ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
(An ISO 3297: 2007 Certified Organization)
Website: www.ijircce.com
Vol. 5, Issue 6, June 2017

Performance Analysis of Real Time Task Scheduling Algorithm

Girish S. Thakare[1], Dr. Prashant. R. Deshmukh[2]

Assistant Professor, Dept. of IT, Sipna COE&T, Sant Gadge Baba University, Amravati, Maharashtra, India[1]

Professor, Dept. of CSE, Sipna COE&T, Sant Gadge Baba University, Amravati, Maharashtra, India[2]

ABSTRACT: Real time task are timed based system. The performance parameters in real time system are accuracy and efficiency of the system. Also there are some other criteria on which the performance of the real time system is based. The scheduling algorithm decides the way of execution of a number of tasks ready for the execution in the system. There are numerous algorithms used for different types of real time system. But EDF is the most successful algorithm for real time scheduling. This paper deals with the performance analysis of EDF scheduling algorithm with energy parameter used in the real time system to find efficiency of the system.

KEYWORDS: Earliest deadline first, performance, real time, scheduling.

I. INTRODUCTION

The periodic [1] tasks are used for recurring process in hard real time system. Every task has priority in real time system in several ways. Real-time system works on two types of priority algorithm. The first is fixed priority scheduling algorithm where task with fixed or static priority schedule according to scheduling policy. In this preference is always given to priority of the task and higher priority task always preempts the lower priority task. The second way is keep changing the priority of the task always. This type of system is also known as dynamic priority scheduling. In this type of system the task changes its priority according to the criteria (condition) in the system. . The execution of the higher priority task sometimes may postpone by the lower priority task.

Normally the systems in which periodic activities are take place schedule by fixed priority algorithm such as RM (Rate Monotonic) algorithm. In this algorithm the priority of the task is decided by the rate (period) of the task. The system in which the activities are usually changing used dynamic priority scheduling algorithm EDF (Earliest Deadline First). In this priority of the task is depends on its deadline. The RM scheduling algorithm is easy to implement, efficient and less complex. But sometimes it is not optimal for successful execution of entire tasks in Real Time system. In addition EDF algorithm is very successful in scheduling Real Time task without missing its deadline.

II. RELATED WORK

Previous work in the area of real-time scheduling has mainly focused on the analysis of preemptive scheduling algorithms. The scheduling algorithms are not used in energy point of view. Many researchers has proven that EDF [2] algorithm well suited for scheduling of periodic task with energy concern. In [3] research has presented lazy scheduling the variation of earliest deadline first algorithm, useful for periodic task. The extension of the preemptive problem to multiprocessors was considered in and work with non-preemptive scheduling algorithms [4] has typically been confined to consideration of models where processes are invoked only once, there is a precedence order between the processes, and each process requires only a single unit of computation time and must be completed before a deadline. EDF is an optimal algorithm for single processors.

The EDF is optimal for non-idling situation, test for real-time scheduling policy proposed in [**5**].The author given the context of energy harvesting with no clairvoyance at all relative to both task arrival times and energy production. They proved that EDF is still optimal for online non-idling settings. Vishnu Swaminathan [6] presented a dynamic power management scheme that uses a low-energy EDF scheduler to guarantee real time execution with significant energy savings. The algorithm fails in handling preemption. The application such as mission-critical systems, preemption is an

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
(An ISO 3297: 2007 Certified Organization)
Website: www.ijircce.com
Vol. 5, Issue 6, June 2017

extremely important method to guarantee safety. In [7] presented a novel energy-aware scheduling algorithm, namely ED-H (earliest deadline harvesting), proved to be optimal and appropriate for the scheduling of real time jobs in general. But the proposed algorithm fails for fixed priority environment and also dimensioning i.e. calculation of the smallest capacity for the energy storage unit, the smallest harvester, etc. This kind of analysis refers to sensitivity analysis. It is observed that EDF is always performing well in real time task execution. This paper also deals with the performance analysis of EDF algorithm in concern with energy factor.

III. PROPOSED METHODOLOGY

In this paper, we address the problem of scheduling real-time tasks using EDF on processor to meet its timing and energy constraints. The system consists of a processing unit, rechargeable energy storage such as battery. We propose that EDF has optimal performance of real time system with the energy concern.

We consider a real time task set with four parameters in a renewable energy environment defined by a set of n periodic and independent tasks *{T1, T2……Tn}*. First parameter of task *Ti* is characterized by its arrival time *Ai*, second execution time *Ci*, third its deadline *Di* and fourth energy consumption of the task *Ei*. The execution time *Ci* and the energy consumption *Ei* of a task are fully independent means there is no relation of task execution time and its energy consumption. Consider two tasks *Ti* and *Tj*, we have $Ci < Cj$ and $Ei > Ej$. We will assume that for all task $0 < Ci \leq Di \leq Ti$.

The processor utilization as, $Up = \sum_{i=1}^{n} Ci/Ti$ Eq. (1).

The energy utilization as, $Ue = \sum_{i=1}^{n} Ei/Ti$ Eq. (2).

The system consider here will use an ideal energy storage unit battery that has a nominal capacity *E*.

$E = Emax - Emin.$

The *Emax* is the maximum and *Emin* is minimum level of the battery.

The schedulability test given for EDF is as follows,

$$\sum_{i=1}^{n} \mathrm{Ci/Ti} \leq 1 \qquad \text{Eq. (3).}$$

The task set satisfies the given condition Eq. (3), eligible for the successful execution of the entire task set.

Algorithm

Step 1: Create an initial schedule for all the tasks in ready task queue, where task with earlier deadline has higher priority.
Step 2: Tune the scheduling according to timing and energy constraints of the task.
Step 3: If the schedule of the tasks in not possible due to energy shortage, then delay the task without missing its deadline and compute the slack time.
Step 4: The charging process aims to charge at maximum level provided there is sufficient slack time.
Step 5: Reschedule the task which is postponed due to insufficient energy for scheduling.
Step 8: End.

IV. SIMULATION RESULTS

This section gives the brief overview of execution task set mention in Table 1 with four parameters and arrival time consider as random. Normally task set generation can take place in two ways based on its arrival time i.e. Random Arrival Time and Zero Arrival Time. In Random Arrival Time, task set generation arrival time is explicitly randomly generated but in case of Zero Arrival Time entire set is ready for execution at time 0.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
(An ISO 3297: 2007 Certified Organization)
Website: www.ijircce.com
Vol. 5, Issue 6, June 2017

Table1. Task Information Table 10 tasks with Random Arrival Time.

Sr No.	Arrival Time	Compilation Time	Deadline	Energy
1	5	4	15	10
2	6	3	45	8
3	9	2	30	7
4	7	1	18	6
5	7	1	60	5
6	5	2	10	4
7	6	1	20	3
8	0	5	25	2
9	9	2	120	12
10	3	2	90	8

After executing the 10 task with parameters mentioned in Table 1 with basic requirements such as Energy Rate to charge the battery. Processor Load of given 10 task, and Task scheduling sequence after applying EDF policy. The storage capacity and Hyper Period for duration of task execution.

Energy Rate: 2
Processor Load: 0.9800000000000002
Task Scheduling Sequence [8, 6, 1, 4, 7, 3, 2, 5, 10, 9]
Storage: 25
Hyper Period: 1800

Following is the execution of 10 given task, the complete execution sequence is too large here only first phase is given and end with hyper period value. In every task execution updated value of the storage is also given.

Execution Sequence:
Task 8 is executed from 0 to 5
Remaining Energy in Battery= 23
Updated **Storage**: 33
===
Task 6 is executed from 5 to 7
Remaining Energy in Battery= 29
Updated **Storage**: 33
===
Task 1 is executed from 7 to 11
Remaining Energy in Battery= 23
Updated **Storage**: 31
===
Task 4 is executed from 11 to 12
Remaining Energy in Battery= 25
Updated **Storage**: 27
===
Task 7 is executed from 12 to 13
Remaining Energy in Battery= 24
Updated **Storage**: 26
===
Task 3 is executed from 13 to 15
Remaining Energy in Battery= 19

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Website: www.ijircce.com

Vol. 5, Issue 6, June 2017

Updated **Storage**: 23

==

Task 2 is executed from 15 to 18
Remaining Energy in Battery= 15
Updated **Storage**: 21

==

Task 5 is executed from 18 to 19
Remaining Energy in Battery= 16
Updated **Storage**: 18

==

Task 10 is executed from 19 to 21
Remaining Energy in Battery= 10
Updated **Storage**: 14

==

Task 9 is executed from 21 to 23
Remaining Energy in Battery= 2
Updated **Storage**: 6

==

System is in ideal state from 1795 to 1796

==

System is in ideal state from 1796 to 1797

==

System is in ideal state from 1797 to 1798

==

System is in ideal state from 1798 to 1799

==

System is in ideal state from 1799 to 1800

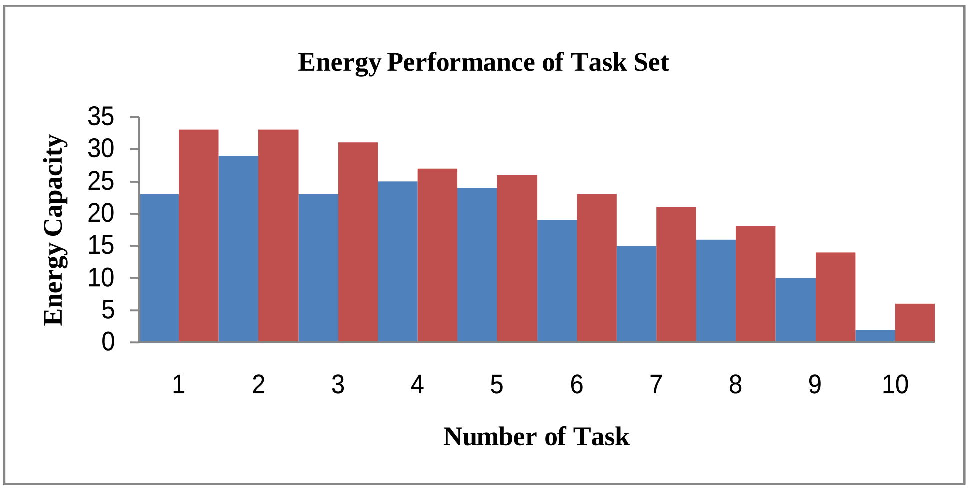

Figure 1. Energy performance of the task set after and before execution.

It is observed that from Figure 1 before executing the task energy is less in the storage but sufficient for task execution shown by blue line. At the next time after task execution energy is gain and storage is updated to new value shown by red lines. The EDF scheduling is successfully executed the entire task without energy shortage. The execution sequence shown as below, the unsuccessful task is zero, the total overload and energy utilized is also mentioned.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering
(An ISO 3297: 2007 Certified Organization)
Website: www.ijircce.com
Vol. 5, Issue 6, June 2017

Successful Tasks: [8, 6, 1, 4, 7, 3, 2, 5, 10, 9, 4, 8, 1, 7, 3, 7, 8, 2, 7, 3, 5, 8, 7, 3, 8, 2, 10, 7, 7, 5, 8, 3, 2, 7, 8, 9, 7, 8, 3, 7, 3, 5, 8, 2, 10, 7, 7, 8, 3, 7, 5, 8, 2, 3, 7, 8, 9, 7, 8, 2, 10, 7, 5, 8, 2, 7, 7, 8, 7, 2, 5, 10, 8, 9, 7, 8, 7, 2, 7, 5, 8, 7, 8, 2, 10, 7, 8, 7, 5, 9, 2, 8, 7, 7, 8, 7, 2, 5, 10, 8, 7, 8, 7, 2, 8, 7, 5, 9, 7, 8, 2, 10, 7, 8, 7, 5, 8, 2, 7, 8, 7, 7, 2, 5, 10, 9, 8, 7, 8, 7, 2, 8, 7, 5, 8, 7, 2, 10, 7, 8, 7, 5, 9, 8, 2, 7, 8, 7, 8, 7, 2, 5, 10, 7, 8, 7, 2, 8, 7, 5, 9, 8, 7, 2, 10, 8, 7, 7, 5, 8, 2, 7, 8, 7, 8, 7, 2, 5, 10, 9, 8, 7, 7, 8, 2, 7, 5, 8, 7, 2, 10, 8, 7, 8, 7, 5, 9, 2, 7, 8, 7, 8, 7, 2, 5, 10, 8, 7, 8, 7, 2, 7, 5, 9, 8, 7, 8, 2, 10, 7, 8, 7, 5, 2, 8, 7, 7, 8, 7, 2, 5, 10, 9, 8, 7, 8, 7, 2, 8, 7, 5, 7, 8, 2, 10, 7, 8, 7, 5, 9, 8, 2, 7, 8, 7, 7, 2, 5, 10, 8, 7, 8, 7, 2, 8, 7, 5, 9, 8, 7, 2, 10, 7, 8, 7, 5, 8, 2, 7, 8, 7]

Unsuccessful Tasks: []
Average Overhead: 845.729537366548
Average Energy Utilized: 1323.

V. CONCLUSION AND FUTURE WORK

The EDF algorithm is optimal for real time scheduling. We test it for energy parameter for task execution. The performance of EDF is found to be improved. In most real time systems periodic task are used. So it is very important to find the optimality of the EDF scheduling algorithm. The EDF algorithm is always efficient is task execution in several aspects such as resource constraint, overloaded conditions. From the simulation results consider here it is clear that EDF also works fine by maintaining sufficient energy every time the next task execution. In future it can be test for other parameters affecting the performance of the EDF algorithm.

REFERENCES

1. LIU C. LAYLAND J. "Scheuling algorithum for multiprogramming in hard real time environment", Journal of ACM 20, 1 , 47-61, 1972.
2. H. Chetto and M. Chetto, "Some Results of the Earliest Deadline Scheduling Algorithm", IEEE Trans. Software Eng., vol. 15, no. 10, pp. 1261-1270, Oct. 1989.
3. C. Moser, D. Brunelli, L. Thiele, and L. Benini, "Real-Time Scheduling for Energy Harvesting Sensor Nodes", Real-Time Systems, vol. 37, no. 3,pp. 233-260, 2007.
4. Marko Bertogna and Sanjoy Baruah, "Uniprocessor Scheduling of Sporadic Task Systems under Preemption Constraints".
5. Maryline Chetto, Audrey Queudet, " A Note on EDF Scheduling for Real-Time Energy Harvesting Systems", IEEE Transactions On Computers, Vol. 63, No. 4, pp 1037-1040, April 2014.
6. Vishnu Swaminathan, Charles B. Schweizer, Krishnendu Chakrabarty and Amil A. Patel, "Experiences in Implementing an Energy-Driven Task Scheduler in RT-Linux", Proceedings of the Eighth IEEE Real-Time and Embedded Technology and Applications Symposium RTAS'02.
7. Maryline Chetto," Optimal Scheduling for Real-Time Jobs in Energy Harvesting Computing Systems", IEEE Transactions On Emerging Topics In Computing Vol 2,No.2, pp 122-133, JUNE 2014.
8. N. Ardsley A. Burns, " REAL-TIME SYSTEM SCHEDULING" , Department of Computer Science, University of York, UK.
9. BARUAH, S., MOK, A., AND ROSIER, L. " Preemptively scheduling hard-real-time sporadic tasks on one processor", In Proceedings of the 11th Real-Time Systems Symposium, Orlando, Florida, IEEE Computer Society Press, pp. 182–190,1990.