

Energy minimization in the speed scaling model

Pankaj Agarwa, IVivek Kumar Gupta

Research Scholar (Dept. of Electronics & Comm.) Asstt. Professor (Dept. of Electronics & Comm.)

Dehradun Institute of technology Dehradun Institute of technology

Dehradun Dehradun

panagr123@gmail.com vivek_gupta79@rediffmail.com

Abstract—Energy use of computer communication systems has quickly become a vital design consideration. One effective method for reducing energy consumption is dynamic speed scaling, which adapts the processing speed to the current load. We study network optimization that considers energy minimization as an objective. Studies have shown that mechanisms such as speed scaling can significantly reduce the power consumption of telecommunication networks. The dynamic voltage and frequency scaling in CPUs is an example of adjusting a device's control variable to trade off power consumption and performance. This idea of energy optimization through speed control has been subsequently applied to other components of electronic systems such as disk drives and wireless transceivers.

A processor is equipped with variable clock frequencies (speedy) feature and is used to schedule a set of given jobs with deadlines. Each speed change involves time/energy overhead and also impacts negatively the processors lifetime reliability. Based on above facts, problem of “energy aware scheduling, considering the number of speed changes and cost associated due to number of speed changes” is studied. Designing speed schedules to satisfy all jobs deadline and at the same time optimize the energy consumption and total cost involved due to speed changes. We develop algorithms that work close to the optimal and analyze its time complexities.

Index terms: dynamic voltage scaling, energy management, real time scheduling, speed scaling.

I. INTRODUCTION

POWER management is increasingly important in computer communication systems. Not only is the energy consumption of the internet becoming a significant fraction of the energy consumption of developed countries, but cooling is also becoming a major concern. Consequently, there is an important tradeoff in modern system design between reducing energy use and maintaining good performance.

There is an extensive literature on power management, reviewed in [2] and [4]. A common technique, which is the focus of the current paper, is dynamic speed scaling. This dynamically reduces the processing speed at times of low workload, since processing more slowly uses less energy per operation. This methodology is adopted in many chip designs. Now days, speed scaling has been proposed for many network devices, such as switch fabrics [11], OFDM modulation clocks, etc.

Energy management remains an important problem for computer systems, and in particular, for real time embedded systems. Here the target is to design and analyse algorithms for maximizing energy usage efficiency with the consideration of system performance requirements.

The rapid advance of processor design technology provides fast computing. Now a days processors like Intel *SpeedStep* and AMD *PowerNOW*, have been equipped with a feature to vary the clock frequency dynamically. The operating system is able to adjust the processor's clock frequency (speed) on the fly along with the supply voltage to execute jobs and reduce consumption at lower speeds. This functionality is called *speed scaling* and also *dynamic voltage scaling*. Speed scaling is expected to satisfy some quality of service measures as well as to reduce overall energy cost, by manipulating modern processors' multiple speeds. Under the speed $f(t)$ at time t , the processor consumes energy $e(f(t))$ per unit time and the function $e(\cdot)$ is assumed to be convex. The objective is to construct a schedule satisfying all jobs' deadline constraints and to minimize the total energy consumption, which is defined as $\int e(f(t))dt$.

Existing studies considered energy minimization through speed scaling without much attention to the impact of speed changes. Such changes typically involve time and energy overhead. Moreover recent studies indicate that the lifetime reliability of a CMOS circuit is directly related to the number and span of speed changes.

Hardware failures, such as cracks and fatigue failures are created not by sustained high temperatures but by the repeated heating and cooling of sections of the processor. This phenomenon is called as thermal cycling.

Using MTTF (Mean Time To Failure) to describe the expected processors life, the following Coffin Manson formula is used to characterize a processor's lifetime reliability:

$$MTTF \propto \frac{1}{C_0 (\Delta T_{mp} - \Delta T_0) q_x}, \quad (1)$$

Where C_0 is a material dependent constant, ΔT_{mp} is the entire temperature cycle range of the device ΔT_0 is the portion

of the temperature range in the elastic region, q is the Coffin-Manson exponent, and χ is frequency (number of occurrences per unit time) of thermal cycles.

Above equation clearly indicates that an algorithm which frequently changes the processor's speed results in large χ and ΔT_{mp} . Thus a schedule that frequently changes speed may result in large temperature cycle range and therefore affect the processor's life time reliability adversely. Simulations have confirmed that various speed scaling energy aware policies have different impacts on processor's reliability in terms of MTTF. The number of speed changes (x in equation 1) is a critical factor in determining a processor's reliability under the thermal cycling phenomenon.

The main purpose is to undertake a theoretical investigation of speed scaling algorithms by considering the cost and number of speed changes. The results remain valid for arbitrary convex energy consumption functions $e(\cdot)$ with $e(0)=0$. The requirement is not that $e(\cdot)$ should be some closed function form; it may be given by various closed formulas in different frequency ranges. In this paper, we assume that only the CPU's clock frequency can be adjusted.

II. MODELS AND PROBLEM DEFINITION

We consider a single processor setting. The processor has variable clock frequencies (speeds). Under a speed f , the processor consumes energy $e(f)$ per unit time and assume that the function $e(\cdot)$ is convex and $e(0)=0$. We note that in scaling speeds, the processor's frequency and supply voltage are both adjusted (dynamic voltage/frequency scaling). Hence, in rest of the paper we will understand that frequency/speed changes always involve the corresponding voltage change.

Consider a single processor setting. The processor has variable clock frequencies (*speeds*). Under a speed f , the processor consumes energy $e(f)$ per unit time and assume that the function $e(\cdot)$ is convex and $e(0)=0$.

In scaling speeds, the processor's frequency and supply voltage are both adjusted (dynamic voltage/frequency scaling). Rest of the report is to understand that frequency/speed changes always involve the corresponding voltage change.

Consider a set of n real time jobs $J = \{J_1, J_2, \dots, J_n\}$. Each job J_i has a release time $r_i \in \mathbb{R}^+$, a processing time (also called as worst execution time), $p_i \in \mathbb{R}^+$, and a deadline $d_i \in \mathbb{R}^+$. Under the speed f , it takes time p_i/f to complete the job J_i . Considering the pre-emptive scheduling and assuming the cost of pre-emption is negligible.

The objective in this study is to design scheduling algorithms to finish all the jobs before their deadlines by considering the objectives of minimizing the energy consumption, as well as the number and cost of speed changes.

Definition 1 (Speed schedule): A speed schedule can be viewed as piecewise constant curve, specifying the speed the processor employs in each time interval. Assume that the CPU changes speed m times during the execution. Then the speed

each of these intervals. Let the m time intervals be:

$$I_1 := (t_0, t_1], I_2 := (t_1, t_2], \dots, I_m := (t_{m-1}, t_m].$$

The triple (t_{i-1}, t_i, s_i) corresponds to the i^{th} time interval $I_i = (t_{i-1}, t_i]$ ($0 < i < m$ and $t_0 = 0$) in which the processor runs at a speed $s_i \geq 0$. Hence the speed scheduler Ψ is specified by m triples:

$$\Psi := \{(t_0, t_1, s_1), (t_1, t_2, s_2), \dots, (t_{m-1}, t_m, s_m)\}$$

Figure below illustrates an example schedule. The schedule employs 4 distinct speeds f_1, f_2, f_3, f_4 in 6 time intervals, where $s_1 = s_4 = f_1, s_2 = s_6 = f_4, s_3 = f_2$, and $s_5 = f_3$.

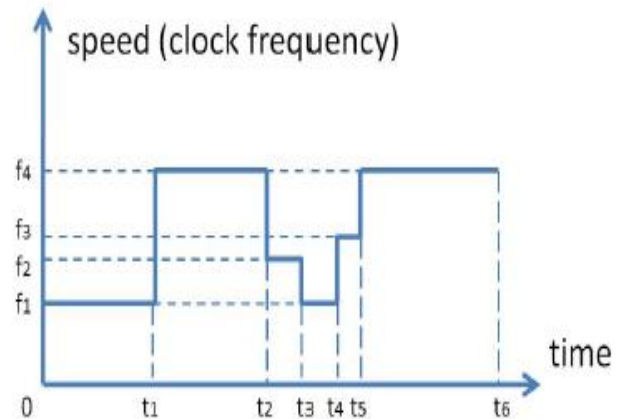


Figure 1. A piecewise curve describing the time intervals and the processor's speed in each interval.

We can call t_i a *speed switching point*. Without loss of generality, we can assume that the processor is in *idle* state initially at time 0 ($s_0 = 0$) and gets back to the idle state after processing all the jobs ($s_{m+1} = 0$). Thus a schedule with m time intervals has $m+1$ speed switching points $t_0, t_1, t_2, \dots, t_m$.

The total energy consumption of such a schedule is calculated as:

$$E^\Psi = \sum_{i=1}^m e(s_i)(t_i - t_{i-1})$$

For the example in figure:

$$\begin{aligned} E^\Psi &= e(f_1)(t_1 - 0) + e(f_4)(t_2 - t_1) + e(f_2)(t_3 - t_2) + e(f_1)(t_4 - t_3) \\ &\quad + e(f_3)(t_5 - t_4) + e(f_4)(t_6 - t_5) \\ &= e(f_1)(t_1 + t_4 - t_3) + e(f_4)(t_2 - t_1 + t_6 - t_5) + e(f_2)(t_3 - t_2) \\ &\quad + e(f_3)(t_5 - t_4). \end{aligned}$$

To incorporate the penalty of changing clock frequencies, consider that each speed change from the frequency s_i (in interval I_i) to the frequency s_{i+1} (in the interval I_{i+1}) involves a cost $c_{i,i+1} \in \mathbb{R}^+$ ($c_{i,i+1}$ may, for example, reflect the speed change's negative impact on the processor's lifetime reliability).

Assuming that the cost of a speed change is a convex function of the difference between the previous and the new speed values. For instance, switching from f_4 to f_1 may be more costly than switching from f_4 to f_3 , if $f_1 < f_3 < f_4$. Consequently, the function $c(\cdot)$ is convex and $c_{i,i+1}$ is the value

of $c(\cdot)$ for $|s_i - s_{i+1}|$.

$$c_{i,i+1} := c(|s_i - s_{i+1}|), \text{ where } s_0 = s_{m+1} = 0 \quad (2)$$

We now present the formulation of two optimization problems.

Problem1: Minimizing sum of energy consumption and costs associated due to speed changes.

Let E^Ψ denote the total energy consumed by the schedule Ψ to complete the set of jobs \mathbf{J} by their deadlines. Assume Ψ has m time intervals. The total cost associated with all the clock speed changes during this schedule is $\sum_{i=0}^m c_{i,i+1}$ where s_0 and s_{m+1} are defined as 0 (assumed above). In this problem, the aim is to minimize $E^\Psi + \beta \sum_{i=0}^m c_{i,i+1}$, where β is a given constant. After normalizing $c_{i,i+1}$, we can remove β and formulate the problem as

$$\text{Min.} (\sum_{i=1}^m e(s_i)(t_i - t_{i-1}) + \sum_{i=0}^m c'_{i,i+1}),$$

$$\text{where } s_0 = s_{m+1} = 0 \text{ and } c'_{i,i+1} = c_{i,i+1} / \beta.$$

Problem2: Minimizing sum of energy consumption under a fixed number of speed changes.

Let E^Ψ and m denote the total energy consumed and the total no of speed changes in the schedule Ψ to complete the jobs in \mathbf{J} by their deadlines, respectively. Let M be the upper bound on the number of speed changes. The objective is to minimize E^Ψ subject to $m \leq M$. That is,

$$\text{Min. } \sum_{i=1}^m e(s_i)(t_{i-1}), \text{ subject to } m \leq M.$$

Problem2 considers the number of speed changes as a constraint.

III. ALGORITHMS AND ANALYSIS

In the following, we present algorithmic solutions for the two problems discussed above. We analyse their performance as well.

The algorithms have no restrictions over the number of processor's speed changes. The models discussed above have their own algorithmic challenges.

A. The formulation for minimizing sum of energy consumption and costs incurred due to speed changes is discussed here.

Assuming that the schedule Ψ has m time intervals $I_i = (t_{i-1}, t_i]$ ($1 \leq m$) and within each interval I_i , the processor keeps running at constant speed $s_i \geq 0$. The system does not consume any energy after finishing the last job.

The objective is

$$\text{Min.} (\sum_{i=1}^m e(s_i)(t_i - t_{i-1}) + \sum_{i=0}^m c_{i,i+1}) \quad (3)$$

where $c_{i,i+1}$ is scaled by a factor β from its definition as discussed before.

Defining OPT as an optimal algorithm minimizing the sum of energy consumption and the costs incurred due to speed

changes. The job is to determine all the candidate values that t_i in above equation (3) can take place. The function $c(\cdot)$ is assumed to be convex function, hence determining the optimal schedule's speed switching points heavily depends on the function $c(\cdot)$ itself.

It is possible that the processor will need to change its speed in each time slot to optimize equation(3). Instead of designing algorithm for some specific functions $c(\cdot)$, a large class of algorithms called *event driven DVS* (dynamic voltage scaling) is discussed. The purpose of introducing an event driven DVS algorithm for the problem is to show that there exists an optimal convex programming based solution, and this solution's framework can be proved to generate optimal solutions for other future problems.

Event Driven DVS Algorithms: For event driven DVS algorithms, speed changes (speed switching points) only happen at jobs' release times and /or deadlines.

Event Driven DVS algorithms have the distinct advantage of keeping the run time overhead due to DVS low, as opposed to DVS algorithms that requires speed change at arbitrary points during execution :The CPU scheduler, That is invoked at task release times and deadlines, can also regulate the frequency according to the pre-determined speed schedule during the same invocation. As a result, the optimality of event driven DVS algorithms will prove very useful in practice.

Let $\mathbf{J} = \{J_1, J_2, \dots, J_n\}$ denote the n jobs to be scheduled. A job J_j is represented by a triple (r_j, d_j, p_j) .

Let $\mathbf{R} = \{r_1, r_2, \dots, r_n\}$ and

$\mathbf{D} = \{d_1, d_2, \dots, d_n\}$.

Let $\mathbf{Z} = \mathbf{R} \cup \mathbf{D}$ to denote the union of all the release times and deadlines of jobs. Note that $|\mathbf{Z}| = |\mathbf{R} \cup \mathbf{D}| \leq |\mathbf{R}| + |\mathbf{D}| \leq 2n$. Sort all the values in \mathbf{Z} in increasing order and index them as z_1, z_2, \dots, z_n , where $n' \leq 2n$. Without loss of generality, assume $z_1 = 0$. Before the last deadline z_n , the time range is divided into $n' - 1$ non overlapping intervals $(z_i, z_{i+1}]$, $\forall 1 \leq i \leq n' - 1$. We name the interval $T_{i,i'}$:= $(z_i, z_{i'}]$ as a scheduling interval. For each scheduling interval $T_{i,i'}$, compute its corresponding work load by a variable $W_{i,i'}$ and processing capacity as $P_{i,i'}$, given the speed s_i assumed for each interval $(z_{i-1}, z_i]$.

$$P_{i,i'} = \frac{\sum_j p_j}{z_{i'} - z_i}, \text{ where } z_i < r_j < d_j < z_{i'}$$

$$W_{i,i'} = \sum_{l=i+1}^{i'} s'_l, \quad i < l < i'$$

Where s_i is the speed variable to denote at which speed the processor runs in the interval $(z_{i-1}, z_i]$. In order to complete all the jobs by their deadlines, the processing capacity should be at least the workload requirement for each time interval.

The remaining task is to determine s'_i such that all the jobs in \mathbf{J} can be finished by their deadlines and the objective

$$\sum_{l=2}^{|Z|} e(s'_l)(z_l - z_{l-1}) + \sum_{l=2}^{|Z|+1} c(|s'_l - s'_{l-1}|) \text{ is minimized.}$$

The problem is formulated using a convex program as below :

$$\text{Min.} \sum_{l=2}^{|Z|} e(s'_l) (z_l - z_{l-1}) + \sum_{l=2}^{|Z|+1} c(|s'_l - s'_{l-1}|)$$

$$\text{Subject to } W_{i,i'} \geq P_{i,i'}, \forall 1 \leq i \leq |Z|$$

Studied above, is the analysis of correctness and time complexity for the algorithm that computes an event-driven DVS schedule for the problem.

For a given set of real time jobs with known processing times, pre-emptive EDF is optimal in the sense that any feasible job set can be also scheduled in a feasible manner by the EDF policy.

ALGORITHM

1. Let $J = \{ J_1, J_2, \dots, J_n \}$; i.e. 'n' jobs to be scheduled.
2. $J_j = \{ r_j, d_j, p_j \}$; where 'j' is any general term for (1, 2, ..., j, ..., n) i.e. in between 1 and n.
3. $R = \{ r_1, r_2, \dots, r_j, \dots, r_n \}$ and $D = \{ d_1, d_2, \dots, d_j, \dots, d_n \}$
4. $Z = R \cup D$
5. As there are n values in r and n values in d, hence, $|Z| \leq |R \cup D| \leq |R| \cup |D| \leq 2n$
6. Sorting in increasing order i.e. z_1, z_2, \dots, z_n , where $n' \leq 2n$. As there is union of two i.e. R and D, hence the values are doubled i.e. from (1 to 2n) or (1 to n') as $n' \leq 2n$.
7. Now divide these intervals into $(n'-1)$ non-overlapping intervals i.e. (z_i, z_{i+1}) , $T_{i,i'}$ = scheduling interval.
8. There are at most nC_2 scheduling intervals.
9. For each scheduling interval, we can calculate $P_{i,i'} = \frac{\sum_j p_j}{z_{i'} - z_i}$, where $z_i < r_j < d_j < z_{i'}$. [Processing capacity]

$$W_{i,i'} = \sum_{l=i+1}^{i'} s'_l, i < l < i'. \quad [\text{Workload}]$$

10. Minimize the energy using the formula :

$$\text{Min.} \sum_{l=2}^{|Z|} e(s'_l) (z_l - z_{l-1}) + \sum_{l=2}^{|Z|+1} c(|s'_l - s'_{l-1}|)$$

$$\text{subject to } W_{i,i'} \geq P_{i,i'}, \forall 1 \leq i \leq |Z|$$

11. Repeat a schedule running jobs in order using different speeds $\{s'_l\}$.

B. Minimizing energy consumption under limited number of speed changes.

Assuming M be the upper limit on the number of speed changes that a speed schedule Ψ is allowed to schedule jobs, we present algorithm for the problem. Similar to the problem1 we sort all the values in Z in increasing order and index them as z_1, z_2, \dots, z_n , where $n' \leq 2n$. Thus, the whole time s_i divided into $n' - 1$ non overlapping intervals (z_i, z_{i+1}) in which the processor runs possible positive speeds, $\forall 1 \leq i \leq n' - 1$. The

interval $T_{i,i'} := (z_i, z_{i'})$ ($i' \geq i+1$) is a scheduling interval .

There are at most $\binom{n'}{2} = \frac{n'(n'-1)}{2}$ such scheduling intervals .

For each scheduling interval $T_{i,i'}$, we can calculate its corresponding workload $W_{i,i'}$, and its processing capacity $P_{i,i'}$ as earlier, given the speeds assumed for each interval $(z_{i-1}, z_i]$.

The remaining task is to determine s'_l such that all the jobs in J can be finished by their deadlines and the objective is to bound the number of speed changes by M.

ALGORITHM

1. Let $J = \{ J_1, J_2, \dots, J_n \}$; i.e. 'n' jobs to be scheduled.
2. $J_j = \{ r_j, d_j, p_j \}$; where 'j' is any general term for (1, 2, ..., j, ..., n) i.e. in between 1 and n.
3. $R = \{ r_1, r_2, \dots, r_j, \dots, r_n \}$ and $D = \{ d_1, d_2, \dots, d_j, \dots, d_n \}$
4. $Z = R \cup D$
5. As there are n values in r and n values in d, hence, $|Z| \leq |R \cup D| \leq |R| \cup |D| \leq 2n$
6. Sorting in increasing order i.e. z_1, z_2, \dots, z_n , where $n' \leq 2n$. As there is union of two i.e. R and D, hence the values are doubled i.e. from (1 to 2n) or (1 to n') as $n' \leq 2n$.
7. Now divide these intervals into $(n'-1)$ non-overlapping intervals i.e. (z_i, z_{i+1}) , $T_{i,i'}$ = scheduling interval.
8. There are at most nC_2 scheduling intervals.
9. For each scheduling interval, we can calculate $P_{i,i'} = \frac{\sum_j p_j}{z_{i'} - z_i}$, where $z_i < r_j < d_j < z_{i'}$. [Processing capacity]

$$W_{i,i'} = \sum_{l=i+1}^{i'} s'_l, i < l < i'. \quad [\text{Workload}]$$

10. Minimize the energy using the formula :

$$\text{Min.} \sum_{l=2}^{|Z|} e(s'_l) (z_l - z_{l-1})$$

$$\text{subject to } W_{i,i'} \geq P_{i,i'}, \forall 1 \leq i \leq |Z|$$

$$\sum_{l=2}^{|Z|+1} c(|s'_l - s'_{l-1}|) \leq M$$

11. Repeat a schedule running jobs in order using different speeds $\{s'_l\}$

IV. SIMULATED RESULTS

The program asks for different values like, Processing Speed of processor, Release Times, Deadline Times. The data entered calculates different scheduling times, processing capacity, and the various speeds proportional to the scheduling times. The speed is indicated for each of the time segments. Finally the main subject is displayed i.e. energy consumption before optimization and energy consumption after optimization.

The results can be checked for various processors having different processing speeds and different timing instants.

Different Speed schedules according to the deadlines and the overall energy consumption is calculated with the help of this program.

For example:

Enter the speed value of the processor: 3000Hz

First release time is always zero:

Enter the first dead line which is more than zero: 8ms

Enter the second release time: 14ms

Enter the second dead line: 62ms

Enter the third release time: 81ms

Enter the third dead line: 105ms

$z = 0 \quad 8 \quad 14 \quad 62 \quad 81 \quad 105$

$t = 8 \quad 6 \quad 48 \quad 19 \quad 24$

$p1 = 6 \quad 11 \quad 60 \quad 81 \quad 84$

Processing Capacity will be 11.5965

Speed during First segment is 60.00Hz

Speed during Second segment is 1260.00Hz

Speed during Third segment is 870.00Hz

Speed during Fourth segment is 150.00Hz

Final energy after optimization is 71.30

Final energy before optimization is 78.00

V. CONCLUSION

Motivated by enhancing processor's lifetime reliability from the perspective of designing speed scaling algorithms, investigation is about energy-aware scheduling algorithms in this paper. Contributions include a scheduling algorithm for one model, optimizing energy consumption and cost of frequency changes. Convex Programming Technique is applied for general model. The algorithm that is provided is close to optimal.

VI. FUTURE WORK

In the future research, study of relationship between the frequency and the temperature/heat generated by the processor, in order to get a better understanding processor's lifetime reliability can be envisaged. By doing so a more precise model on processor's lifetime reliability and a good algorithm solution can be implemented.

REFERENCES

- [1] H. Aydin, R. Melhem, D. Mosse, and P. Mejia-Alvarez, "Power-aware scheduling for periodic real-time tasks," *IEEE Transactions on Computers*, vol. 53, no. 10, pp. 584–600, 2004.
- [2] F. Yao, A. Demers, and S. Shenker, "A scheduling model for reduced CPU energy," in *Proceedings of the 36th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 1995, pp. 374–382.
- [3] M. Li, A. C. Yao, and F. F. Yao, "Discrete and continuous min-energy schedules for variable voltage processors," *Proceedings of the National Academy of Sciences of the USA*, vol. 103, no. 11, pp. 3983–3987, 2005.

- [4] H.-L. Chan, W.-T. Chan, T.-W. Lam, L.-K. Lee, K.-S. Mak, and P. W. H. Wong, "Energy efficient online deadline scheduling," in *Proceedings of the 18th Annual ACM-SIAM symposium on Discrete algorithms (SODA)*, 2007, pp. 795–804.

- [5] S. Albers and H. Fujiwara, "Energy-efficient algorithms for flow time minimization," *ACM Transactions on Algorithms (TALG)*, vol. 3, no. 4, p. Article No. 49, 2007.

- [6] K. Pruhs, P. Uthaisombut, and G. Woeginger, "Getting the best response for your erg," *ACM Transactions on Algorithms (TALG)*, vol. 4, no. 3, p. Article No. 38, 2008.

- [7] A. K. Coskun, R. Strong, D. M. Tullsen, and T. S. Rosing, "Evaluating the impact of job scheduling and power management on processor lifetime for chip multiprocessors," *Proceedings of the 11th ACM International Joint Conference on Measurement and Modeling of Computer Systems (SIGMETRICS/Performance)*, 2009, pp. 169–180.

- [8] "Failure mechanisms and models for semiconductor devices, JEDEC publication JEP122C

- [9] H. Yun, P.-L. Wu, A. Arya, T. Abdelzaher, C. Kim, and L. Sha, "System-wide energy optimization for multiple DVS components and real-time tasks," in *Proceedings of the 22nd Euromicro Conference on Real-Time Systems (ECRTS)*, 2010, pp. 133–142.

- [10] M. Li, B. J. Liu, and F. F. Yao, "Min-energy voltage allocation for tree-structured tasks," *Journal of Combinatorial Optimization*, vol. 11, no. 3, pp. 305–319, 2006.

- [11] H. Su, F. Liu, A. Devgan, E. Acar, and S. Nassif, "Full chip leakage-estimation considering power supply and temperature variations," in *Proceedings of the 2003 International Symposium on Low Power Electronics and Design (ISLPED)*, 2003, pp. 78–83.

- [12] Z. Zhang, F. Li, and H. Aydin, "Optimal speed scaling algorithms under speed change constraints," Department of Computer Science, George Mason University.

- [13] M. Dertouzos, "Control robotics: the procedural control of physical processes," in *Proceedings of the IFIP Congress*, 1974, pp. 807–813.