Adaptive Online Voltage Scaling Scheme Based on the Nash Bargaining Solution

Sungwook Kim

In an effort to reduce energy consumption, research into adaptive power management in real-time systems has become widespread. In this paper, a novel dynamic voltage scaling scheme is proposed for multiprocessor systems. Based on the concept of the Nash bargaining solution, a processor's clock speed and supply voltage are dynamically adjusted to satisfy these conflicting performance metrics. In addition, the proposed algorithm is implemented to react adaptively to the current system conditions by using an adaptive online approach. Simulation results clearly indicate that the superior performance of the proposed scheme can strike the appropriate performance balance between contradictory requirements.

Keywords: Dynamic voltage scaling, multiprocessor power management, Nash bargaining solution, energy efficiency, online decisions.

Manuscript received July 14; revised Sept. 10, 2010; accepted Oct. 4, 2010.

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (2010-0021163) and by the Sogang University Research Grant of 2011 (201110011.01).

Sungwook Kim (phone: +82 2 705 8932, email: swkim01@sogang.ac.kr) is with the Department of Computer Science, Sogang University, Seoul, Rep. of Korea.

doi:10.4218/etrij.11.0110.0417

I. Introduction

With the advanced technology of very large-scale integration (VLSI) circuit designs, modern embedded systems have evolved from a uniprocessor to a multiprocessor approach to enhance performance. Therefore, the use of multiprocessor systems is dramatically increasing and becoming the de-facto standard [1]-[3]. In addition, applications in a multiprocessor system are consuming more energy due to the continuously increasing functionality. To support these energy-consuming task services, energy-aware multiprocessor system design has become a new frontier research field [1]-[4].

The dynamic voltage scaling (DVS) technique has been investigated extensively and several commercial DVS microprocessors have been developed [1], [4]. A significant feature of DVS processors is in changing the operational voltage and frequency dynamically to adapt to current system conditions. Since power consumption has a quadratic dependency on the supply voltage, lowering the supply voltage is the most effective way to enhance system efficiency. However, many different applications, namely avionics, traffic control, automated factories, and military systems have specific requirements based on different characteristics, that is, execution cycles and a relative deadline; task services need to guarantee its deadline to ensure required quality of service (QoS). Therefore, DVS systems should consider the timing issue to meet deadline requirements. However, between energy efficiency and QoS provisioning, there is usually a tradeoff involved. To carefully balance among conflicting performance criteria, voltage scaling becomes a more challenging and complex control problem in real-world system operations [1], [4], [5].

The fundamental problem faced by control algorithms is to design a decision mechanism; an effective decision mechanism is a key factor for system performance. Game theory is a field of applied mathematics that provides an effective tool in modeling the interactions among independent decision makers [6]-[8]. It can describe the possibility of reacting to the actions of the other decision makers and analyze the situations of conflict and cooperation. The rational decision makers, referred to as 'players' in a game model, try to maximize their expected benefits through a strategy set. Many applications of the game theory are related to economics, but it is also a powerful tool to model a wider range of real life situations, such as political science, sociology, psychology, and biology, where conflict and cooperation exist. Since the early 1990s, computer science and system management have been added to this list [8].

In 1950, John Nash introduced a new model of cooperative game theory, that is, the Nash bargaining solution (NBS), to allocate resources fairly and optimally [6], [7]. The NBS can achieve a mutually desirable solution with a good balance between efficiency and fairness. In addition, the NBS does not require global objective functions, unlike conventional optimization methods, such as Lagrangian or dynamic programming [6]. Based on these appealing properties, the basic concept of the NBS has found widespread use in many engineering fields. In this paper, we adopt the NBS model to design a real-time DVS control algorithm. However, for various reasons, the classical NBS method cannot be directly applied to the processor power management. First, the traditional NBS was derived in the context of economics, so it is not appropriate for dynamic multiprocessor systems. Second, a static one-shot game model is an impractical approach to justify realistic system operations. Third, it is technically unable to take into account complex interactions for multiple objectives. In addition, due to the model complexity, it is not amenable to mathematical modeling and numerical analysis.

For efficient power management, control decisions must be dynamically adjustable. However, future task requests are generally not known; these decisions have to be made in real time and without the knowledge of future information at the decision time. Therefore, online algorithms are natural candidates for the design of efficient control problems in real-time system operations. Offline algorithms are unrealizable because they need full knowledge of the future for an online problem [2], [9]. Besides, under dynamically changing system environments, traditional static control strategies are not acceptable. Based on these considerations, we would employ a dynamic online methodology for power management, which can improve adaptability under widely different and diversified multiprocessor system situations.

Motivated by the above discussion, we propose a new online multiprocessor power control scheme based on the NBS model. The main design goal of the proposed scheme is to simultaneously maximize energy efficiency while ensuring all task deadlines. The following are the important features of the proposed scheme: i) the ability to maintain energy efficiency as high as possible, ii) the ability to respond to current system situations based on the real-time information, iii) the ability to achieve load balancing among processors, (iv) the adaptive online process to make control decisions, (v) the ability to meet all task deadlines, and (vi) a well-balanced system performance among conflicting requirements. The principal novelties of the proposed scheme are its adaptability, feasibility, and effectiveness for realistic system operations.

Recently, several DVS control schemes have been presented for processor power management. The energy efficient power control (EEPC) scheme in [1] provides a suitable solution model for the power-aware control problem. For dynamic environments, the EEPC scheme also investigates resource allocation techniques to control real-time multiple tasks. The transition-overhead aware voltage scheduling (TAVS) scheme [4] is designed to reduce the energy consumption of systems by considering transition time and energy overhead. Therefore, the TAVS scheme can effectively estimate the actual execution cycles and lead to better transition overhead management. All the earlier work has attracted much attention and introduced unique challenges. However, there are several shortcomings as described in section III. Compared to these existing schemes [1], [4], the proposed scheme provides excellent system performance under different system load distributions.

This paper is organized as follows. Section II explains the proposed algorithms in detail. In section III, performance evaluation results are presented along with comparisons with the existing schemes [1], [4]. Finally, concluding remarks are given in section IV.

II. Proposed Power Management Algorithm

In this section, the proposed algorithm is explained in detail. Based on current system conditions, individual tasks are scheduled to globally optimize system performance. To get a satisfactory solution, processors adjust their power levels according to the NBS model and schedule tasks in an online interactive manner.

1. Basic System Model

Multiprocessor systems use two or more processors to execute tasks. Therefore, efficient task assignment to multiprocessors is one of the key issues for the effective system utilization. In this paper, a new task scheduling algorithm is developed to adaptively spread workload among processors while ensuring energy efficiency for the total task processing.

As mentioned earlier, one promising power and energy reduction technique is voltage control. However, there is a power-delay tradeoff by controlling the supply voltage. For instance, under the Dhrystone 1.1 benchmark programs, an ARM7D processor can run at 33 MHz and 5 V as well as at 20 MHz and 3.3 V. The energy-performance measures at these two operation modes are 185 MIPS/watt and 579 MIPS/watt, and the MIPS measures are 30.6 and 19.1, respectively. Thus, if a system switches from 33 MHz and 5 V to 20 MHz and 3.3 V, there will be around (579–185)/579=68% reduction in energy consumption at an expense of (30.6–19.1)/19.1=60% increase of processing time [10].

In this paper, we define Ψ as the set of accepted service requests sr by the real-time system, $\Psi = \{sr_1, sr_2, sr_3, \ldots, sr_i, \ldots, sr_n\}$, where n is the total number of accepted services, that is, $n = \|\Psi\|$. During real-world system operations, n is dynamically changed. The service request i is characterized by $\{a_i, d_i, t\_c_i\}$ where a_i is the arrival time, d_i is the deadline, and t_c_i is the total workload of service sr_i to be completed.

In the DVS system, processors have different power states according to the set of voltage levels; each power state of a processor is characterized by a different speed (performance). Therefore, between its arrival time and the deadline, task services are amenable to adaptation with a variable processor speed, which is defined as a discrete voltage level with multiple grades of clock frequency. At the current time c_t, processor j's speed $S_{p\_j}(c_t)$ is defined as the sum of the assigned processor frequency level for each running service, which is given by

$$S_{p\_j}(c_t) = \sum_{i=1}^{n} \{PS_i(c_t) \times A(i,j)\}, \tag{1}$$

where $A(i,j) = \begin{cases} 1 \text{ if } sr_i \text{ is accepted to the processor } j, \\ 0 \text{ otherwise,} \end{cases}$

where n is the total number of requested tasks, and $PS_i(c_t)$ is the frequency level (speed) for the task i service at the current time.

When the offered system load is heavy, that is, the sum of the requested tasks exceeds the available processor capacity, an admission control algorithm needs to be employed. Based on the acceptance condition, the admission procedure makes a decision whether to accept a requested task or not.

Acceptance condition:

$$\max_{1<j<k} \left[\int_{c_t}^{d_t} MP_j(t)dt - \int_{c_t}^{d_t} S_{p\_j}(t)dt \right] \geq t\_c_i, \tag{2}$$

where MP_j is the maximum speed (total processor computation capacity) of the processor j, and k is the total number of processors in the system. When a new task arrives, this task is accepted if the system meets the acceptance condition. Otherwise, the new requested task cannot be completely served within the deadline; the requested task is rejected. The control parameters used in the proposed algorithm are given next table.

Table 1. Control parameters in proposed algorithm.

Parameter	Description
Ψ	Set of task requests by the system
sr_i	Service request i to be executed by the system
n	Total number of accepted services
k	Total number of processors in the system
a_i	Time at which the sr_i arrives to be executed
d_i	Time by which the task tr_i must be completed
t_c_i	Total execution cycles of the i-th task.
c_t	Current time during the system operation
$S_{p\_j}(c_t)$	Processor j's speed
$PS_i(c_t)$	Frequency level for the task i service
MP_j	Maximum speed of the processor j

2. Bargaining Model for DVS Control Algorithm

Based on physical law [11], energy is reduced in direct proportion to the processor power state; it is more energy efficient to slow down the processor power as much as possible. However, due to the required QoS, the time deadline should not be overlooked altogether. In this paper, to approximate the optimal voltage, the proposed multiprocessor power control algorithm is developed as a cooperative game model. Usually, the cooperative game approach is attractive for resource allocation and load balancing problems [6], [7]. The first cooperative game model was conceived by John Nash. Based on traditional game theory, Nash proposed the NBS, which can be formulated as follows: there are n players; player i has its own utility function (u_i). Assume $\mathbb{S} = \{(u_1, \ldots, u_n)\} \subset \mathbb{R}^n$ is a joint-utility solution set that is a nonempty, convex, closed, and bounded feasible utility set. In set $\mathbb{S}$, some solutions are characterized such that one player cannot increase his utility without decreasing the utility of any other players. This solution set is called as the Pareto optimal points/surface, which are payoff pairs in the cooperative tradeoff area [6].

One agreement point $\boldsymbol{u}$ ($\boldsymbol{u} \in \mathbb{S}$), which is an action vector in the Pareto surface, is a possible outcome of the bargaining process. A disagreement point $\boldsymbol{d}$ is an action vector $\boldsymbol{d} = (d_1, \ldots, d_n) \in \mathbb{S}$ that is expected to be the result if players cannot reach agreement. This, at least, is guaranteed for each user in the cooperative game. Therefore, the payoff at any agreement point is always higher or equal to the payoff achieved at the disagreement point. The pair ($\mathbb{S}$, $\boldsymbol{d}$) defines the bargaining problem. The bargaining solution can formally be defined as

$$\prod_i (u_i^* - d_i) = \max_{u_i \in \mathbb{S}} \prod_i (u_i - d_i), \tag{3}$$

where $u_i^* \in \mathbb{S}$ and $d_i \in \boldsymbol{d}$.

In game theory terminology, an outcome vector $< u_1^*, u_1^*, \ldots, u_n^* >$ is called the NBS. Therefore, in multiple Pareto optimal solutions, the NBS decides the best one, that is, the one which can give a unique and fair-efficient solution [6].

The main advantage of using the NBS is that the overall management strategy allows use of lower level information, freeing itself from the need to execute complex algorithms [1], [6]. Especially, with the concept of fairness and efficiency, the NBS has some good features to develop control algorithms. In this paper, we propose an adaptive voltage scaling algorithm based on the basic concept of the NBS. Based on multiprocessors, the proposed game model can be formulated as follows.

- Players: Processors are assumed to be players.
- Strategies: Each player has a finite number of strategies. The strategy for each player is the voltage level; the DVS processor's voltage level is related to an associated processor speed.
- Utility functions: Each player has a utility function which represents the amount of satisfaction of a player toward the outcome of the game; the higher the value of the utility, the higher satisfaction of the player for that outcome. In the proposed game model, the energy efficiency and timing requirement are considered simultaneously to quantify a player's satisfaction.

To provide a reliable and energy efficient solution for multimedia applications, the utility function should take into account QoS requirements and voltage/energy consumption at the same time. In this paper, two utility functions F_e and F_d are defined by using current system information. Based on the adaptive online manner, these functions for the processor j can dynamically estimate the degree of energy efficiency and QoS level as

$$F_e(c_t) = \left(1 - \frac{S_{p\_j}(c_t)}{MP_j}\right)^2,$$
$$F_d(c_t) = 1 - \left(\frac{1}{S_{p\_j}(c_t)} \times \sum_{i=1}^{n} \frac{R_i(c_t) \times A(i,j)}{fd_i - c_t}\right), \tag{4}$$

where $R_i(c_t)$ and fd_i are the amount of remaining workload at the current time and the time deadline of the service i, respectively.

To provide the best compromise of the F_e and F_d functions, the proposed algorithm uses a well-known weighted-average multiobjective optimization method [12]. Finally, the utility function for player i (UF_i) is defined as

$$UF_i = [(1-\gamma) \times F_e(v_c)] + [\gamma \times F_d(c_t)], \tag{5}$$

where the parameter γ controls the relative weights given to energy efficiency and the time deadline. Under diverse system environments, a fixed value for γ cannot effectively adapt. In this paper, we treat this as an online decision problem and adaptively modify the γ value. When a new task request is rejected, we can put more emphasis on the processor capacity, that is, on the F_d. In this case, a higher value of γ is more suitable. When all recent task requests are accepted, energy efficiency must be considered more seriously. Therefore, the UF should strongly depend on the F_e. In this case, a lower value of γ is more suitable. By considering the mutual-interaction relationship, the value of γ is dynamically adjusted based on the recent task blocking probability (TBP), which is the ratio of rejected tasks to the recently requested n tasks. Therefore, in the proposed model, the current TBP value is assigned to the γ. By using this real-time online monitoring, the system can be more responsive to current system conditions.

Metaphorically speaking, each processor is a member of a team willing to compromise its own objective to gain a total optimal solution–in other words, a Pareto optimal solution. By employing the NBS model, the team players cooperate with each other and make a collective decision. If an agreement among the k players cannot be reached, the payoff that the players will receive is given by the disagreement point $UF^d = (UF_1^d, \ldots, UF_k^d)$, for example, 0 in my system. Finally, the desirable best solution, $UF^* = (UF_1^*, \ldots, UF_k^*)$, is given by

$$UF^* = \arg\max_{1 \le i \le k} \prod_i (UF_i - UF_i^d). \tag{6}$$

In the multiprocessor system, the task scheduler collectively

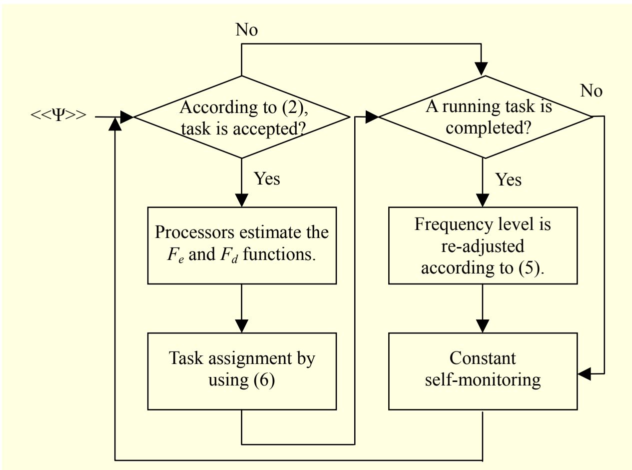


Fig. 1. Flowchart of NBS-based voltage scaling algorithm.

controls a new task allocation for the best system performance. The proposed algorithm transforms the task scheduling problem into the celebrated NBS model. According to (2), (5), and (6), a fair-efficient solution is obtained under widely different and diversified system situations. The flowchart of the NBS-based scheduling algorithm is given in Fig. 1.

Due to the adaptive online approach, the parameters in the proposed formulas are flexible, adaptable, and able to sense dynamic changing current environments. Therefore, an important design principle underlying the proposed algorithm is real-time decision making to solve the NBS model. It is essential in order to be close to the optimized system performance and can ensure efficient system performance.

3. Proposed Management Algorithm

During task execution in multiprocessor systems, the system should adaptively schedule the requested workload. In order to dynamically schedule the accepted tasks in real-time, an adaptive online approach becomes more effective. In this paper, the multiprocessor system constantly monitors current conditions and dynamically changes the control parameters. Therefore, the proposed scheme can adjust adaptively past decision inaccuracies and repeatedly estimate the *UF* value for each processor. With only current information, the system can act to maximize the corresponding utility payoff. To approximate the optimized system performance, the main steps for the proposed online voltage scaling scheme are given next.

Step 1. When a new task request arrives, the task admission algorithm decides whether to accept this task or not according to (2).

Step 2. If a new task is rejected, go to step 5. Otherwise, this task is assigned a specific processor; proceed to step 3.

Step 3. Processors, which are satisfying the acceptance condition, estimate the F_e and F_d functions and get the *UF* based on the possible outcomes.

Step 4. An accepted task is assigned to a specific processor by using (6).

Step 5. When a running task is completed, the processor speed (frequency level) is re-adjusted to maximize the *UF* according to (5).

Step 6. To adaptively estimate the *UF*, the current system *TBP* value is assigned to the γ.

Step 7. The system is constantly self-monitoring the current system situation.

4. Discussion

Performance optimization is one of the most important issues in system management. Up to the present, much optimization research has been conducted. To get the optimal solution in system operations, multiple objective functions are necessary to take into account conflicting system interactions. However, due to model complexity, optimal solution approaches are impractical for implementation with realistic system operations. Unlike conventional optimization methods, we do not focus on trying to get an optimal solution based on the traditional bargaining solution itself, but instead, an adaptive online feedback model is proposed. This approach can significantly reduce computational complexity and overhead. It is an important feature of the proposed scheme.

There are various performance analysis methods. In particular, realistic benchmark analysis is able to provide useful performance information. However, the high cost of these tests is obvious; generating fair benchmark results is very costly. In addition, given that these test results are oriented to specific machines, they have to be limited in scope. In this paper, the main challenge is to design a new task scheduling scheme and to confirm the superiority of the proposed scheme by a fair comparison with the existing schemes. To satisfy this goal, we propose a simulation model; this performance analysis method is a simulated-benchmark, which is a cost effective way to develop and evaluate system designs.

III. Performance Analysis

In this section, we evaluate the performance of the proposed scheme by using a simulation model. Based on this simulation model, we compare the performance of the proposed scheme with other existing schemes [1], [4]. The assumptions for my simulation study are as follows.

- The maximum voltage of the processor is assumed to be 100 MHz at 5 V.
- Time overhead to vary the supply voltage is negligible.
- The arrival process for new task requests is Poisson with rate λ (request/s), and the range of offered load is varied from 0 to 3.0.
- The energy assumptions of a processor are 1.0 nJ/cycle.
- The maximum speed of each processor (MP) is assumed to be the same.
- In order to represent various heterogeneous tasks, eight different task types are assumed based on the processing requirement and required deadline. They are generated with equal probability.
- The total workload and deadlines for different type tasks are exponentially distributed with different means.

Performance measures obtained on the basis of 50 simulation runs are plotted as a function of the offered system load (task requests arrival rate) λ. The simulation model is

Table 2. System parameters used in simulation.

Task	Required total workload (t_c)	
1	2×10^6 kHz	
2	256×10^6 kHz	
3	16×10^6 kHz	
4	64×10^6 kHz	
5	128×10^6 kHz	
6	256×10^6 kHz	
7	384×10^6 kHz	
8	512×10^6 kHz	
Parameter	Value	Description
k	3	Total number of processors in system
MP	100 MHz	Maximum processor capacity of processor
γ	0 to 1	Relative weights given to efficiency and deadline

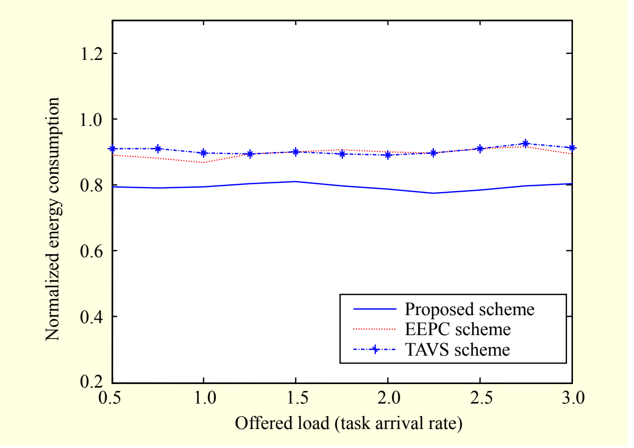


Fig. 2. Normalized energy consumption.

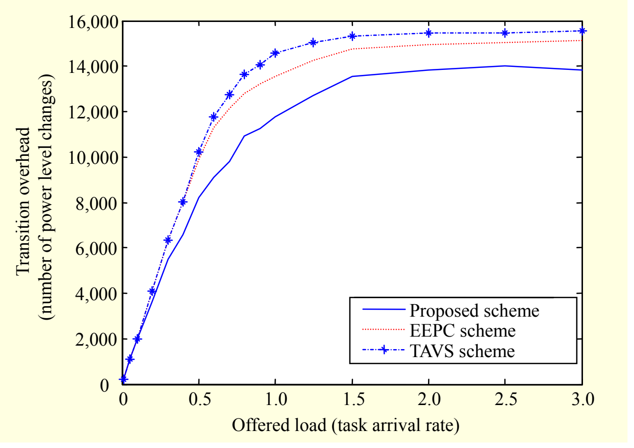


Fig. 3. Transition overhead (number of level changes).

constructed in order to mimic a real system including all necessary elements. For real circumstances and fair comparison, we used the data types, characteristics, and system parameters given in realistic simulation model [2]. Table 2 shows the task types and system parameters used in the simulation. Each task has different task characteristics; its own requirements in terms of desired processor capacity and deadline.

In recent years, some adaptive processor management schemes [1], [4] have been developed in which the main objectives of these schemes are similar to the objectives developed in this paper. To confirm the effectiveness of the proposed scheme, we compare the performance with two existing schemes: the EEPC scheme [1] and TAVS scheme [4]. Even though these existing schemes dynamically control the processor power for efficient system management, there are several disadvantages. First, at the starting time, these schemes require the exact information about the total requested tasks. It is an inappropriate underlying assumption in the system's open nature. Second, they operate running tasks according to fixed system parameters. Under dynamic real-time system environments, control algorithms using static thresholds can cause potential erroneous decisions. Third, these schemes are designed for a specific system performance parameter. Therefore, they cannot maintain well-balanced performance among conflicting criteria under widely different system load distributions.

Figure 2 shows the normalized energy consumption of each scheme. From the simulation results we obtained, it can be seen that all the schemes have similar trends. However, due to the adaptive scheduling strategy, the proposed scheme can reduce energy consumption. Therefore, the proposed scheme performs more efficiently than the other existing schemes from low to heavy system load distributions.

The curves in Fig. 3 indicate the transition overhead in the system. It is shown that for low task arrival rates, the transition overhead is virtually the same for all the schemes. However, as the task arrival rate increases, the proposed scheme can reduce the transition overhead, which substantially helps to decrease energy consumption.

Figure 4 shows the performance comparison in terms of task blocking probability. When the task arrival rate is low, the performance of all the schemes is identical. This is because all the schemes have enough processor capacity to accept new requested tasks. As the task arrival rate increases, the average amount of unused processor capacity decreases. Thus, new task requests are likely to be rejected, and blocking probability increases. However, as λ increases, the proposed scheme has significantly better performance than the other existing schemes.

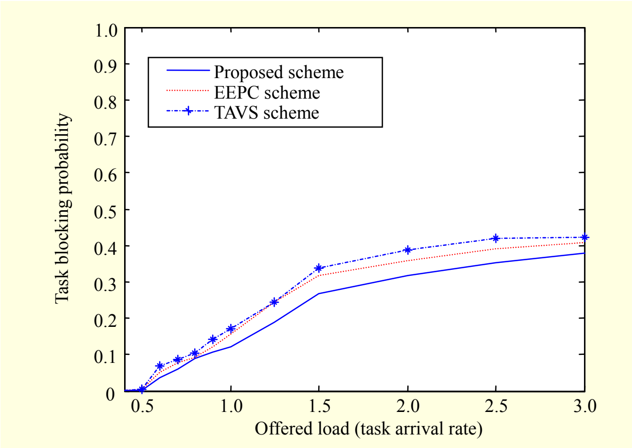


Fig. 4. Task blocking probability (TBP).

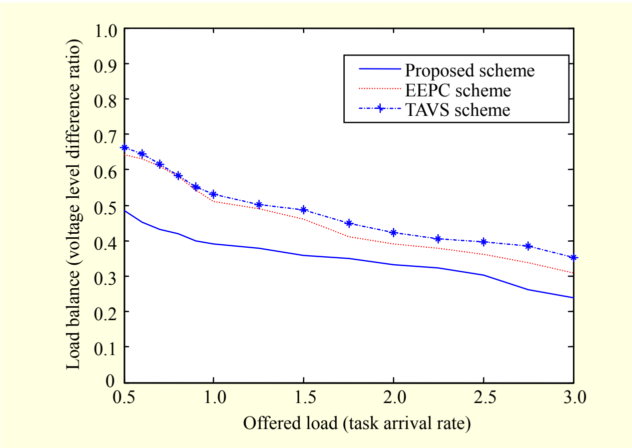


Fig. 5. Load balancing in multiprocessor system.

The curves presented in Fig. 5 indicate the load balancing in the multiprocessor system. In this paper, load balancing is defined as the difference ratio between the maximum and minimum energy consumption among processors. Under various system constraints, the proposed scheme is able to maintain a stable energy consumption balance in the multiprocessor system, which is a highly desirable property for real-time system management.

From the simulation results, it can be seen that the proposed voltage scaling scheme based on the NBS model generally exhibits superior performance compared with the other existing schemes under light to heavy system load distributions.

IV. Conclusion

The dynamic voltage scaling (DVS) technique plays a key role in real-time systems to reduce energy consumption. In this paper, we proposed a new DVS scheme for real-time multiprocessor systems. Based on the NBS approach, the voltage-clock scaling problem is modeled as a cooperative game model. The novelty of the proposed approach is its adaptability, flexibility, and responsiveness to current system conditions. This feature is highly desirable for real-time system management. Simulation results clearly indicate that the proposed scheme provides well-balanced system performance among contradictory criteria, while other existing schemes cannot offer such an attractive system performance.

References

[1] S.U. Khan and C. Ardil, "Energy Efficient Resource Allocation in Distributed Computing Systems," *Int. Conf. Distributed, High-Performance and Grid Computing (DHPGC)*, Singapore, Aug. 2009, pp. 667-673.

[2] S.W. Kim, "Adaptive Online Processor Management Algorithms for Multimedia Data Communication with QoS Sensitivity," *Int. J. Commun. Syst.*, vol. 22, no. 4, 2009, pp. 469-482.

[3] B. Mochocki, X.S. Hu, and G. Quan, "A Realistic Variable Voltage Scheduling Model for Real-Time Applications," *Int. Conf. Computer Aided Design*, 2002, pp. 726-731.

[4] B. Mochocki, X.S. Hu, and G. Quan, "Transition-Overhead-Aware Voltage Scheduling for Fixed-Priority Real-Time Systems," *ACM Trans. Design Autom. Electr. Syst.*, vol. 12, no. 2, 2007, pp. 1-12.

[5] S. Irani, S. Shukla, and R. Gupta, "Online Strategies for Dynamic Power Management in Systems with Multiple Power-Saving States," *ACM Trans. Embedded Computing Syst.*, vol. 2. no. 3, 2003, pp. 325-346.

[6] H. Park and M. Van Der Schaar, "Bargaining Strategies for Networked Multimedia Resource Management," *IEEE Trans. Signal Process.*, vol. 55, no. 7, July 2007, pp. 3496-3511.

[7] J.E. Suris et al., "Cooperative Game Theory for Distributed Spectrum Sharing," *IEEE ICC*, June 2007, pp. 5282-5287.

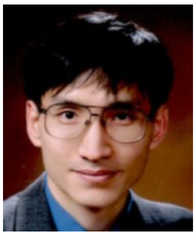
[8] J. Leino, "Applications of Game Theory in Ad Hoc Networks," Master's Thesis, Helsinki University of Technology, 2003.

[9] S. Kim and P.K. Varshney, "An Integrated Adaptive Bandwidth-Management Framework for QoS-Sensitive Multimedia Cellular Networks," *IEEE Trans. Vehicular Technol.*, vol. 53, no. 3, 2004, pp. 835-846.

[10] T.D. Burd and R.W. Brodersen, "Processor Design for Portable Systems," *J. VLSI Signal Process.*, vol. 13, Aug. 1996, pp. 203-221.

[11] D. Ramanathan, S. Irani, and R.K. Gupta, "An Analysis of System Level Power Management Algorithms and Their Effects on Latency," *IEEE Trans. Computer Aided Design Integr. Circuits Syst.*, vol. 21, no. 3, 2002, pp. 291-305.

[12] S. Mehmet and K. Ramazan, "A Comparative Study of Multiobjective Optimization Methods in Structural Design," *Turkish J. Eng. Environmental Sci.*, vol. 25, no. 2, 2001, pp. 69-78.



Sungwook Kim received the BS and MS in computer science from Sogang University, Seoul, Rep. of Korea, in 1993 and 1995, respectively. In 2003, he received the PhD in computer science from Syracuse University, Syracuse, New York, supervised by Prof. Pramod K. Varshney. Since 2006, he has been with Sogang University, where he is currently an associate professor with the Department of Computer Science & Engineering. He is also a research director of the Internet Communication Control Research Laboratory (ICC Lab.). His research interests include resource management, online algorithms, multimedia network management, bandwidth allocation, adaptive QoS control, and game theory for wireless network management.