Energy-efficient scheduling for multiprocessors

W.Y. Lee and H. Lee

An energy-efficient scheduling algorithm is proposed for parallel tasks in a multiprocessor system. The proposed algorithm utilises the dynamic voltage scaling (DVS) method for low energy consumption and executes tasks in parallel to compensate for the execution delay induced by the DVS method.

Introduction: Dynamic power management (DPM) [1] and dynamic voltage scaling (DVS) [2] methods are proposed to manage the energy consumption of electronic systems. In DPM, unused components are turned off for the purpose of reducing energy consumption. In DVS, the voltage supplied to a processor is dynamically decreased. Its energy consumption ratio and speed then decrease on the basis of the following relationships [3, 4]: Energy $\propto V \times (V - V_T)^2$ and Speed $\propto (V - V_T)^2/V$, where V denotes the voltage supplied to the processor and V_T denotes the threshold voltage of the processor chip. For the design of low-energy consuming systems, previous work [3, 4] only focused on the case where a task is executed on a single processor, even though there are available processors and the task can be executed in parallel on multiple processors. Some tasks, such as image calculation, geographical information processing, and particle simulation, can easily be divided into multiple computational components. If a task can be executed on multiple processors, its parallel execution time decreases as the number of allocated processors increases [5]. Additionally, if a lower voltage is applied to the processors executing the task, the parallel execution time of the task increases but the total energy consumption of the multiple processors decreases, as presented in Fig. 1. In this Letter, an energy-efficient scheduling algorithm is proposed. This algorithm executes tasks in parallel to achieve fast execution, and utilises the DVS method for low-energy consumption while maintaining equal execution time.

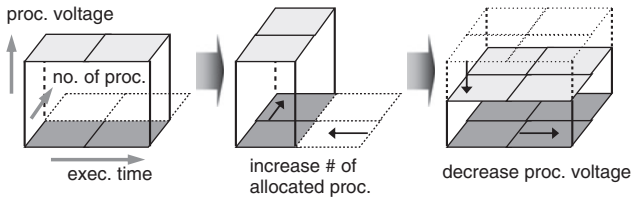

Fig. 1 *Execution of task on multiple processors with low voltage*

Problem definition: Based on previous studies [4, 5], the execution time to complete the workload (number of processor cycles) W on P processors with V voltage is formulated as follows:

$$\boldsymbol{E}(W, P, V) = h_1/V \times (W + p_o)/P$$

where h_1 is a hardware-dependent constant and p_o is the overhead for parallel execution, such as initial distribution process of subtasks, communication among processors and unbalanced load distribution ($p_o = 0$ if $P = 1$) [5]. In addition, based on the previous studies [3, 4], the energy consumption rate at time τ on P processors with V voltage is formulated as follows:

$$\boldsymbol{C}^\tau(P, V) = h_2 \times V^3 \times P$$

where h_2 is a hardware-dependent constant. In this Letter, we investigate an algorithm which finds a schedule to execute M tasks on N identical processors with minimum energy consumption. Each task T_m has its workload W_m, starts execution at s_m and finishes execution at $f_m = s_m + \boldsymbol{E}(W_m, 1, V_o) = s_m + h_1/V_o \times W_m$, where V_o is the initial voltage supplied to the processors. V_m^τ and N_m^τ denote the voltage supplied to T_m and the number of processors allocated to T_m at time τ, respectively. The total amount of energy consumption to execute T_m is

$$\int \boldsymbol{C}^\tau(N_m^\tau, V_m^\tau)d\tau + e_o \times \delta$$

where e_o is the overhead required to dynamically change the voltage supplied to processors and δ is the number of voltage changes during the execution of T_m.

Scheduling algorithm: M tasks are sorted in increasing order of their start times and are stored in a list $TL = [T_1, \ldots, T_m, \ldots, T_M]$. All s_m and f_m values of the M tasks are sorted in increasing order and their values are labelled with another index r^k when their corresponding ranking is kth. A range R^k can then be defined as $R^1 = [r^1, r^2]$, $R^2 = [r^2, r^3]$, $\ldots$, $R^{2M} = [r^{2M}, \infty]$. The proposed algorithm, called Energy-efficient Scheduling for Multiprocessor (ESM), is described below:

Algorithm Energy-efficient Scheduling for Multiprocessor (ESM)
Input: TL, N and V_o
Output: N_m^τ and V_m^τ such that $s_m \le \tau \le f_m$ for each T_m
For $k=1$ **to** $k=(2M-1)$ **do**
{ $W^k \leftarrow V_o/h_1 \times (r^{k+1} - r^k)$;
$M^k \leftarrow$ number of tasks T_m such that $s_m \le r^k < r^{k+1} \le f_m$;
$P_q \leftarrow N$ **div** M^k; $P_r \leftarrow N$ **mod** M^k;
For $x=1$ **to** $x=(M^k - P_r)$ **do**
{ $N_x^\tau \leftarrow P_q$ **and** $V_x^\tau \leftarrow V_o/W^k \times (W^k + p_o)/P_q$ **from** $\tau = r^k$ **to** $\tau = r^{k+1}$;}
For $x=(M^k - P_r + 1)$ **to** $x = M^k$ **do**
{ $N_x^\tau \leftarrow (P_q + 1)$ **and** $V_x^\tau \leftarrow V_o/W^k \times (W^k + p_o)/(P_q + 1)$ **from** $\tau = r^k$ **to** $\tau = r^{k+1}$;}
}

The ESM algorithm divides the workload $W_m = V_o/h_1 \times (f_m - s_m)$ of each T_m into several parts $W^k = V_o/h_1 \times (r^{k+1} - r^k)$ along with ranges R^k, such that $s_m \le r^k < r^{k+1} \le f_m$. It allocates N processors evenly to the M^k tasks which can be executed in R^k. When $\boldsymbol{\Phi}(P) = V_o/W^k \times (W^k + p_o)/P$, each workload W^k of the M^k tasks is executed on P_q processors with $\boldsymbol{\Phi}(P_q)$ voltage (or on $(P_q + 1)$ processors with $\boldsymbol{\Phi}(P_q + 1)$ voltage), because $\boldsymbol{E}(W^k, 1, V_o) = \boldsymbol{E}(W^k, P_q, \boldsymbol{\Phi}(P_q)) = \boldsymbol{E}(W^k, P_q + 1, \boldsymbol{\Phi}(P_q + 1))$. The workload W^k of the M^k tasks is completely executed at time r^{k+1} for any R^k. The ESM algorithm always utilises all processors, while the DPM method turns off the power of idle processors. The total energy consumption of ESM in each R^k is

$$(M^k - P_r) \times \int_a^b \boldsymbol{C}^\tau(P_q, \Phi(P_q))\mathrm{d}\tau + P_r \times \int_a^b \boldsymbol{C}^\tau(P_q + 1, \Phi(P_q + 1))\mathrm{d}\tau + e_o$$

where $a = r^k$ and $b = r^{k+1}$. The total energy consumption of DPM is

$$M^k \times \int_a^b \boldsymbol{C}^\tau(1, V_o)d\tau + e_o$$

If $p_o \le 0.58 \times W^k < (4^{1/3} - 1) \times W^k$, then

$$\int_a^b \boldsymbol{C}^\tau(1, V_o)\mathrm{d}\tau > \int_a^b \boldsymbol{C}^\tau(P_q, \Phi(P_q))d\tau > \int_a^b \boldsymbol{C}^\tau(P_q + 1, \Phi(P_q + 1))d\tau$$

for any $P_q > 1$. Hence, the energy consumption of ESM is always better than that of DPM when the parallel execution overhead of a task is less than 58% of its workload. The ESM algorithm allocates an available processor to the task having fewest allocated processors. If $p_o \le 0.37 \times W^k < ((18/7)^{1/3} - 1) \times W^k$, then

$$\int_a^b \boldsymbol{C}^\tau(1, V_o)d\tau - \int_a^b \boldsymbol{C}^\tau(2, \Phi(2))d\tau > \int_a^b \boldsymbol{C}^\tau(2, \Phi(2))d\tau - \int_a^b \boldsymbol{C}^\tau(3, \Phi(3))d\tau \Leftrightarrow (W^k)^3 - (W^k + p_o)^3/4 > (W^k + p_o)^3/4 - (W^k + p_o)^3/9$$

If $1 < N_x < N_y$, then

$$\int_a^b \boldsymbol{C}^{\tau}(N_x, \Phi(N_x))d\tau - \int_a^b \boldsymbol{C}^{\tau}(N_x + 1, \Phi(N_x + 1))\mathrm{d}\tau > \int_a^b \boldsymbol{C}^{\tau}(N_y, \Phi(N_y))d\tau - \int_a^b \boldsymbol{C}^{\tau}(N_y + 1, \Phi(N_y + 1))d\tau$$

Hence, this allocation method maximises the decrement of energy consumption when the parallel execution overhead of a task is less than 37% of its workload.

Evaluation: ESM is compared with DPM through simulation. The ratio of the total energy consumption amount of all tasks in ESM to that in DPM is referred to as Relative Energy Consumption Ratio and used for our evaluation metric. The load imposed on the system by task arrivals is referred to as System Load, and formulated as $\lambda \times \theta/N$ where λ is the task arrival rate of Poisson process and θ is the mean computation time of tasks. The workload of tasks (the value of W_m) is normally distributed with a mean of 100. The overhead e_o is set as $0.05 \times \theta$ and the overhead p_o is set as a constant ratio of W_m. The average values of 20 runs for independent tasks are shown and each run is performed over 1 000 000 time units. Fig. 2*a* shows the performance of the dual-processor system ($N = 2$). 'ESM:α' denotes the performance of ESM when $p_o = \alpha \times W_m/100$. 'ESM:00', 'ESM:20', 'ESM:40' and 'ESM:60' show 43, 57, 76 and 102% energy consumption of DPM, respectively, when System Load is 0.1. Their differences become smaller when the system is more heavily loaded. Fig. 2*b* shows the performance when System Load is fixed at 0.5 and N increases exponentially with the function of 2^X. 'ESM:00', 'ESM:20', 'ESM:40' and 'ESM:60' show 76, 82, 90 and 101% energy consumption of DPM, respectively, when $N = 2$. Their differences increase as the value of N increases. When $N = 1024$, 'ESM:00', 'ESM:20', 'ESM:40' and 'ESM:60' show 35, 49, 70 and 90% energy consumption of DPM, respectively.

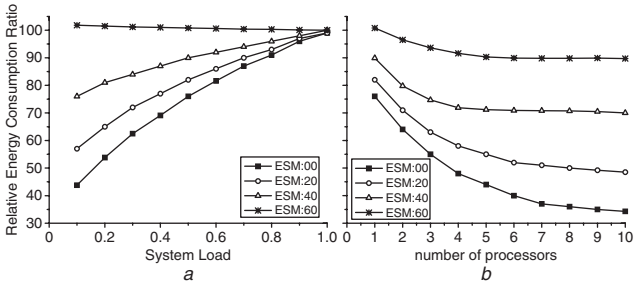

Fig. 2 *Relative energy consumption ratio with values of System Load and N*

Conclusions: We propose an energy-efficient scheduling algorithm which utilises all available processors to execute tasks in parallel and uses the lowest voltage possible, while maintaining equal execution time. The proposed algorithm achieves much lower energy consumption when the parallel execution overhead is smaller, the system is more lightly loaded, or the system has a larger number of processors.

Acknowledgment: This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD) (KRF-2005-214-D00336)

29 May 2006
Electronics Letters online no: 20061672
doi: 10.1049/el:20061672

W.Y. Lee (*Department of Computer Engineering, Hallym University, Chunchon 200-702, South Korea*)

E-mail: wanlee@hallym.ac.kr

H. Lee (*Department of Computer Science and Engineering, Korea University, Seoul 136-701, South Korea*)

References

1 Benini, L., Dogliolo, A., and Micheli, G.D.: 'A survey of design techniques for system-level dynamic power management', *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, 2000, **8**, (3), pp. 299–316
2 Pillai, P., and Shin, K.G.: 'Real-time dynamic voltage scheduling for low-power embedded operating systems'. Proc. ACM Symp. Operating System Principles, 2001, Vol. 18, pp. 89–102
3 Anderson, J.H., and Baruah, S.K.: 'Energy-efficient synthesis of periodic tasks systems upon identical multiprocessor platform'. Proc. Int. Conf. Distributed Computing Systems, 2004, Vol. 24, pp. 428–435
4 Yang, C., Chen, J., and Kuo, T.: 'An approximation algorithm for energy-efficient scheduling on a chip multiprocessor'. Design Automation and Test in Europe, 2005, Vol. 8, pp. 468–473
5 Lee, W.Y., Hong, S.J., and Kim, J.: 'On-line scheduling of scalable real-time tasks on multiprocessor systems', *J. Parallel Distrib. Syst.*, 2003, **63**, (12), pp. 1315–1324