Secure Real Time Scheduling on Cluster with Energy Minimization

Rudra Pratap Ojha[1], Rama Shankar Yadav[2], and Sarsij Tripathi[2]

[1] Galgotis College of Engineering & Technology
Greater Noida, India
[2] Department of Computer Science & Engineering
Motilal Nehru National Institute of Technology, Allahabad, India
{rpojha,mentor.sarsij}@gmail.com, rsy@mnnit.ac.in

Abstract. Security critical real time applications running over clusters are increasing day by day. These applications some times are battery operated thus they should consume minimum energy while providing both timeliness and security. Conventional real time scheduling algorithm performs poorly when used for scheduling real time application with above said constraints. So it is required to develop a scheduling approach which satisfies the above said constraints i.e. Security, Energy in such applications. We present an approach based on Dynamic Voltage Scaling (DVS) which guarantees at least minimum security (one of the QoS parameter) for the tasks with energy

Keywords: Real time System, Scheduling, Security Services, Energy.

1 Introduction

Real-Time systems span over several domains of computer science. They are defense and space systems, networked multimedia systems, embedded automotive electronics [2] etc. Real-time systems are considered to be those types of systems which have to respond to certain stimuli within a finite and specified delay. Real-Time systems are classified into two types [1] i.e. Hard Real time and Soft Real time.

Many real time applications are using clusters [3][4] for satisfying the need of high computing power where nodes are interconnected through high speed network. These applications have to satisfy timeliness of response and security requirements [13]. Applications and users can be source of security threats to cluster [20]. The security threats to these applications are primarily related to the authentication [24][27], integrity, and confidentiality[25] of application. [5].

Among these applications there are some applications which are battery operated. Therefore these applications have to satisfy security requirements [7] with minimizing energy[34]. For minimizing energy discrete speed level of processors are used based on DVS.

K. Singh, A.K. Awasthi, and R. Mishra (Eds.): QSHINE 2013, LNICST 115, pp. 904–914, 2013.

1.1 Dynamic Voltage Scaling (DVS):

To achieve better energy saving performance *dynamic voltage scaling* (DVS) technique is being used. DVS saves on the energy consumption by dynamically changing the processor supply voltage levels a characteristic supported in many modern processors such as Intel's XScale, Transmeta's Crusoe, and AMD's Duron processors. Processor power consumption can be represented by

$$P \propto \alpha C_L V^2 f \quad (1)$$

Where α is the switching activity, C_L is the load capacitance, V is the supply voltage, and f is the system clock frequency. Due to the quadratic relationship between the voltage and power consumption, reducing voltage can significantly save the power consumption for the processor. On the other hand, however, reducing the voltage supply increases the circuit delay, and thus the processor speed (s), which is given by

$$s \propto \frac{(V - V_T)^2}{V} \quad (2)$$

where V_T is the threshold voltage will decrease. This can lead to deadline miss of task. Many DVS techniques, e.g. [4, 11, 12, 13] have been proposed to reduce the energy consumption for real-time computing system.

Now problem is how to judiciously meet the two conflicting requirements i.e. Security and Energy. We have proposed an approach which ensures minimum security requirement of task while minimizing energy consumption for real time applications.

2 Related Work

Scheduling algorithms are categorized as Offline (static) and Online (dynamic) according to the time when scheduling decisions are taken. In [8] authors have proposed an uniprocessor based algorithm whereas scheduling algorithm for multiprocessor system is given in [9][11][16].In [10] a non preemptive static scheduling algorithm is used whereas dynamic scheduling algorithm for multi-processor system is given in [21][26].These algorithms did well for the real time systems but they are not applicable to applications demanding security and energy constraints.

The work reported in [14, 15] addresses scheduling on clusters with security for non real time applications. In [17, 18] authors have proposed approaches for Cluster and Grid respectively.

R. David & S. Son has given approach for secured real time databases[21]. In [22] authors have proposed a secure concurrency control protocol in real time systems. S.H. Son, C. Chaney, has also proposed security policies in this area [23]. Tao Xie and Xiao Qin proposed a security aware scheduling policy for scheduling real time applications on clusters [5]. Abhishek Songra et. al. has proposed a security criticality based approach as Modified approach for securing Real time applications on clusters (MASA) [6].

Energy-aware computing has been realized as one of the key area for research in real time systems [31]. Energy-driven voltage scheduling algorithms have been developed to reduce system's energy consumption while satisfying the timing constraints [28, 29, 30, 31, 32, 33,]. They are applicable for system having frequency dependent component (such as processors) as resource. Hua and Qu [30] introduce greedy based approach to reduce the energy consumption of the systems by utilizing the concept of DVS.

A lot of work for scheduling with security [19] and energy aware scheduling has been reported separately. But there is no work to best of our knowledge which has tackled the issues simultaneously. Thus, it is required to develop a secure energy aware efficient algorithm. Timeliness, energy and security are often seen as conflicting goals while scheduling a real-time cluster based applications. For such systems we have to find a trade off between the energy as well as security while honoring the deadline of the applications. The algorithm should guarantees the minimum security with lesser energy consumption while meeting the timing constraints.

We have proposed an approach *EMBS* in which scheduling decisions are taken in two phases: first satisfies the minimum -security requirement whereas energy saving is with speed fitting is done in second phase. The results showed the effectiveness of our approach.

3 System Model and Proposed Approach

Cluster is a group of N nodes {N_1, N_2, N_3 ….N_n) connected through a high speed network where real time applications submit task having security requirements at different speed of processor. Real time task is accepted if the cluster can schedule the task so that they complete within their respective deadline and ensures for at least minimum security requirement (related to application) in phase 1. Improvement over energy minimization may be achieved through utilization of available slack in schedule by variation of speed (Phase 2). We consider a task set having n tasks, T = { $\mathcal{T}_1, \mathcal{T}_2$ …… $\mathcal{T}_n$}. Each task $\mathcal{T}_i$ is described with the attribute ($\mathbf{a_i}$,$\mathbf{e_i}$,$\mathbf{d_i}$,$\mathbf{L_i}$,$\mathbf{S_i}$, $\mathbf{V_i}$) where $\mathbf{a_i}$ is the arrival time, $\mathbf{e_i}$ is the execution time, $\mathbf{d_i}$ is the deadline, $\mathbf{L_i}$ is the amount of data to be secured, $\mathbf{S_i}$ is security level requirement, $\mathbf{V_i}$ speed level of processor As snooping, alteration and spoofing are three common attacks on cluster that can be handled by security services such as Authentication, Integrity and Confidentiality. These services incurred computational overhead, which depends upon amount of data secured used for securing these attacks. The overhead model for different security services i.e. Authentication [24], Integrity and Confidentiality [25] has been taken from [5].

3.1 Energy Minimization Based Scheduling

In proposed Energy minimization based scheduling algorithm we used the DVS technique to reduce the energy requirement by dynamically changing processor supply voltage. Many modern processors such as Intel's XScale, Transmeta's Crusoe

and AMD's Duron processor support DVS technique. In this work we assumed that the processors have five discrete voltage levels and the corresponding normalized speed frequencies were (0.2, 0.4, 0.6, 0.8, and 1.0). We assumed that the processor speed is proportional to the supply-voltage and the processor power consumption is a cubic function of the processor speed [35]. Therefore, Energy can be calculated as:

$$\text{Energy } E = e_i (S_i) * S_i^3 \tag{3}$$

Where e_i is the total execution time of the task at the assigned speed S_i. Speed and execution time depends on each other. If speed of the processor increases, execution-time will be decreased and vice-versa.

3.2 System Architecture Used

System architecture used in this architecture consists of 'm' identical nodes connected through a high speed network, where real time task submitted by the 'R number of users is shown in Figure 1 .The schedule queue holds incoming tasks submitted by users. The task submitted by the user is dispatched to the accepted queue if it pass acceptance test (Phase 1). A task is said to be pass acceptance test if task is able to complete in its deadline with minimum security requirement within available speed of the processor. Real time scheduler performs feasibility analysis (Phase 2) of newly accepted task along with other task according waiting for service or partially executed. A task passes feasibility analysis join dispatch queue where speed variation is done to minimize energy. A task fail to satisfy feasibility test join rejected queue and accepted task is dispatched to local queue of nodes in cluster.

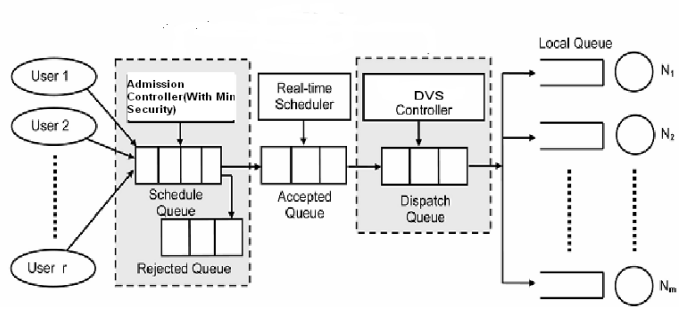

Fig. 1. System Architecture Used

Energy Minimization Based Scheduling Algorithm (EMBS):

1. for each task T_i
2. for each node N_j
3. Compute Utilization of nodes
4. If (Utilization is zero)

4 (i) Select minimum speed level of processor.

(ii).Check feasibility of task T_i at present speed level with minimum security.

(iii). If feasible go to step 10
Else
if speed level is maximum reject the task and
go to step 11
(iv) Increase the speed level and go to step 4(ii).
5. Else Compute Utilization of each each node N_j
6 (i). If U_i>1
(ii) Select a node N_J whose (U_i > 1)
For all task T_n to T_1 at local queue of node N_j
(i) Increase speed of task T_n by one level
(ii) Calculate Utilization
(iii) If (U_j>1) and speed is maximum of T_n and there
is no task in local queue of node reject task T_i goto
Step 11 else goto step 10.
7. Else select a node N_j whose utilization is minimum.
Initialize minimum speed level for task T_i.
8. Calculate Utilization after including task T_i (with current speed level and minimum security) on node N_j
8 (i). If (U_j < 1) go to step 10
Else
If (U > 1) and speed level of task T_i is maximum go
to step 9.
Else
Increase speed level of task T_i and go to step 8.
9 For all tasks T_k to T_1 in local queue of N_j
(i) Increase speed of T_k by one level at node N_j
(ii) Calculate Utilization on node N_j if U_j<1 go to
Step 10.
(iii) If U_j >1 and speed of T_k is maximum and there
is no task left in local queue of node N_j reject
task T_i and go to step 11.
Else
Increase speed of task T_k by one level.
10. Assign the task to node N_j with assigned speed.
11. Rejection ratio =Rejected task / No. of tasks
12. Continue if any task is there in the assigned queue.

Analytical Example

Let there are 4 tasks which have to be scheduled are as

T_1 = (0, 10, 180, 50, 0.4, 0.3, 0.4) T_2 = (10, 10, 250, 100, 0.3, 0.3, 0.5)

T_3 = (20, 10, 350, 50, 0.37, 0.4, 0.4) T_4 = (25, 10, 275, 100, 0.3, 0.4, 0.3)

Given data at maximum speed and minimum security required.
Number of nodes = 2, Communication Overhead=Data size in bytes/ Bandwidth

Case1: Energy consumption in case of Non DVS

Task T_1 arrived at t=0

For task T_1:- Confidentiality Overhead [0.4] = 50/33.75 = 1.481, Integrity Overhead [0.3]=50/12 = 4.167 and Authentication Overhead [0.4] = 90
Communication Overhead.(CO)= 50 KB/100Mbps = 4ms
Min. Security overhead (MSO)=1.481+4.167+90= 95.648
Finish time for task T_1 = 95.648 + 10 + 4 = 109.648
So completion Time (ET) of T_1 is e_1=109.648<150

Similarly for task T_2:-
MSO =2.667+ 8.33 + 90 =100.997
CO= 100 KB/100Mbps = 8ms
So ET for T_2 is e_2 = 100.997 +10+8 =118.997 < 250

For task T_3:-
MSO = 1.481+ 5.139 + 90 = 96.62
CO= 50 KB/100Mbps = 4ms
So ET of T_3 is e_3 = 96.62+10+4=110.62 < 350

For task T_4:-
MSO=2.667+10.277+90 =102.944
CO = 100 KB/100Mbps = 8ms
So ET for T_4 is e_4= 102.944 +10+8= 120.944 < 275

In this case tasks are executed at the maximum speed so energy consumption is
Energy $E = e_i(S_i) * S_i^3$

Let speed (maximum speed) = V
So = $[109.648V^3 + 118.997\ V^3 + 110.62\ V^3 + 120.944\ V^3]$ = $460.209\ V^3$

Case2: Using the concept of DVS (Dynamic Voltage Scaling)
Let Discrete speed levels are 0.2V, 0.4V, 0.6V, 0.8V, 1.0V.

For task T_1:at 0.2V $e_1 = 5*109.647 = 548.24>180$
at 0.4V $e_1 = 274.12 >180$
at 0.6V $e_1 = 182.7467 >180$
at 0.8V $e_1 = 137.06 < 180$
and at t=0 Utilization of nodes are zero so task T_1 dispatched to node N_1 with speed of 0.8V.
For task T_2:-
at 0.2 V $e_2 = 594.985> 250$
at 0.4V $e_2 =297.4925 >250$
at 0.6V $e_2 = 198.328< 250$
at t =10 node is free so task T_2 dispatch to node N_2 with speed 0.6V

For task T_3:-

When task T_3 arrived at t =20
Now find the utilization of nodes
For node N_1 U_1= (137.06-20)/180= 0.6503
For node N_2 = (198.328-10)/250 = 0.7533
Here $U_1<U_2$, so select node N_1

For task T_3 at 0.2V e_3 = 553.1>350
At 0.4v e_3 = 276.55 <350
So total U_1= 0.6503+276.55/350= 1.4404>1
Again increase sped of task T_3 at 0.6 V e_3 = 184.367<350
Total U_1= 0.6503 + 0.527 =1.1773 >1
Again increase the speed of T_3 at 0.8V e_3= 138.275
Total U_1 = 0.6503+ 0.3951 = 1.0454 >1
Again increase the speed of T_3 at speed 1.0V e_3=110.62
Total U_1= 0.6503 + 0.3160 = 0.9663 < 1
So task T_3 dispatch on node N1 with speed of 1.0 V.

For task T_4:-
Now T_4 arrived at t=25
Again calculate utilization:-
Utilization of N_1 U_1 =0.6225+ 0.3018 = 0.9243<1
Utilization of N_2 U_2= (198.328 -15)/250 = 0.73312 <1
Here $U_2< U_1$
So select node N_2
And for task T_4:-
at 0.2V e_4 = 604.72 > 275
at 0.4V e_4 = 302.36 >275
at 0.6 V e_4= 201.5733 < 275
So total U_2 = 0.733 + 0.733 =1.466>1

Again increase the speed of task T_4 at node N_2
At 0.8 V e_4 = 151.180
Total Utilization U_2 = 0.733+ 0.5497 = 1.2827>1
Again increase the speed
At 1.0 V e_4 = 120.944
Total Utilization U2 = 0.733 +0.4398 = 1.1728>1
Now again increase the the speed of task T_2 at node N_2
For task T_2 at 0.8V e_2 = 148.74625 <250
Total U_2 = 0.594985 + 0.4398= 1.034785 >1
Again increase the speed of task T_2 at node N_2
For task T_2 at 1.0V e_2 = 118.997
Total U_2 = 0.475988 +0.4398 = 0.915788<1
So T_4 schedule on node N_2.

Now calculate the consumed energy in case of DVS
E_1 = [137.06 *(0.8 V)^3 +118.997 V^3+110.62*V^3 + 120.944 V^3] = 420.73572 V^3
Save in energy = E – E_1 = 39.47328 V^3
% saving in energy = 8.577 %

4 Simulation and Result

The performance of proposed algorithm has been measured through simulation studies. To reveal the performance improvements gained by our proposed algorithm, we compare the Energy minimization Based Scheduling Algorithm with Non DVS.

Table 1. Simulation Parameters

Parameter	Value (Fixed) - (Varied)
Required Security Service	(Mixed)-(Confidentiality,Integrity,Authentication)
Bandwidth of Network	Fixed 100Mbps
Data size	Varied(10,50,100….500KB)
Processor Speed	Varied(0.2,0.4,0.6,0.8,1.0)

4.1 Simulator and Simulation Parameters

Studied the behavior of the EMBS algorithm under various load conditions by varying different simulation parameters. Like number of tasks, utilization of the task set, mean arrival rate.

Effect of Utilization on Energy. As from the figure 2 we can observe that almost 27% of energy is reduced in lower utilization task set 0.1 to 0.3 as compare to non DVS approach. However, at medium level utilization from 0.3 to 0.6 energy saving is 21 % and at the highest utilization energy saving is nearly 3 to 5%, this is due to the fact that at higher utilization DVS technique has lesser scope to execute task at lower speed. Hence at higher utilization DVS and non DVS have approximately same energy consumption.

Effect of Security Services over Energy. In the figure 3 we can observe that the increase in the minimum security level increases the energy. At minimum security level overhead of the security services becomes lesser due to less overhead.

Execution time of task will be minimum and large amount of slack available for task execution. If slack is available then task can execute at lower speed in case of DVS and energy is saved. When security level increases overhead of the security services and execution time of task increase and slack decreases. Due to decrease in slack, task will execute at higher speed so energy consumption increases. As security level increases at maximum value1.0 overhead and execution time will be maximum, so task will be executed at maximum speed in both the cases.

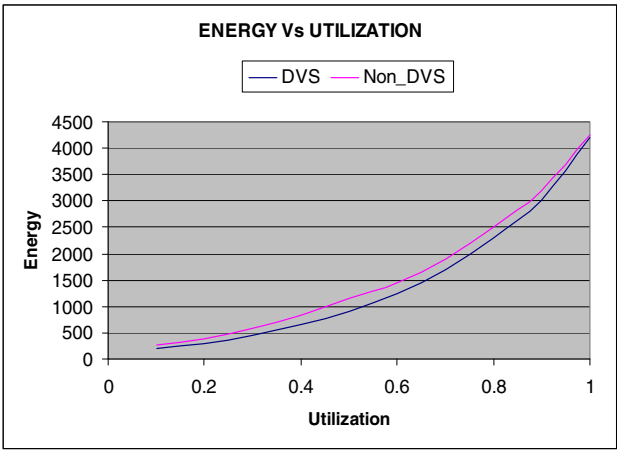

Fig. 2. Energy vs. Utilization

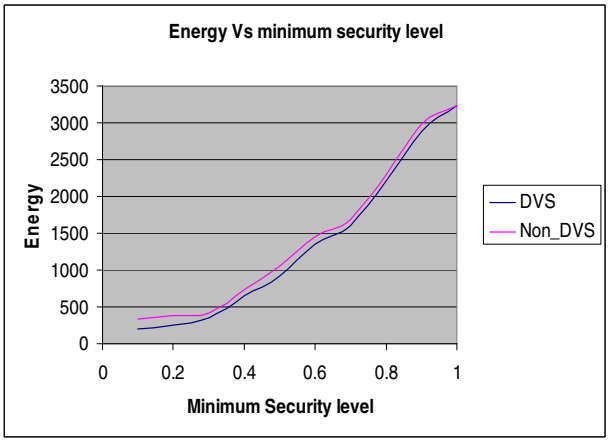

Fig. 3. Energy vs. Minimum security level

5 Conclusion

While emphasizing on security and energy we compromise with the guaranteed completion of the task and which is more important. Thus, our scheduling strategy considers the timing requirements first; the rest of the requirements come under quality. In our work we have seen that energy consumption decreased by using, our proposed approach and ensures minimum security requirement to all release of each task in such a way that receives higher guarantee ratio.

In future we would work on how to optimize security as well as the energy, where energy is minimized and the security would be maximized

References

1. Liu, J.W.S.: Real Time Systems
2. Stewart, D.B.: Courtesy of Embedded System Design
3. Pourzandi, M., Haddad, I., Levert, C., Zakrzewski, M.: A New Architecture for Secure Carrier-Class Clusters. IEEE International Conference on Cluster Computing, 494–497, 23-26 (September 2002)
4. Dessouly, A.A., Ammar, R., Ayman, E.: Scheduling Real Time Parallel Structures on Cluster Computing with Possible Processor Failure. In: IEEE 9th International Symposium on Computers and Communications, 28 June-1 July, vol. 1, pp. 62–67 (2004)
5. Xie, T., Qin, X.: Sheduling Security Aware Real Time Applications on Clusters. IEEE transactions on computers 55(7), 864–879 (2006)
6. Yadav, R.S., Songra, A., Tripathi, S.: Modified approach for securing Real time applications on clusters. International Journal of Security 1(1) (July 2007)
7. Stalling, W.: Cryptography and Network Securities (2003)
8. Parnas, D.L., Xu, J.: Scheduling Processes with Release Times, Deadlines, Precedence and Exclusion Relations. IEEE Transactions on Software Engineering 16(3), 360–369 (1990)
9. Gagne, T., Shepard, M.: A Pre-Run-Time Scheduling Algorithm for Hard Real Time Systems. IEEE Transactions on Software Engineering 17(7), 669–677 (1991)
10. Kavi, Li, W., Krishna: A Non Premptive Scheduling Algorithm for Soft Real Time Systems. Computers & Electrical Engineering 33(1), 12–29 (2007)
11. Dertouzos, M.L., Mok, A.K.: Multi-Processor Online Scheduling of Hard Real- Time Tasks. IEEE Transactions on Software Engineering 15(12), 1497–1506 (1989)
12. Hong, A.J., Tan, X., Towsley, D.: Performance analysis of minimum laxity and earliest deadline scheduling in a real time system. IEEE Transactions on Computers 38(12), 1736–1744 (1989)
13. Martel, C.U., Jeffay, K.: On Non-Preemptive Scheduling of periodic and Sporadic Tasks. In: Proceedings of the 12th IEEE Real-Time Systems Symposium, pp. 129–139. IEEE Computer Society Press, San Antonio (1991)
14. Sterling, T., Savarese, D.: A parallel workstation for scientific computation. In: Proceedings of the 24th International Conference on Parallel Processing, August 14-18. Architecture, vol. I, Urbana-Champain, Illinois (1995)
15. Shiloh, O.T., Barak, A.: Scalable cluster computing with MOSIX for LINUX. In: Proceedings of 5th Annual Linux Expo., pp. 95–100 (May 1999)
16. Genesis, M.H., Goscinski, A.M., Silock, J.: The operating system managing parallelism and providing single system image on cluster. In: Kosch, H., Böszörményi, L., Hellwagner, H. (eds.) Euro-Par 2003. LNCS, vol. 2790, Springer, Heidelberg (2003)
17. Lee, M., Kim, E.J., Yum, K.H.: An overview of security issues in cluster interconnects. In: Sixth IEEE International Symposium on Cluster Computing and the Grid Workshops, May 16-19, vol. 2, p. 9 (2006)
18. Foster, I., Kesselman, C., Tsudik, G., Tuecke, S.: A security architecture for computational grids. In: Proceedings of the 5th ACM Conference on Computer and Communications Security CCS 1998 (1998)
19. Foster, I., Karonis, N.: Managing Security in High Performance Distributed Computations. Journal of Cluster Computing 1(1), 95–107 (1998)
20. Ferrari, A., et al.: A flexible security system for Metacomputing Environments (1999), `http://www.cs.virginia.edu/papers/hpcn99.pdf`

21. David, R., Son, S., Mukkamala, R.: Supporting Security Requirements in Multilevel Real Time Database. In: IEEE Symposium on Security and Privacy, May 8-10, pp. 199–210 (1995)
22. Mukkamala, R., Son, S.: A Secure Concurrency Control Protocol for Real-Time Database. In: IFIP Workshop on Database Security (1995)
23. Son, S.H., Chaney, C., Thomlinson, N.: Partial Security policy to Support Timeliness in Secure Real Time Databases. In: IEEE Symposium on Security and Privacy, May 3-6, pp. 136–147 (1998)
24. Bosselaers, A., Govaerts, R., Vandewalle, J.: Fast Hashing on the Pentium. In: Koblitz, N. (ed.) CRYPTO 1996. LNCS, vol. 1109, pp. 298–312. Springer, Heidelberg (1996)
25. Elkeelany, O., Matalgah, M., Sheikh, K.: Performance Analysis of IPSEC Protocol: Encryption & Authentication. In: IEEE International Conference on Communication, vol. 2, pp. 1164–1168 (2002)
26. Zhang, X., Qu, Y., Xiao, L.: Improving Distributed Workload Performance by Sharing both CPU and Memory Resources. In: International Conference on Distributed Computing Systems (2000)
27. Deepkumara, J., Heys, H.M., Venkatesan, R.: Performance Comparison of Message Authentication Code for Internet protocol Security (2003), http://www.engr.mun.ca/~howard/PAPERS/necec_2003b.pdf
28. Mossé, D., Aydin, H., Childers, B., Melhem, R.: Compiler-Assisted Dynamic Power-Aware Scheduling for Real-Time Applications. In: Workshop on Compiler and OS for Low Power (2000)
29. Qiu, Q., Wu, Q., Pedram, M.: Dynamic Power Management in a Mobile Multimedia System with Guaranteed Quality-of-Service. In: ACM/IEEE Design Automation Conference, pp. 834–839 (2001)
30. Hua, S., Qu, G.: Energy-Efficient Dual-Voltage Soft Real-Time System with (m, k)-Firm Deadline Guarantee. In: CASES 2004, Washington, DC, USA, September 22–25, pp. 116–123 (2004)
31. Niu, L., Quan, G.: Energy-Aware Scheduling for Real-Time Systems With (m; k)-Guarantee., Dept. Comput. Sci. Eng., Univ. South Carolina. Tech. Rep. (2005)
32. Qu, G., Potkonjak, M.: Power Minimization Using System-Level partitioning of Applications with Quality of Service Requirements. In: IEEE/ACM International Conference on Computer-Aided Design, pp. 343–346 (1999)
33. Quan, G., Hu, X.: Energy Efficient Fixed-Priority Scheduling for Real-Time Systems on Variable Voltage Processors. In: 38th IEEE/ACM Design Automation Conference (2001)
34. Niu, L., Quan, G.: System-wide dynamic power management for multimedia portable devices. Accepted by IEEE International Symposium on Multimedia, ISM 2006 (2006)
35. Niu, L., Quan, G.: Energy Minimization for Real-Time Systems With (m; k)-Guarantee. IEEE Transaction on Very Large Scale Integration (VLSI) System 14(7) (July 2006)