}$, $\hat{\beta}$, ρ, and Θ_a.

Instead of directly considering the absolute temperature $\Theta(t)$, it is often easier to work with the *adjusted temperature* $\theta(t)$:

$$\theta(t) \stackrel{\text{def}}{=} \frac{\Theta(t)}{\alpha} - \frac{\sigma}{\alpha\beta}. \tag{7}$$

Thus, the power function can be reformulated,

$$P(s,\theta,t) = h s^{\gamma}(t) + \delta\alpha\theta(t) + (\rho + \frac{\delta\sigma}{\beta}). \tag{8}$$

and the (adjusted) temperature at time t may be determined by, given an initial temperature $\theta(t_0)$ at t_0,

$$\theta(t) = \int_{t_0}^{t} s^{\gamma}(x) e^{-\beta(t-x)} dx + \theta(t_0) e^{-\beta(t-t_0)}. \tag{9}$$

3 Open Problems

In this section we present two (high-level) open theoretic questions for power-aware scheduling under the sporadic task model. In the next section, we add further complexity to the problems by discussing some practical constraints and considerations of power-aware systems.

Problem 1 (Peak Temperature Minimization) *Given a sporadic task system τ and scheduling algorithm $\mathcal{A}$, determine function s which minimizes peak system temperature subject to the the constraints that τ is $\mathcal{A}$-schedulable.*

Problem 2 (Energy Minimization with Temperature as a Constraint) *Given a sporadic task system τ, scheduling algorithm $\mathcal{A}$, and a temperature threshold T_{thresh}, determine function s which minimizes energy consumption subject to the constraints that τ is $\mathcal{A}$-schedulable and the CPU temperature never exceeds T_{thresh}.*

4 Practical Considerations

There are numerous practical considerations that must be considered for a proposed power-aware scheme to be applicable. In fact, it has been observed (e.g., see [4]) that many previously suggested power-aware schemes that purport to be energy-efficient may actually increase energy consumption over non-power-aware approaches! In the following subsections, we briefly outline some practical complexities which may constrain solutions obtained for Problems 1 and 2.

4.1 Speed Function Considerations

In a system, the speed function may be constrained by the following factors:

- **Piecewise, Finite Speed Schedule:** In order for the speed function to be realizable in an actual system, the s function should either be efficiently computable online or storable in a table residing in main memory. If is stored in a table in main memory, the s function would likely have to be a piecewise, periodic function with a finite number of discontinuities. Furthermore, due to system clock granularity, the time interval length between speed changes may be restricted to be larger than some $\Delta > 0$.

- **Finite Set of Achievable Speeds:** Some results on power-aware scheduling assume that the speeds of the processor are drawn from the range $[0, \infty)$. Realistically, most systems have both an upper bound $s_{\max}$ and a lower bound $s_{\min}$ on speed from which $s \in [s_{\max}, s_{\min}]$. Furthermore, in many systems there are only a finite number of achievable speed modes $\mathcal{S} = \{s_1, s_2, \ldots, s_r\}$. Speeds not in the set $\mathcal{S}$ might not be supported by the processor.

- **Non-Linear Program Runtimes:** Schmidt et al. [4] have observed program runtimes do not necessarily decrease linearly with respect to increased CPU frequency. Thus, the function s used in the power function $P(s, \theta, t)$ may not be the same function which determines the execution time of each job generated by the task system; furthermore, different segments of the program may exhibit different execution behavior for the same speed. For example, consider a job consisting of a sequential code segment A followed by a code segment B. At speed s_1, assume that the segment

A takes x time units and B takes y time units. It is entirely possible (due to program I/O or cache misses) that when the CPU executes at $s_2 = 5s_1$ that segment A takes $x/5$ time and segment B takes $y/2$ time; in this case, even though the processor is executing five times faster, the realizable speed-up is strictly less than five.

- **Speed Transition Overheads:** During a transition from one speed mode s_i to another s_j, the CPU may be unavailable for execution of jobs for a $\delta(s_i, s_j)$-length transitory interval.

4.2 Non-CPU Power Considerations

The CPU is not the only consumer of power in the system or source of heat. For instance, many modern memory controllers also offer power management features. However, power management in memory systems may be complicated due to the fact that some types of memory (e.g., DRAM) requires periodic refreshes to avoid corruption; furthermore, putting some memory controllers in a low-power state may lead (counter-intuitively) to higher energy consumption, as an entire memory refresh is required when the controller becomes active.

References

[1] W. LIAO, L. HE, AND K. M. LEPAK. Temperature and supply-voltage aware performance and power modeling at microarchitecture level, *IEEE Transactions on CAD of Integrated Circuits and Systems*, vol. 24, no. 7, pp. 1042–1053, 2005.

[2] A. K. MOK. *Fundamental Design Problems of Distributed Systems for The Hard-Real-Time Environment*. PhD thesis, Laboratory for Computer Science, Massachusetts Institute of Technology, 1983. Available as Technical Report No. MIT/LCS/TR-297.

[3] J. M. RABAEY, A. CHANDRAKASAN, AND B. NIKOLIC. *Digital Integrated Circuits*. Prentice Hall, 2002, 2nd Edition.

[4] D. SCHMIDT AND N. WEHN A Review of Common Belief on Power Management and Power Consumption. White paper, Technische Universität Kaiserslautern. April, 2009. Available at `http://ems.eit.uni-kl.de/uploads/tx_uniklwehn/myths.pdf`.

[5] S. WANG, J.J. CHEN, Z. SHI, AND L. THIELE. Energy-Efficient Speed Scheduling for Real-Time Tasks under Thermal Constraints. In *Proceedings of the 15th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications*, pages 201–209, Beijing, China, August 2009. IEEE Computer Society Press.