3rd National Conference on Green Technology and Science for Sustainable Development
© 2020 IJSRST | Volume 5 | Issue 6 | Print ISSN: 2395-6011 | Online ISSN: 2395-602X
International Journal of Scientific Research in Science and Technology

Comparative Study of Real -Time Task Scheduling Algorithm

Dr. Girish S Thakare*1, Dr Prashant R Deshmukh2
*1Computer Science and Engineering, Sipna COET, Amravati, Maharashtra, India
2Electroincs and Telecommunication, Government COE, Nagpur, Maharashtra, India

ABSTRACT

In this paper we discuss the real time system and its scheduling issues. We put focus on what is real time system? It's scheduling categories and criteria. The role of energy in the scheduling algorithm and type of the energy recently demanded in the currently real time system, also several scheduling algorithm and their comparison. We present our energy driven scheduling algorithm and compare it with existing algorithm. The ambient energy taken in to the consideration here from sun light using solar. In demonstration of scheduling algorithm we implement it for various metrics such as energy overhead, average energy level, and preemption count.

Keywords : Scheduling Algorithm, Energy Overhead, Average Energy Level, Preemption Count.

I. INTRODUCTION

The Future of computer industry growing rapidly with real time system. The energy harvesting in real time system it is going to be the milestone in the industry. But the question arises, what is Real time system? There are many definitions of real time system but the suitable one which we found as,

"Any system in which the time at which output is produced is significant. This is usually because the input corresponds to some movement in the physical world, and the output has to relate to that some movement. The lag from input time to output time must be sufficiently small for acceptable timeliness." — Oxford Dictionary of Computing [1].

This is not the exact description of real time system but covers all the aspect of it. All the different real time systems can be classified into two main categories depending on the nature of timing requirement i.e. the Hard Real Time system and Soft Real Time system [2].

Types of Scheduling Algorithm

The real time system is designed to provide predictability and reliability in the system. The predictability means guaranteed the system execution in design timing constraint and reliability means the ability of the system to perform in given condition for specific period of time. The scheduler is a mechanism by which processor time is allocated to the ready task on given specification. The scheduling algorithm used in real time system divided [3] into following types

1) Clock Driven Scheduling: Clock driven scheduling approaches is offline and already stored in memory. In this task are scheduled based on their time of arrival. The task parameters are normally fixed and do not change during scheduling. This is also known as time driven scheduling.

2) Round Robin Scheduling: Round robin scheduling algorithms repeats task execution after a specific time interval if task execution in not completed with given time period. In this every task is given equal chance to

share processor time. A queue is maintained for incomplete task first in first out basis.

3) Priority Driven Scheduling: In a priority driven scheduling algorithm, tasks or jobs are allocated a priority and scheduled accordingly. The priorities can be allocated based on different

criterion such as earliest deadline first, least laxity first, arrival rate of a task, shortest execution

time first, shortest deadline first etc.

Energy Harvesting System

Energy harvesting system is the new approach for energy source in the system. In this energy gain from the environmental sources such as solar (sunlight), ocean, waves, wind suitable for the application. In this paper we assume the energy source from sunlight using solar, which available almost everywhere around. The solar converts light energy into electricity using photovoltaic cells as shown in figure. In a semi-conductor that is exposed to light, a photon with enough energy can extract an electron and create a gap. Usually, this electron finds quickly another gap to fill, and the energy brought by the photon is dissipated. The principal of a photovoltaic cell is to put the free electrons in one side of the material and the gaps in the other one instead of filling the gaps of only one side. Therefore, a voltage difference and an electric tension appear like in a battery.

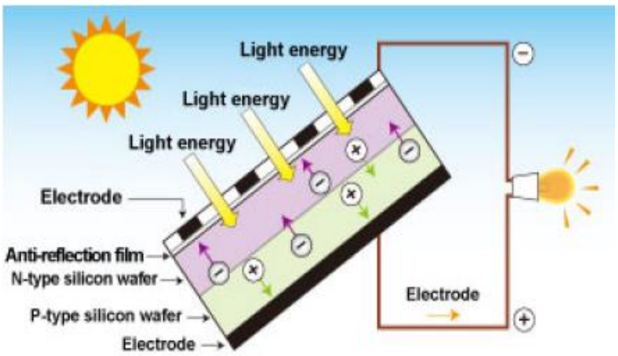

Figure 1. Energy conversion from sunlight using solar.

The efficiency of power conversion of a photovoltaic cell is given by the ratio between the input of photons energy and the output of the yielded energy. A solar cell of 100 cm2 can generate 1.3W if the irradiation is 1000 W/cm2 and the efficiency of such cell is 13 % [4].

II. LITRATURE SURVEY

In recent years many solutions have been proposed for enhancing the energy utilization in the system. Most of them are focusing on the energy consumption of the system but surely this fails in many cases. Later the new approach energy harvesting system emerge which not only optimizes the energy consumption but also manages the energy. Broadly the energy management in the system divided into two categories mainly focusing CPU voltage and frequency and managing the execution by using scheduling techniques.

In dynamic voltage and frequency the processor's speed slowly down to maintain the energy consumption in the system [5, 6]. This approach relies on the fact that the energy consumption in small embedded systems comes mainly from processor's dynamic energy consumption [7]. In some real time scheduling, the slack time and response time can be used to reduce energy consumption by slowing down the processor [8, 9]. The challenges of such an approach are to select the right subset of jobs for which are applied frequency scaling and to select the right CPU frequency to apply for each task. Unfortunately, this was proved to be a Non-deterministic Polynomial-time Hard (NP-Hard) problem in [10], thus, works dealing with this problem focused mainly on proposing adapted heuristics [11,12]. The main drawback of scaling the voltage and frequency is that it increases the risk of transient errors in Complementary Metal–Oxide–Semiconductor (CMOS) logic circuits [13, 14]. To cope with this difficulty, fault tolerance techniques have been used by introducing redundancy. Zhu et al. in [15] proposed a heuristic that uses slack-time to perform DVFS and reserve time for rollback tasks as fault tolerance method.

Energy harvesting scheduling overcomes the problems in the dynamic voltage and frequency scheduling. It collects energy from the environments stored in to the battery, it executes perfect period of task based on the available energy. In energy harvesting system the slack time is used to manage the energy in the system. In [16] multiple slack reclamation technique is given as compared to traditional. The earliest deadline as late as possible (EDL) [17] algorithm executes the task by postponing the task execution to gain the required energy. In [18, 19] earliest deadline with energy guaranteed (EDeg) algorithm check the feasibility of the task execution on available energy. The lazy scheduling algorithm [20, 21] (LSA) algorithm most famous in energy harvesting for energy management.

III. REAL TIME TASK SCHEDULING

The system consider here executes Real Time tasks, with the energy assume to be taken from the external source i.e. ambient energy, as per the requirement of the task. The energy source generates the continuous energy.

A. Task Set

We consider here periodic task set of n tasks T= {ti, i = 1….n}.Each task is independent and execute on uniprocessor system. Every task is characterized by (ai,ci,di,ei), where ai represents arrival time of the task in the ready state, ci is the task execution time for the processor , task deadline is given by di, and the energy of the task represented by ei. The execution of task ci, and energy consumption ei are independent.

B. Energy Source

Here we are considering harvesting energy by solar source, which we assume constant on average for long duration. We predict average energy from the source with rate pr (t) at given time t. This includes all the losses caused by power conversion and charging process.

C. Energy Storage

We consider the appropriate battery as an energy storage with time the capacity given E. The upper bound value Emax is the maximum storage value and Emin is minimum bound on the energy storage E= Emax - Emin. Complete power is used by processor for execution from the battery without any leakage. E (t) is storage capacity at any given time t.

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 ≤ di.

The processor utilization as,

$$Up = \sum_{i=1}^{n} ci/di \quad \text{………………….Eq. (1).}$$

The energy utilization as,

$$Ue = \sum_{i=1}^{n} ei/di \quad \text{…………Eq. (2).}$$

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,

$$\sum_{i=1}^{n} \mathrm{ci/di} \leq 1 \quad \text{………………….Eq. (3).}$$

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

Implementation of Dynamic Slack Calculation Algorithm

System consists of idle time and it is utilized for scheduling i.e. scheduling task without missing its deadline. Idle time is the time when system is ready to execute the task but task queue is empty. In addition to that energy harvesting system uses slack time, particularly use to gain the energy. Idle time is available in the system, but slack time has to claim by adjusting the task execution i.e. postponing the task execution without missing its deadline. Idle time can be used as slack time when it is available in the system. In earliest deadline first algorithm task is executed without slack time as per their deadline shown in Algorithm 1. We have designed new approach implemented on earliest deadline first algorithm based on slack as shown in Algorithm 2.

Algorithm 1: EDF WITHOUT SLACK TIME

```
1. For every ready task Ti
2. Do
        If Tie < E (t)
        Execute task Ti
        Endif
        While (1)
3. Otherwise discard task Ti
4. End.
```

Algorithm 2: EDF-WITH SLACK TIME

```
1. For every ready task Ti
2. Do
        If Tie < E (t)
        Execute task Ti
        Endif
        While (1)
3. Otherwise discard task Ti
        Tis  += X
        E (t) += X * Pr
        Goto step 2
 4. End.
```

The EDF-Slack algorithm works in two modes. In first it checks for the available energy E(t) ,if sufficient then execute the task Ti, otherwise in second mode slack Tis of X unit is calculated without missing deadline and energy is gain for the same period of time. This process is continuing either X units of slack is exhausted or task reaches to its deadline. The is known as dynamic slack calculation because in advance we don't know the amount of slack needed and it will be available on time.

Performance Evaluation and Comparison

We run the simulator for given task sets with three different energy level as Minimum, Average and Maximum. The performance is test on two metrics such as, Success Rate and Average Overhead.

Average Overhead: It is the amounts of time spend by the processor for task execution in one hyper period. This parameter represents the average overhead of the entire scheduling tasks.

Success Rate: It is the percentage of successful tasks.

Table 1. Performance of EDF Scheduling algorithm on Average Overhead.

Sr No.	Number of Task	Processor Utilization	Energy Capacity		
			Min(25)	Avg(50)	Max(100)
1	10	0.98	845.72	835.17	758.96
2	20	0.78	801.09	796.38	704.10
3	30	0.40	2193.02	1943.03	1825.47
4	40	0.31	10768.24	10617.16	10451.21
5	50	0.58	1368.35	976.21	763.35

Table 2. Performance of EDF-Slack Scheduling algorithm on Average Overhead.

Sr No.	Number of Task	Processor Utilization	Energy Capacity		
			Min(25)	Avg(50)	Max(100)
1	10	0.98	848.64	837.17	758.90
2	20	0.78	742.07	796.38	704.10
3	30	0.40	1870.52	1870.26	1863.28
4	40	0.31	9011.76	9278.05	9273.60
5	50	0.58	429.07	418.16	395.0

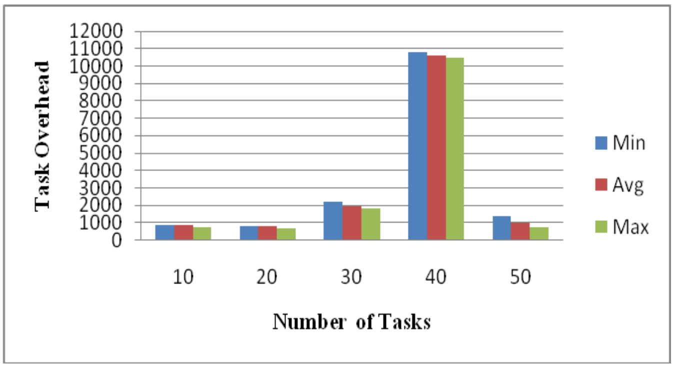

Figure 2. Performance of EDF Scheduling algorithm on Average Overhead.

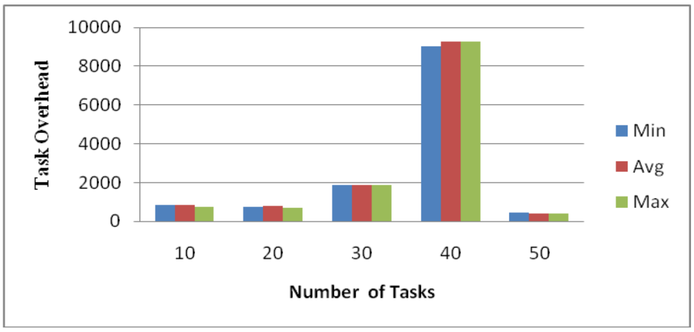

Figure 3. Performance of EDF-Slack Scheduling algorithm on Average Overhead.

Table 3. Performance of EDF Scheduling algorithm on Success Rate (in %).

Sr No.	**Number of Task**	**Processor Utilization**	**Energy Capacity**		
			Min(25)	**Avg(50)**	**Max(100)**
1	10	0.98	90	100	100
2	20	0.78	85	100	100
3	30	0.40	20	37	50
4	40	0.31	13	30	32
5	50	0.58	28	28	40

Table 4. Performance of EDF-Slack Scheduling algorithm on Success Rate. (in %)

Sr No.	**Number of Task**	**Processor Utilization**	**Energy Capacity**		
			Min(25)	**Avg(50)**	**Max(100)**
1	10	0.98	100	100	100
2	20	0.78	100	100	100
3	30	0.40	100	100	100
4	40	0.31	100	100	100
5	50	0.58	100	100	100

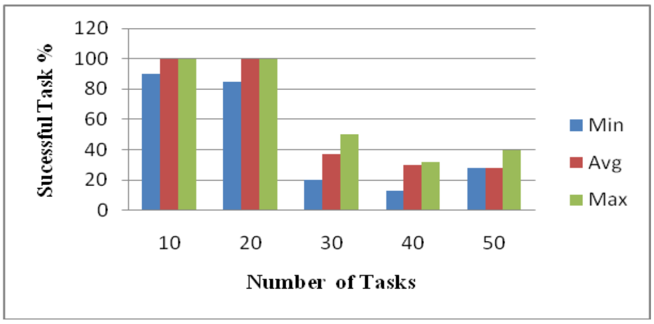

Figure 4. Performance of EDF Scheduling algorithm on Success Rate (in %).

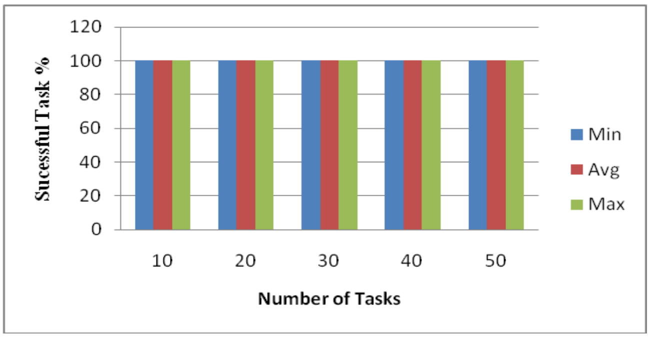

Figure 5. Performance of EDF-Slack Scheduling algorithm on Success Rate (in %).

IV. CONCLUSION

The Implementation of scheduling EDF and EDF-slack algorithm for real time system gives comparative analysis. The EDF algorithm works on task deadlines with no additional slack consideration but in EDF-Slack algorithm the slack is adjusted in away to increase the task execution. This also increases the successful task execution in higher processor load. The EDF-Slack algorithm works efficient with slack in the task execution.

V. REFERENCES

[1] John Daintith Edmund Wright. A dictionary of computing.

[2] A. Burns and A.J.Wellings. Real-Time Systems and Programming Languages: Ada, Real-Time Java and C/Real-Time POSIX. International Computer Science Series. Addison-Wesley, 2009.

[3] J.W.S. Liu. Real-Time Systems. Prentice Hall, 2000.

[4] Simon Mathew et al. “Dye-sensitized solar cells with 13% efficiency achieved through the molecular engineering of porphyrin sensitizers”. In: Nature Chemistry 6 (2014), pages 242–247.

[5] Faruk Yildiz. “Potential Ambient Energy-Harvesting Sources and Techniques”. In: The Journal of Technology Studies (2011).

[6] Trevor Pering, Tom Burd, and Robert Brodersen. “The Simulation and Evaluation of Dynamic Voltage Scaling Algorithms”. In: Proceedings of the International Symposium on Low Power Electronics and Design. ACM, 1998, pages 76–81.

[7] Mark Weiser, Brent Welch, Alan Demers, and Scott Shenker. “Scheduling for Reduced CPU Energy”. In: Proceedings of the USENIX Conference on Operating Systems Design and Implementation. USENIX Association, 1994.

[8] Trevor Pering, Tom Burd, and Robert Brodersen. “The Simulation and Evaluation of Dynamic Voltage Scaling Algorithms”. In: Proceedings of the International Symposium on Low Power Electronics and Design. ACM, 1998, pages 76–81.

[9] Mark Weiser, Brent Welch, Alan Demers, and Scott Shenker. “Scheduling for Reduced CPU Energy”. In: Proceedings of the USENIX Conference on Operating Systems Design and Implementation. USENIX Association, 1994

[10] Dakai Zhu and Hakan Aydin. “Energy Management for Real-time Embedded Systems with Reliability Requirements”. In: Proceedings of the IEEE/ACM International Conference on Computer-aided Design. ACM, 2006, pages 528–534.

[11] Dakai Zhu, Xuan Qi, and Hakan Aydin. “Priority-Monotonic Energy Management for Real-Time Systems with Reliability Requirements”. In: Proceedings of IEEE International Conference on Computer Design. 2007, pages 629–635.

[12] Dakai Zhu and Hakan Aydin. “Reliability-Aware Energy Management for Periodic Real-Time Tasks”. In: IEEE Transactions on Computers (2009).

[13] T. Juhnke and H. Klar. “Calculation of the soft error rate of submicron CMOS logic circuits”. In: IEEE Journal of Solid-State Circuits 30 (1995), pages 830 –834.

[14] J. F. Ziegler. Trends in Electronic Reliability: Effects of Terrestrial Cosmic Rays. http://www.srim.org/SER/SERTrends.htm. technical report. 2014. url: http://www.srim.org/SER/SERTrends.htm.

[15] Dakai Zhu and Hakan Aydin. "Energy Management for Real-time Embedded Systems with Reliability Requirements". In: Proceedings of the IEEE/ACM International Conference on Computer-aided Design. ACM, 2006, pages 528–534.

[16] H. Aydin, R. Melhem, D. Mosse, and Alvarez P. Mejia. Dynamic and aggressive scheduling techniques for power-aware real-time systems. In Proceedings of the 22nd IEEE Real-Time Systems Symposium, pages 95 – 105, dec. 2001.

[17] Maryline Silly. "The EDL Server for Scheduling Periodic and Soft AperiodicTasks with Resource Constraints". In: Real-Time Systems 17.1 (1999), pages 87–111.

[18] Hussein EL Ghor, Maryline Chetto, and Rafic Hage Chehade. "A Real-time Scheduling Framework for Embedded Systems with Environmental Energy Harvesting". In: Computers & Electrical Engineering 37.4 (2011), pages 498–510.

[19] Maryline Chetto. "Optimal Scheduling for Real-Time Jobs in Energy Harvesting Computing Systems". In: IEEE Transactions on Emerging Topics in Computing (2014).

[20] C. Moser, L. Thiele, L. Benini, and D. Brunelli. "Real-Time Scheduling with Regenerative Energy". In: Proceedings of the Euromicro Conference on Real-Time Systems (ECRTS). ECRTS '06. IEEE Computer Society, 2006, pages 261–270.

[21] C. Moser, D. Brunelli, L. Thiele, and L. Benini. "Lazy Scheduling for Energy Harvesting Sensor Nodes". In: Proceedings of the IFIP Working Conference on Distributed and Parallel Embedded Systems. 2006.

[22] C. Moser, D. Brunelli, L. Thiele, and L. Benini. "Real-time scheduling for energy harvesting sensor nodes". In: Real-Time Systems 2007.