Corollaries to Amdahl’s Law for Energy

Sangyeun Cho, *Member, IEEE* Rami G. Melhem, *Fellow, IEEE*
Dept. of Computer Science, University of Pittsburgh
{cho,melhem}@cs.pitt.edu

Abstract—This paper studies the important interaction between parallelization and energy consumption in a parallelizable application. Given the ratio of serial and parallel portion in an application and the number of processors, we first derive the optimal frequencies allocated to the serial and parallel regions in the application to minimize the total energy consumption, while the execution time is preserved (i.e., speedup = 1). We show that dynamic energy improvement due to parallelization has a function rising faster with the increasing number of processors than the speed improvement function given by the well-known Amdahl’s Law. Furthermore, we determine the conditions under which one can obtain both energy and speed improvement, as well as the amount of improvement. The formulas we obtain capture the fundamental relationship between parallelization, speedup, and energy consumption and can be directly utilized in energy aware processor resource management. Our results form a basis for several interesting research directions in the area of power and energy aware parallel processing.

Index Terms—Parallel processors, power management.

1 Introduction

Amdahl’s Law [1] provides a simple, yet extremely useful method to predict the potential performance of a parallel computer given the ratio of the serial and parallel portion in a program and the number of processors allocated to the program. It has been widely applied in determining speedups even for single CPU machine architectures and has earned the alias “law of diminishing returns” [4]. The following equation succinctly describes the law:

$$\textit{Speedup} = \frac{1}{s + p/N} \tag{1}$$

where $(s+p) = 1$, s (p) is the ratio of the serial (parallel) portion in the program, and N is the number of processors.

Despite the insight and usefulness it provides, Amdahl’s Law does not consider variable processor speed or power consumption. All processor speeds are implicitly assumed to have the same (maximum) value. Because energy and power are one of the most critical shared resources in a multicore-based parallel processor [6], it is not only interesting, but also necessary, to consider the implications of parallelization on program performance and energy consumption together.[1] The current technology and design trends strongly indicate that future processors will be capable of *Dynamic Voltage and Frequency Scaling* (*DVFS* or *DVS* in short) [5], [8], [10]–[12].

In this paper, we begin with the same input parameters as in Amdahl’s Law, namely s (p) and N, and derive the minimum energy consumption one would get with optimal frequency allocation to the serial and parallel region in a program while the execution time is unchanged. We obtain:

$$\textit{Improvement in Dynamic Energy} = \frac{1}{\left(s + \frac{p}{N^{(\alpha-1)/\alpha}}\right)^{\alpha}} \tag{2}$$

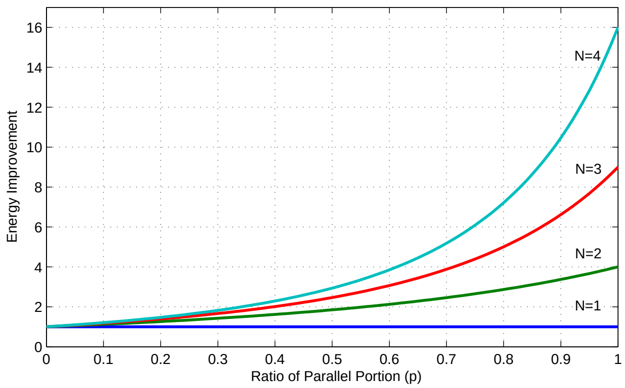

Fig. 1. Achievable dynamic energy improvement assuming $\alpha = 3$ and using 1, 2, 3, and 4 processors given the parallel portion’s ratio of a program.

when the dynamic power consumption of a processor running at f is proportional to f^{α}.[2] The above equation suggests that more parallelism (larger p) and more processors (larger N) help reduce energy consumption. Fig. 1 presents a plot of (2).

We also derive the amount and conditions of the maximum energy improvement given an achievable speedup, or the speedup when the energy is minimized. We show that the total energy is minimized when the dynamic energy is equal to $\frac{1}{\alpha-1}$ the static energy regardless of the value of N or s. Moreover, we find that $\lambda < \frac{\alpha-1}{N}$ is the condition to achieve the minimum energy with the processor speeds smaller than the maximum speed, where λ is the ratio of the static power consumption to the dynamic power consumption at the maximum processor speed. Under this condition, the ratio between the processor speed of the serial and parallel section to minimize energy is $N^{1/\alpha}$. When the condition does not hold, the program’s serial section must be executed at the full speed. If $\lambda > \alpha - 1$, processor speeds in both the serial and parallel sections must be set to the maximum speed to get the minimum total energy.

Manuscript submitted: 23-Oct-2007. Manuscript accepted: 26-Nov-2007. Final manuscript received: 11-Dec-2007.
This work was supported in part by National Science Foundation grant CCF-0702236.

[1] While closely related, power and energy are in general different objectives to optimize. We study only the energy aspect of parallelization in this work.

[2] In DVS literature α is between 2 and 3, typically 3.

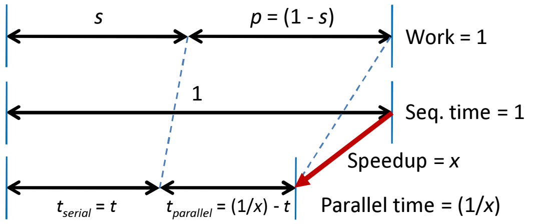

Fig. 2. Normalized "work" and "time." The "parallel time" is partitioned into serial and parallel regions.

2 Energy Improvement with Parallelization

We assume that processors can be run at an arbitrary clock frequency, subject to a maximum frequency, F_{max}. The speedup, x, one would achieve with parallelization and processor speed scaling is subject to:

$$1 \leq x \leq \frac{1}{s+p/N} \tag{3}$$

based on Amdahl's Law in (1).

2.1 Problem formulation

We normalize the sequential execution time of the program to be 1, in order to present our derivation in an intuitive way. Similarly, we normalize the amount of work (*i.e.*, number of cycles) in the program to be 1. Therefore, the maximum clock frequency, F_{max}, has a relative speed of 1 and the program has the serial portion whose amount of work is represented with s, and the parallel portion with p (or $1-s$). Fig. 2 shows this arrangement. We also assume that the dynamic power consumption of a processor running at F_{max} is normalized to 1 and that the static power consumption of a processor is λ. That is, the ratio of the static power to the dynamic power consumption (at maximum speed) of a processor is λ.

The clock frequencies for the two regions in the work, namely s and p, are calculated as follows:

$$f_s = \frac{s}{t} \tag{4}$$

$$f_p = \frac{1-s}{(\frac{1}{x}-t)\cdot N} \tag{5}$$

For a given problem, s is fixed, and for a given architecture, N and λ are fixed. Hence, the energy consumption, E, is a function of t and x. Specifically,

$$E(t,x) = t\cdot f_s^\alpha + N\cdot\left(\frac{1}{x}-t\right)\cdot f_p^\alpha + N\cdot\lambda\cdot\frac{1}{x} \tag{6}$$

In (6) the three terms represent energy for the serial portion, energy for the parallel portion, and energy for the static power consumption during the whole execution time, respectively. The dynamic power consumption of a processor running at f is assumed to be f^α. We do not consider the processor temperature and hence the term for the static energy is the product of per-processor power consumption rate, λ, the number of processors, N, and the total execution time.

2.2 The case of $x = 1$

We first obtain the minimum energy consumption when x is 1 (*i.e.*, program execution time is unchanged).[3] It is trivial that the minimum dynamic energy is obtained at the minimum allowed speedup in (3) since any timing slack due to a speedup can be utilized to reduce energy. To minimize the total energy, we rewrite (6) as

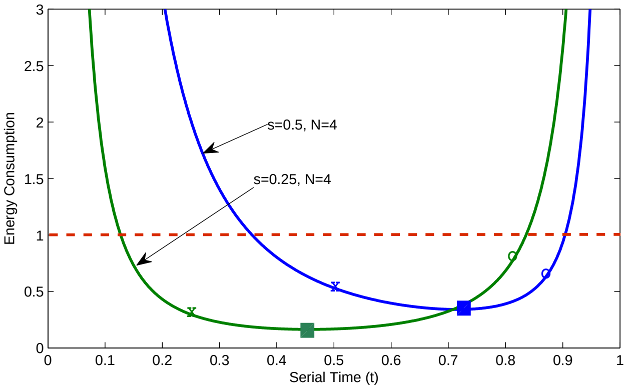

Fig. 3. Dynamic energy consumption vs. t (serial time) for two cases, $s = 0.25$ and $s = 0.5$ when $N = 4$. The bound of t is marked with 'X' (when $f_s = F_{max} = 1$) and 'O' (when $f_p = F_{max} = 1$). The minimum energy point in each curve (at $t = t^*$) is marked with a filled rectangle.

$$E(t) = t\cdot\left(\tfrac{s}{t}\right)^\alpha + N\cdot(1-t)\cdot\left(\tfrac{1-s}{(1-t)\cdot N}\right)^\alpha + N\cdot\lambda \tag{7}$$

Next, we obtain the derivative of $E(t)$ with respect to t,

$$\frac{dE(t)}{dt} = \frac{-(\alpha-1)\cdot s^\alpha}{t^\alpha} + \frac{(\alpha-1)\cdot(1-s)^\alpha}{(1-t)^\alpha\cdot N^{(\alpha-1)}} \tag{8}$$

then, we compute the value of t which minimizes $E(t)$ by setting $dE(t)/dt$ to 0 and obtain:

$$\frac{t}{1-t} = \frac{s}{1-s}\cdot N^{\frac{(\alpha-1)}{\alpha}} \tag{9}$$

Hence, the value of t which minimizes $E(t)$, is:

$$t^* = \frac{s}{s+p/N^{(\alpha-1)/\alpha}} \tag{10}$$

We are ready to obtain the values of f_s and f_p which minimize $E(t)$ using (4), (5) and (10). Specifically,

$$f_s^* = \frac{s}{t^*} = s + \frac{p}{N^{(\alpha-1)/\alpha}} \tag{11}$$

$$f_p^* = \left(s+\frac{p}{N^{(\alpha-1)/\alpha}}\right)\cdot N^{-\frac{1}{\alpha}} \tag{12}$$

$$= f_s^*/N^{\frac{1}{\alpha}} \tag{13}$$

Both f_s^* and f_p^* are a function of s and N in (11), and (12), and (13) shows the relationship between f_s and f_p when $E(t)$ is minimized. Interestingly, the ratio between the two frequencies, f_s^*/f_p^*, is a function of N, but not s.

Finally, from (7) and (10), we obtain the minimum energy consumption:

$$E_{min} = E(t^*)$$

$$= \left(s+\frac{p}{N^{\frac{(\alpha-1)}{\alpha}}}\right)^\alpha + N\cdot\lambda \tag{14}$$

Fig. 1 depicts the maximum energy improvement due to parallelization (E_{min}^{-1}) when the number of processors is varied between 1 and 4, $\alpha = 3$ and $\lambda = 0$. It is clear that energy improvement is a function monotonically increasing with p and N. Fig. 3 shows how the overall energy ($E(t)$) changes

[3] The condition $x = 1$ is similar to setting a deadline (= sequential execution time) to finish the computation by.

as we adjust t. It also presents t^*, the value of t that minimizes $E(t)$. Note that the optimal solution obtained for f_s^* and f_p^* is feasible since both frequencies are smaller than the maximum frequency $F_{max} = 1$.

2.3 The case of $x \geq 1$

Amdahl's Law explores the effect of parallelization on speedup, and in the previous section, we explored the effect of parallelization on energy consumption when the program execution time is unchanged (*i.e.*, $x = 1$). However, because of the effect of the static power, the minimization of the total energy consumption may not occur at $x = 1$. In this section, we revisit the same problem without restricting x. For this, we set the derivatives of (6) with respect to both t and x to zero:

$$\begin{aligned}\frac{\partial E}{\partial t} &= \frac{-(\alpha-1)\cdot s^\alpha}{t^\alpha} + \frac{(\alpha-1)\cdot(1-s)^\alpha}{(\frac{1}{x}-t)^\alpha \cdot N^{(\alpha-1)}} = 0 \implies \\ &\quad \frac{t}{\frac{1}{x}-t} = \frac{s}{1-s}\cdot N^{(\alpha-1)/\alpha} \qquad (15)\\ \frac{\partial E}{\partial x} &= \left(\frac{(\alpha-1)\cdot(1-s)^\alpha}{(\frac{1}{x}-t)^\alpha \cdot N^{\alpha-1}} - \lambda N\right)\cdot\frac{1}{x^2} = 0 \implies \\ &\quad (\frac{1}{x}-t) = \left(\frac{\alpha-1}{\lambda}\right)^{\frac{1}{\alpha}}\cdot\frac{1-s}{N} \qquad (16)\end{aligned}$$

From (15) and (16),

$$t^* = \left(\frac{\alpha-1}{\lambda N}\right)^{\frac{1}{\alpha}}\cdot s \qquad (17)$$

$$x^* = \left(\frac{\lambda N}{\alpha-1}\right)^{\frac{1}{\alpha}}\cdot\left(\frac{1}{s+\frac{p}{N^{(\alpha-1)/\alpha}}}\right) \qquad (18)$$

With t^* and x^*, we can use (4) and (5) to calculate the optimum frequencies

$$f_s^* = \left(\frac{\lambda N}{\alpha-1}\right)^{\frac{1}{\alpha}} \qquad (19)$$

$$f_p^* = \left(\frac{\lambda}{\alpha-1}\right)^{\frac{1}{\alpha}} = f_s^*/N^{\frac{1}{\alpha}} \qquad (20)$$

from which we can compute the minimum energy. An interesting observation is that at f_s^* and f_p^*, the dynamic energy is given by

$$\begin{aligned}E_{dynamic} &= t^*\cdot f_s^\alpha + N\cdot\left(\frac{1}{x^*}-t\right)\cdot f_p^\alpha \qquad (21)\\ &= \frac{1}{\alpha-1}\cdot\frac{N\lambda}{x^*} \qquad (22)\end{aligned}$$

which is equal to $\frac{1}{\alpha-1}$ of the static energy, $E_{static} = \frac{N\lambda}{x^*}$. In other words, the total energy consumption is minimized when the dynamic energy consumption is $\frac{1}{\alpha-1}$ times the static energy consumption. This relation holds during the execution of both the serial and the parallel sections of the program.

The above solution is only applicable if both f_s^* and f_p^* are smaller than F_{max}, however, necessitating that $\lambda N \leq \alpha - 1$. If λ, the ratio between the static and dynamic power, is large so that it is not possible to maintain the aforementioned relation between the static and dynamic energy, we should set $f_s = 1$ and find the values of x and f_p that minimize the total energy consumption. Denoting these values by x^{**} and f_p^{**}, we obtain

$$x^{**} = \frac{1}{s+(\frac{p}{N})\cdot(\frac{\alpha-1}{\lambda})^{\frac{1}{\alpha}}} \qquad (23)$$

$$f_p^{**} = (\frac{\lambda}{\alpha-1})^{\frac{1}{\alpha}} \qquad (24)$$

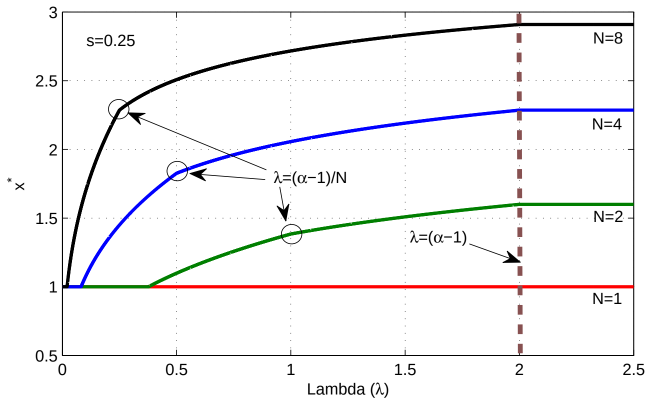

Fig. 4. λ changes x^*, the speedup of a program when its energy consumption is minimized. x^* saturates at the maximum speedup Amdahl's Law dictates when $\lambda = \alpha - 1$. We assume that $\alpha = 3$.

Again, these values result in the dynamic power consumption being $\frac{1}{\alpha-1}$ times the static power consumption during the execution of the parallel portion of the program.

Finally, if the static power consumption is so high such that $\lambda > \alpha - 1$, then the minimum energy is obtained when $f_s = f_p = 1$. That is, when the processors execute at the maximum speed to finish as fast as possible.

In order to summarize the relationship between λ and the speedup that results in the minimum energy consumption, we show that relationship in Fig. 4. In this figure, the values of λ are divided into three regions. When $\lambda \leq \frac{\alpha-1}{N}$, the solution for the optimum energy consumption problem is given by equations (18), (19) and (20). When $\frac{\alpha-1}{N} < \lambda \leq \alpha - 1$, the solution is given by $f_s = 1$, equations (23) and (24). Finally, when $\lambda > \alpha - 1$, the solution is given by $f_s = f_p = 1$, and the speedup is that given by Amdahl's Law (equation (1)).

2.4 Optimal energy consumption given a speedup

We have thus far considered the problem of calculating the optimal speeds of processors (hence program speedup) to minimize the total energy consumption given p, λ, and N. In this section, we consider the problem of how to set the speeds of the processors (f_s and f_p) to minimize the total energy consumption when the target program speedup x is specified. Because the static energy $N \cdot \lambda \cdot \frac{1}{x}$ is immediately determined given x, we only need to minimize the dynamic energy while meeting the program speedup requirement and our solution derived from (4), (5), (18), (19), and (20) is as follows.

$$\text{If} \quad 1 \leq x \leq \frac{1}{s+\frac{p}{N^{(\alpha-1)/\alpha}}}, \quad f_s^* = x f_{s,x=1}^*, f_p^* = x f_{p,x=1}^* \qquad (25)$$

$$\text{If} \quad \frac{1}{s+\frac{p}{N^{(\alpha-1)/\alpha}}} < x \leq \frac{1}{s+\frac{p}{N}}, \quad f_s^* = 1, f_p^* = \frac{px}{N(1-sx)} \qquad (26)$$

where $f_{s,x=1}^*$ and $f_{p,x=1}^*$ are the optimal frequencies when $x = 1$ given in equations (11) and (12). We call the interval in (25) the *linear frequency scaling interval* because the energy-optimal f_s and f_p can be obtained by simply scaling $f_{s,x=1}^*$ and $f_{p,x=1}^*$ by a factor of x. We also note that the upper bound of the condition in (25) is equivalent to $\lambda \leq \frac{\alpha-1}{N}$.

Fig. 5 shows how the minimum energy consumption changes as we target a different program speedup, along with the contribution of the dynamic and static energy consumption. It is noticeable from the plot that the dynamic energy of the

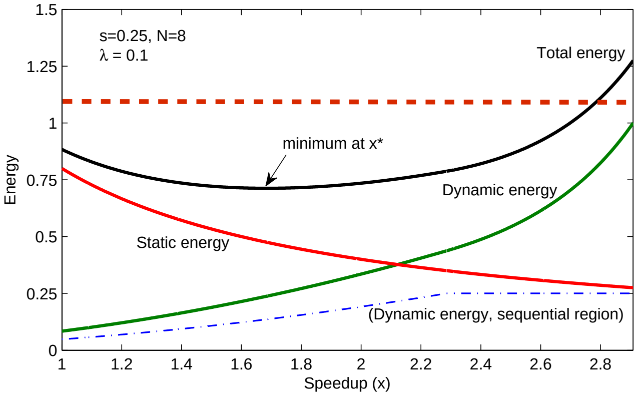

Fig. 5. Optimal energy given the speedup of x, $\alpha = 3$. Total energy is the sum of dynamic and static energy. Dynamic energy for the sequential region is also shown. The thick dotted line shows the sequential machine's energy consumption. Given the parameters, the maximum speedup (Amdahl's Law) is 2.909 and $x^* = 1.684$ from (18). The dynamic energy of the sequential region saturates at 2.286 from (25).

sequential region saturates at around $x = 2.3$. This is due to the inability to scale f_s beyond F_{max}. Finally, when $f_s = f_p = 1$ (*i.e.*, at the maximum speedup), the dynamic energy is 1; it is same as that of the sequential execution.

3 Related Work

Energy-saving techniques that utilize available timing slack with DVS have been extensively studied, especially in the domain of real time task scheduling [5], [8], [10]–[12]. While much previous work has been on improving energy on uniprocessor systems, Zhu *et al.* [12] introduces the concept of *slack sharing* on multiprocessor systems and Mishra *et al.* [8] utilizes the static slack based on the degree of parallelism in a schedule. Horvath *et al.* [5] studied energy optimization strategies for large-scale multi-tier web servers using DVS. Compared with previous heuristic-oriented energy-aware task scheduling strategies, our work in this paper focused on understanding the interaction between parallelization, performance and energy consumption by constructing an analytical model to directly derive optimal energy of a parallelized application.

More recently, Ge *et al.* [3] studied distributed performance-directed DVS strategies for use in HPC clusters, which exploit fine-grained timing slacks to save energy. In their subsequent work, Ge and Cameron [2] studied *power-aware speedup*, defined as the ratio between the single-processor performance at the lowest available on-chip frequency and the parallel execution time. They partition a given workload into a portion of the workload accessing on-chip data and another portion accessing off-chip data. The main goal of their study is to derive a more accurate parallel speedup model for modern processors capable of DVS.

Finally, Li and Martínez [7] presented an analytical model to derive power consumption and performance when nominal parallel efficiency and the number of processors in a chip multiprocessor are given. Unlike our work, they do not consider serial and parallel region in a parallel application separately and assign a single speed to all the available processors.

4 Conclusions and Future Work

We have considered the problem of minimizing the total energy consumption for a given architecture (values of N, α and λ) and a given problem with a known ratio of parallelizable section (value of p). We have analytically derived the formula for the processor speeds that minimize the energy consumption and shown that at those speeds, the dynamic energy consumption is equal to $\frac{1}{\alpha-1}$ the static energy consumption. Hence, to minimize energy, this relation between static and dynamic energy should be maintained as long as the processor speeds do not exceed the maximum allowable speed, in which case, the maximum speed should be used.

In many systems, it is desirable to strike a trade-off between energy consumption and performance by minimizing the energy-delay product rather than the total energy. The same type of analysis can be performed and the results show that the optimal energy-delay is obtained when $f_s = N^{\frac{1}{\alpha}} f_p$ and $f_p = (\frac{2\lambda}{\alpha-2})^{\frac{1}{\alpha}}$, $\alpha > 2$. Details of the analysis is not included in this paper due to space limitation.

The formula derived in this paper also show that, for a given processor implementation (λ and α), the minimum total energy is a monotonically decreasing function of the number of processors, N, as long as the parallel section of the code can be executed on N processors. Hence, from the total energy consumption point of view, all available processors should be used. It should be noted, however, that this result, and all the results given in this paper, assume that the N processors in the system consume static power, even during the serial section of the code. If individual processors can be turned off and back on with low overhead, then the formula for the total energy in equation (6) should be changed such that $N \cdot \lambda \cdot \frac{1}{x}$ is replaced by $(t + (\frac{1}{x} - t) \cdot N) \cdot \lambda$. We leave the solution of the energy minimization problem in this case to future work.

References

[1] G. M. Amdahl. "Validity of the single processor approach to achieving large scale computing capabilities," *AFIPS Conf. Proc.*, pp. 483–485, 1967.

[2] R. Ge and K. W. Cameron. "Power-Aware Speedup," *Proc. Int'l Parallel and Distributed Processing Symp.*, pp. 1–10, March 2007.

[3] R. Ge, X. Feng, and K. W. Cameron. "Performance-constrained Distributed DVS Scheduling for Scientific Applications on Power-aware Clusters," *Proc. Conf. Supercomputing*, pp. 34–44, Nov. 2005.

[4] J. L. Hennessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach*, 4th Ed., Morgan Kaufmann, 2007.

[5] T. Horvath, T. Abdelzaher, K. Skadron, and X. Liu. "Dynamic Voltage Scaling in Multitier Web Servers with End-to-End Delay Control," *IEEE Trans. Computers*, 56(4):444–458, Apr. 2007.

[6] ITRS. 2005 Edition. `http://public.itrs.net`.

[7] J. Li and J. F. Martínez. "Power-Performance Considerations of Parallel Computing on Chip Multiprocessors," *ACM Trans. Architecture and Code Optimization*, 2(4):397–422, December 2005.

[8] R. Mishra, N. Rastogi, D. Zhu, D. Mossé, and R. Melhem. "Energy Aware Scheduling for Distributed Real-Time Systems," *Proc. Int'l Parallel and Distributed Processing Symp.*, pp. 21–29, April 2003.

[9] R. Ronen, A. Mendelson, K. Lai, S.-L. Lu, F. Pollack, and J. P. Shen. "Coming Challenges in Microarchitecture and Architecture," *Proc. IEEE*, 89(3):325–340, March 2001.

[10] D. Shin, J. Kim, and S. Lee. "Intra-Task Voltage Scheduling for Low-Energy Hard Real-Time Applications," *IEEE Design and Test of Computers*, 18(2):20–30, Mar.-April 2001.

[11] F. Yao, A. Demers, and S. Shenker. "A Scheduling Model for Reduced CPU Energy," *Proc. Symp. Foundations of Computer Science*, pp. 374–382, Oct. 1995.

[12] D. Zhu, R. Melhem, and B. R. Childers. "Scheduling with Dynamic Voltage/Speed Adjustment Using Slack Reclamation in Multi-Processor Real-Time Systems," *Proc. Real-Time Systems Symp.*, pp. 84–94, Dec. 2001.