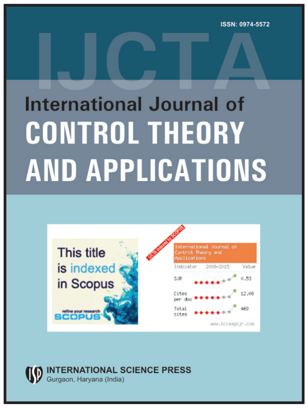

International Journal of Control Theory and Applications

ISSN : 0974-5572

© International Science Press

Volume 9 • Number 43 • 2016

Slack Assessment of the Real Time Scheduling Algorithms

P. Ramesh[a] and Uppu Ramachandraiah[b]

[a]*Assistant Professor, Department of Electronics and Communication Engineering, Hindustan University, India. Email: pramesh@hindustan.ac.in*

[b]*Professor & HOD, Department of Electronics and Instrumentation Engineering, Hindustan University, India. Email: uppur@hindustan.ac.in*

Abstract: Energy efficient system design is critical in global computing environments in both uniprocessor and multi-processor system. Many scheduling algorithms have been proposed in the recent years. Still there is a gap in the scheduling algorithms to avoid the slack existence among the processors. In this scenario the main objective of this work is to evaluate the slack occurrence of the real time scheduling algorithms. Whenever a task completes its execution before the worst case execution time then the slack was introduced in the processor. It increases the energy consumption of the system. Experientially modelled and simulated the Earliest deadline first (EDF), Rate monotonic (RM) and least laxity first (LLF) scheduling algorithms for randomly generated task sets using SimSo simulator. We also verified the Slack occurrence of these real time schedulers. Results concluding that LLF scheduler reduces the slack occurrence compare to EDF and RM.

Keywords: Scheduling, EDF, RM. LLF, Slack.

1. INTRODUCTION

Power management and energy consumption are the two major concerns in the real time embedded systems. Increasing the battery life time is a major challenging issue in the portable sophisticated embedded real time systems. Battery replacement is not possible in all embedded applications. So that energy aware mechanisms are designed in the recent years. New techniques are evolved through the both hardware and software designs. Scheduling is the software implementation method for evolving energy efficient model. The real time applications are split into the hard, soft and firm types based on the timing constraints. Multiprocessor systems are progressed to satisfy the high performance demands in the advanced applications. So many sophisticated algorithms are proposed in the recent years to reduce the energy consumption of the system. The processor consumes the maximum amount of power to execute the tasks. Dynamic voltage scaling (DVS) reduces the energy consumption of the system by controlling the supply voltage. The overall power dissipation consists of both dynamic and static energy. In real-time systems, each task may have timing constraints in the form of earliest start time and deadline that must be satisfied in order to guarantee system correctness and safety [1]. Earliest Deadline First (EDF) and Rate Monotonic (RM) algorithms are the two generally used schedulers for real time systems.

2. LITERATURE SURVEY

Slack arises in a processor when a task finishes its execution earlier than worst-case execution time [2]. DVS is used in the multiprocessor system to decrease the energy consumption by varying the supply voltage [3]. Hybrid optimal scheduler which is a combination of DVFS and DPM for periodic dependent tasks is proposed. The main challenge of the scheduler is to model the idle intervals of the cores. Dynamic power management will address this problem [4]. DP-WRAP scheduler is an extension of DP-FAIR to schedule the sporadic tasks with arbitrary deadlines [5]. Dynamic cache reconfiguration (DCR) is an energy efficient scheduler used to improve the memory performance [6]. By exploiting the characteristic parameters of real-time application tasks, the energy-consciousness of scheduling algorithms and the quality of service of real-time applications can be significantly improved [7]. In Procrastination scheduling, the task execution is extended to the idle interval duration in order to reduce the drain leakage energy [8]. Energy and power are the key concentration in embedded system due to growing technology. In order to reduce the energy consumption dynamic power management and voltage scaling schedulers are mixed [9]. Lucky scheduling is a lottery based scheduler. It is also called as thread scheduling algorithm [10]. Proportionate fair scheduling is an optimal energy efficient scheduling algorithm [11]. The slack occurrence in the processor has to be minimized to reduce the leakage energy.

3. SCHEDULING ALGORITHMS

3.1. Rate Monotonic Scheduling

Rate-monotonic (RM) is a static priority scheduling algorithm here rate means period. The tasks having less period have the highest priority [12]. It follows preemptive mechanism. Processor stops the execution of the current task whenever high priority task arises and it continues current task after completing the high priority task. It is represented in the Figure 1 below. It has the following properties:

1. It doesn't allow precedence constraints
2. Deadline of the task is same as it's period
3. It follows Static mechanism to assign the task priorities
5. It allows Context switching.

T_i	s_i	d_i	p_i	e_i
T1	0	2	3	1
T2	1	3	4	1.2
T3	0	4	4	0.3

S_i: initial start time
d_i: relative deadline
p_i: period i
e_i: maximum execution time

Priority: T1: 1/3; T2: ¼; T3: ¼;
T1: 0, 3, 6, 9, 12, …
T2: 1, 5, 9, 13, …
T3: 0, 4, 8, 12, …

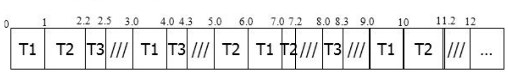

Figure 1: Rate Monotonic Scheduling

3.2. Earliest Deadline First Scheduling

Earliest deadline first (EDF) is comes under the dynamic scheduling category. The task having earliest deadline have high priority [13]. It follows dynamic priority scheme based on the deadline of the tasks. It is an optimal scheduling algorithm. It allows preemptive mechanism. It is represented in the Figure 2 below. It has the following properties:

1. It follows Dynamic mechanism to assign the task priorities
2. It allows Context switching.

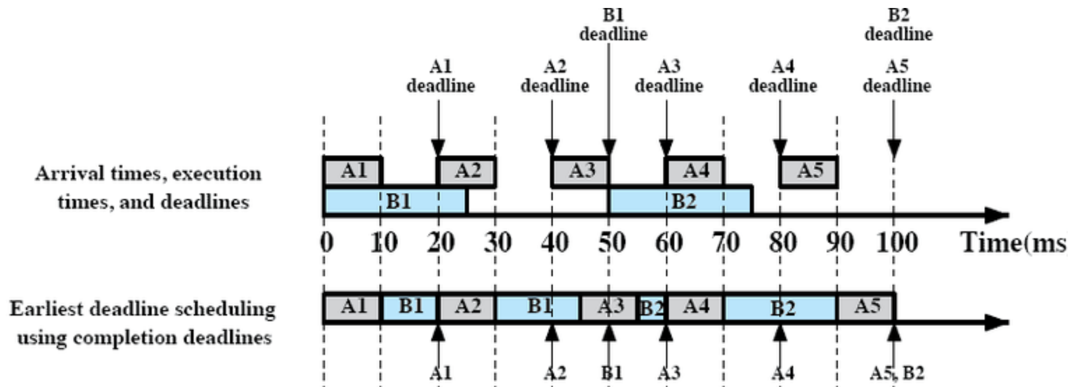

Figure 2: Earliest Deadline First Scheduling

3.3. Least Laxity First Scheduling

It is a dynamic scheduling algorithm. The tasks having least laxity have the high priority. It follows dynamic priority scheme [14]. The urgency of a task to execute before its deadline is called the laxity of a task.

Laxity = Deadline-Remaining computation time.

It has the following properties:

1. It follows Dynamic mechanism to assign the task priorities
2. It allows Context switching.

It allows pre-emptive mechanism. Processor may suffer with the thrashing problem when two or more tasks having the same laxity. Then the processor will spend more time on context switching than the useful work. It will decrease the efficiency of the task. It is also optimal scheduling algorithm like EDF when context switching is ignored.

4. SIMSO SIMULATOR

SimSo is a simulation tool for multiprocessor systems. It will consider overheads into account by using the statistical models. It is a discrete event Simulator and it allows quick simulation and fast prototyping policies using Python [15]. It is open source software which provide graphical user interface. It provides maximum flexibility to the user to run the scheduling algorithms and it displays the processor overhead.

5. RESULTS

The Processor details such as it's the duration cycle, number of cycles, execution time model etc. and task attributes such as type of task, computation time etc. are shown in the Figure 3 below.

The randomly generated task set is simulated by using SimSo simulator for EDF, RM and LLF schedulers. Figure 4 shows scheduling mechanism and slack occurrence for EDF scheduler.

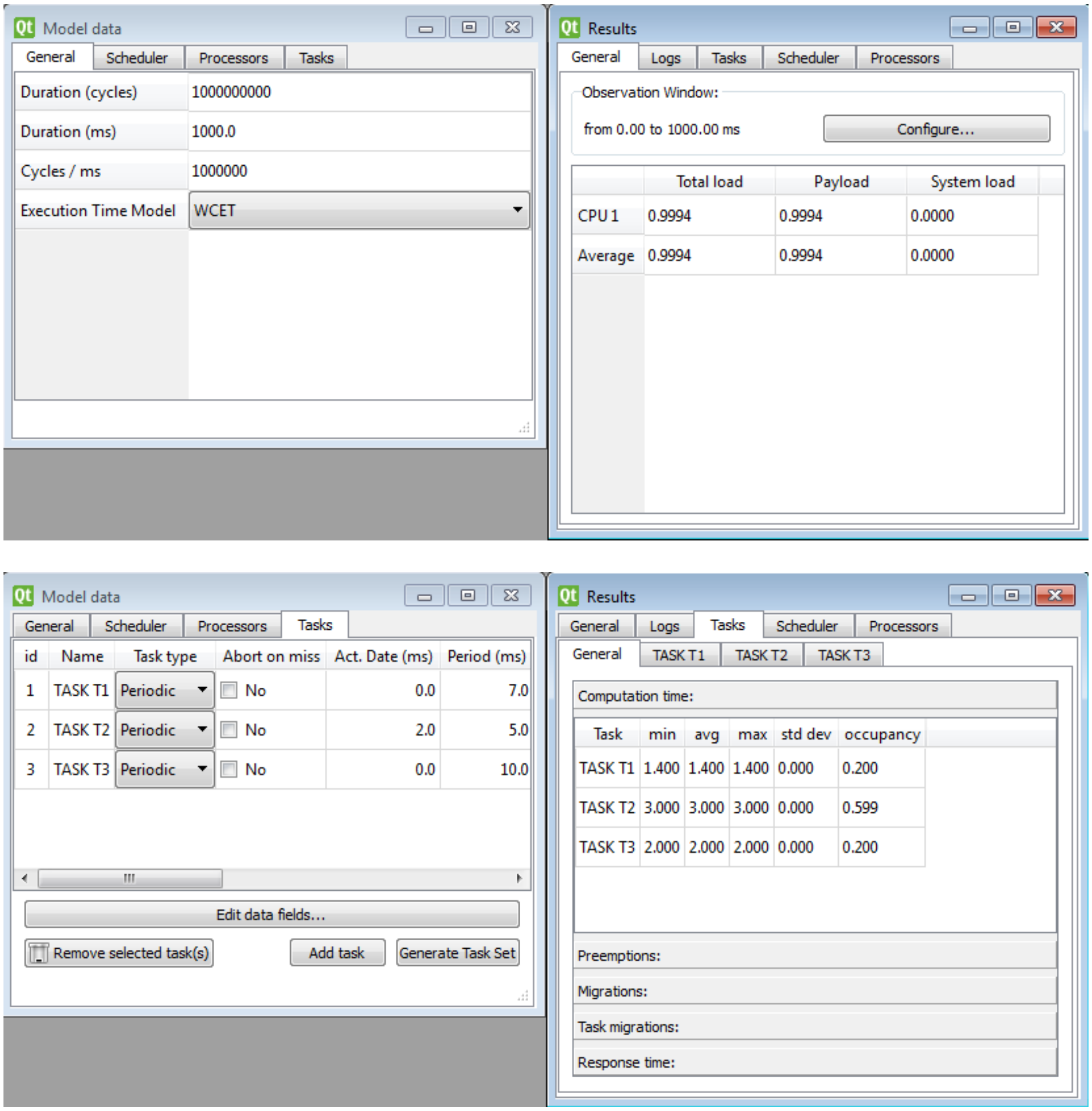

Figure 3: Processor details and task attributes

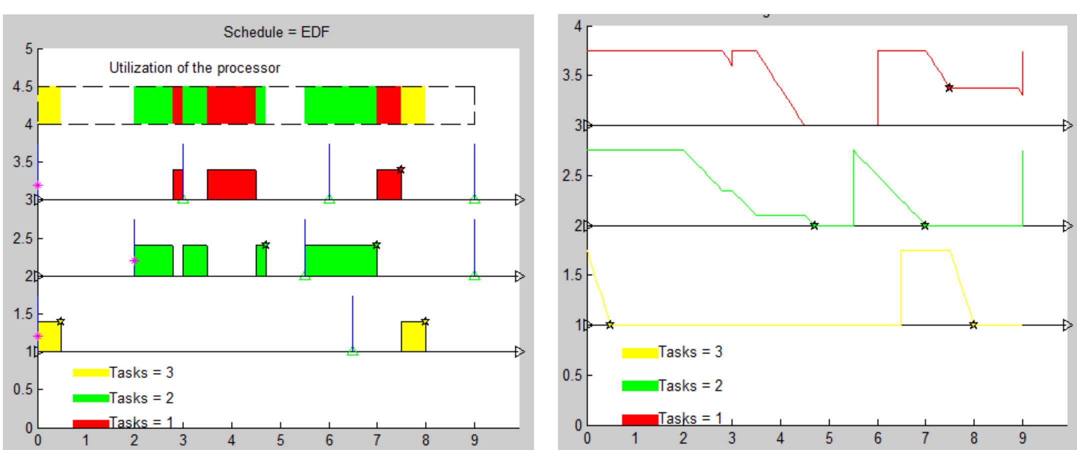

Figure 4: Scheduling mechanism and Slack occurrence for EDF Scheduler

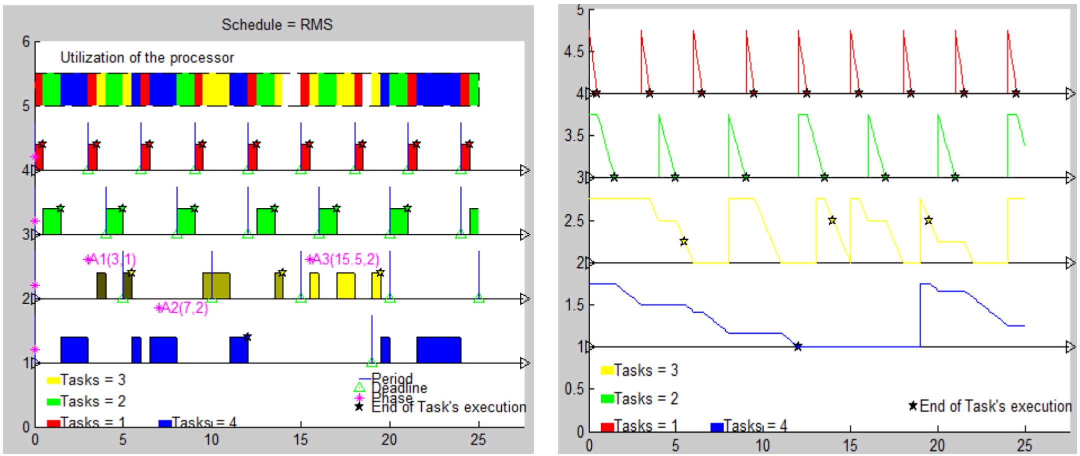

Figure 5: Scheduling mechanism and Slack occurrence for RM Scheduler

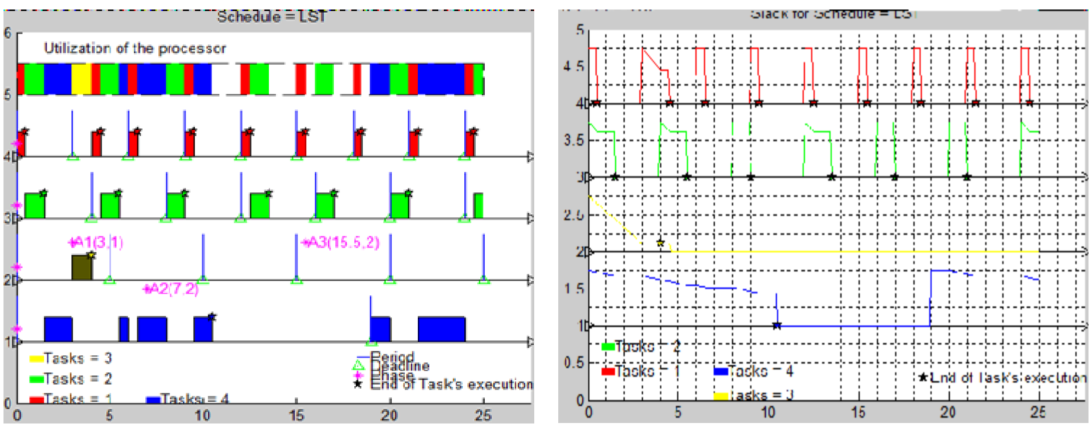

Figure 6: Scheduling mechanism and Slack occurrence for LLF Scheduler

The above Figure 5, Figure 6 represents scheduling mechanism, slack occurrence for RM and LLF schedulers.

Experiential investigation of these algorithms for randomly generated task set clearly indicating that LLF scheduler reduces the slack formation comparing to the EDF and RM schedulers.

6. CONCLUSION

Three scheduling algorithms Earliest deadline first, Rate monotonic and least laxity first are simulated for randomly generated task set through the SimSo simulator. Results indicating that least laxity first scheduler reduces the slack occurrence compare to earliest deadline first and Rate monotonic scheduling schemes. Slack increases the energy consumption of the system. In order to avoid this drawback dynamic slack reclamation or procrastinating algorithms have to be enforced. These advanced mechanisms make the system more energy efficient.

REFERENCES

[1] Liu, C. L., and Layland, J. W. "Scheduling algorithms for multiprogramming in a hard-real time environment", ACM proceedings, 2002.

[2] Weixun Wang, Sanjay Ranka, Prabhat Mishra, "Energy-Aware Dynamic Slack Allocation for Real-Time Multitasking Systems", Sustainable Computing: Informatics and Systems, Elsevier, 2012.

[3] Dakai Zhu, Rami Melhem, And Bruce R. Childers, "Scheduling with Dynamic Voltage/Speed Adjustment Using Slack Reclamation in Multiprocessor Real-Time Systems", IEEE Transactions On Parallel and Distributed Systems, Vol. 14, No. 7, July 2003.

[4] Gang Chen, Kai Huang, Alois Knoll, "Energy Optimization for Real-Time Multiprocessor System-On-Chip with Optimal DVFS and DPM Combination", ACM Transactions on Embedded Computing Systems, June 2013

[5] Greg Leviny, Shelby Funkz, Caitlin Sadowskiy, Ian Pyey, Scott Brandty, "DP-FAIR: A Simple Model for Understanding Optimal Multiprocessor Scheduling", IEEE ECRTS, 2010.

[6] Weixun Wang, Sanjay Ranka, Prabhat Mishra, "Energy-Aware Dynamic Reconfiguration algorithms for Real-Time Multitasking Systems", Sustainable Computing: Informatics and Systems, Elsevier,2011.

[7] Mario Bambagini, Mauro Marinoni, Hakan Aydin, and Giorgio Buttazzo., "Energy-aware scheduling for real-time systems: A survey", ACM Transactions on Embedded Computing Systems, January 2016.

[8] Ravindra Jejurikar, Rajesh Gupta, "Dynamic Slack Reclamation with Procrastination Scheduling in Real-Time Embedded Systems", Proceedings of Design Automation Conference, 2005, pp. 111–116.

[9] D. Shin, J. Kim, "Dynamic voltage scaling of periodic and aperiodic tasks in priority-driven systems", Proceedings of the ASP-DAC, 2004, pp. 653–658.

[10] Daniel Mosse, "Lucky scheduling for energy-efficient heterogeneous multi-core systems", ACM Proceedings of Power aware computing systems conference, 2012.

[11] A. Chandra, M. Adler, and P. Shenoy, "Deadline fair scheduling: Bridging the theory and practice of proportionate-fair scheduling in multiprocessor servers". In Proceedings of the 21st IEEE Real-Time Technology and Applications Symposium, 2001.

[12] J.Goossens, S. Funk, andS. Baruah, "Priority-driven scheduling of periodic task systems on multiprocessors", Real-Time Systems, Vol. 25, No. 2-3, pp. 187–205, 2003.

[13] J. Anderson, V. Bud, and U. C. Devi. "An EDF based scheduling algorithm for multiprocessor soft real-time systems". In IEEE ECRTS, pages 199–208, July 2005.

[14] Dan McNulty, Lena Olson, Markus Peloquin "A Comparison of Scheduling Algorithms for Multi processors", International Conf. on Real-Time and Network Systems, December 13, 2010.

[15] SimSo simulator: http://homepages.laas.fr/mcheramy/simso/.