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

Efficient use of Idle Time in Real -Time System

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

ABSTRACT

Normally idle time in the system is not considered for the system working. This idle time can be use in real time system task execution. Real-time based embedded systems are widely used everywhere. But the problem in today's real-time embedded system is limited power supply. Many technologies are invented to cope-up energy need of mobile devices system. Energy harvesting easy is free energy solutions for such system. Still there is wide area of research in this field. Up till no best possible solution has focused to satisfy the energy requirement of the real-time system using energy harvesting. This paper discusses an assortment of energy harvesting issues related to the real time system with the implementation of EDF algorithm.

Keywords : Idle-Time, Real–time system, Scheduling

I. INTRODUCTION

Real-time embedded system is the demanding field in the industry. Day by day new products are coming to the market. This includes battery powered mobile devices starting from laptop to mobile phones. The devices may used for various multimedia data processing such as audio, video, images and other types of data. Like traditional desktop system, requires continuous power supply for their functioning. Normally desktop system is provided power supply on the wall. There are some situations providing the power supply from the wall is not possible to mobile devices. Portable electronic devices, in which power is given by batteries, rely on energy efficient power management scheduling algorithm to increase the battery lifetime; while non-portable system need energy efficient schedule to reduce the energy cost. Several strategies are used such as changing the battery and providing continuous power supply from the wall. But for remote devices cost of battery replacement is high and regular power is not also possible. An important strategy to achieve energy saving is via dynamic voltage scaling (DVS) [1], which enable a processor to operate at a range of voltage and frequencies. Meikang Qiu *et al.* [2] design a novel loop scheduling algorithm for real–time applications that produce schedule consuming minimal energy. Energy harvesting with idle time provides solution to this type of system.

Most prior real time systems have relied on continuous power supplied by batteries such as lithium-ion cells. Their disadvantage is that they become depleted, must be periodically replaced or recharged and consequently place hard restrictions on products' usability, lifetime, and cost of ownership. If the battery density is increase but practically it is not safe. There are many concern like given improper use, batteries with extremely high energy densities can become dangerous, explosive devices. In many embedded applications, battery replacement is impractical or has high labour costs associated with maintenance. Harvesting energy in surrounding environment to power embedded systems for the lifespan appears to be the alternative to conventional batteries.

II. LITRATURE REVIEW

Energy can be harvested with use of idle time in the system from different sources and used by several types of system. There are growing needs for energy harvesting capabilities in a variety of applications such as military, health and environment due to the limitation of traditional battery power. The lazy scheduling algorithm [3] (LSA) proposed in as clairvoyant algorithm, assume the knowledge of future availability of environmental power. The algorithm works on offline task schedule. With the aim of achieving a more efficient and conservative management of energy resource, new improved lazy scheduling algorithm (LSA), the solution is energy-aware lazy scheduling algorithm (EA-LSA) [4]. But the computational complexity is not reduce up to the extend and scope is remain for the improvement.

The optimality analysis of EDF (earliest deadline first) 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. Moreover, from a practical point of view, this scheduler has not considered any idle period in the scheduling.

An optimal algorithm PFPASAP (preemptive fixed priority as soon as possible) is design for non concrete task sets, and then they built a necessary and sufficient feasibility condition based on PFPASAP algorithm [6]. This paper proved that the worst case scenario for PFPASAP occurs whenever all tasks are requested simultaneously while the battery is at its minimum level. Algorithm is also performed large scale simulations to evaluate PFPASAP performance and compared it to other algorithms. The experiment showed that the main drawback of PFPASAP is its very large number of preemptions. Moreover, the PFPASAP algorithm is only optimal for non concrete task sets.

Some system uses leakage power consumption factor and the cycle time of the process. But the real-time system need to achieve 100% processor performance and energy utilization, there is huge scope of idle time in efficient energy uses. The system can be work very efficiently with proper utilization of idle time in scheduling policy.

III. PERFORMANCE ANALYSIS OF IDLE-TIME IN SCHEDULING

Real time task scheduling refers to determine the order in which tasks are to be executed. There are two popular approaches: fixed-priority algorithms, including the Rate Monotonic and Deadline Monotonic algorithms and dynamic-priority algorithms, including the Earliest Deadline First algorithm [7]. EDF is an optimal scheduling algorithm in the sense that if a set of tasks can be scheduled by any algorithm, then it can be scheduled by earliest deadline first algorithm.

Liu and Layland [8] proved that a periodic task set with deadlines equal to periods is schedulable by earliest deadline first algorithm if and only if the total processor utilization Up is less than or equal to one i.e.

$$\mathrm{Up} = \sum_{i=1}^{n} \mathrm{Ci}/\mathrm{Ti} \leq 1$$

1. Create an initial schedule for all the tasks in ready task queue, where task with earlier deadline has higher priority.
2. Tune the scheduling according to timing and energy constraints of the task.

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.
4. The charging process aims to charge at maximum level provided there is sufficient slack time.
5. Reschedule the task which is postponed due to insufficient energy for scheduling.

Algorithm: EDF

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.

In the scheduling of real time task various parameters are involved with idle time such as battery capacity, charging rate. These parameters affect the successful execution of all the tasks. This section elaborates the effect of these parameters on the execution.

Now, we consider the factors affecting the performance of the task scheduling.

Processor utilization[9], $U_p = \sum C_i / T_i$

Energy utilization [9], $U_e = \sum E_i / T_i$

Charging rate, pr

Battery capacity E,

$E = E_{max} - E_{min}$

Number of task, $T = \{t_i, i = 1 \ldots n\}$

Successful task execution if $\sum C_i/T_i \le 1$

Successful task execution with energy $E_i \le E(t)$*,*

where , *Ei is the task current energy.*

Normally, required energy for task execution is calculated before execution starts, if the energy available is sufficient the task is executed otherwise discarded. But rather than simply discarding the task, if idle time is inserted to gain the energy and if the task is divided in to the parts then some part of the task is executed successfully and only few is discarded. Also idle time can be inserted in between the execution of task as per the need to take additional time space to gain the energy. Case I: Consider the task set with characteristics by (*ri,ci,di,ei*).

Energy of the task at current time is given by,

$E = E_{max} - E_{min}$

$E(t) < E \quad \text{all time} \quad t$

$E(t) = E - E_i + Pr\,(C_i)$

Now,

ti(8,2,20,12)

E(t)=7-12+2(2)

E(t)=-1 .... Energy is short and hence the task is simply discarded.

Now, if the task energy is calculated by dividing it into parts like,

E(t)=7-6+2(1)

E(t)=3 First part of the task is successfully executed,

E(t)=3-6+2(1)

E(t)=-1 ....Only second part we need to discard not the complete task.

Case II: Consider the task periodic set T1 (2, 5, 7), T2 (1, 7, 9), T3 (3, 10, 11) characterized by computation time, deadline and energy. All tasks are arrived at time zero and scheduled by earliest deadline first algorithm. Processor utilization for three tasks is 0.84 and energy utilization is 0.66. We are considering several cases for scheduling with charging parameter (pr) and storage capacity of the battery (Emax). The energy utilization is 0.66 < charging rate hence we have consider charging rate i.e. pr=1for first case. The battery capacity we assume Emax=10 and Emin=0. The task feasibility is checked before its execution on the energy basis, if the sufficient energy is available in the battery then task is executed successfully and it is feasible. The energy of task at current time with charging rate is calculated by E (t) = E – Ei + pr (Ci) , where E(t) energy needed by task at time t , E energy available in battery, Ei energy required and Ci computation time of the task.

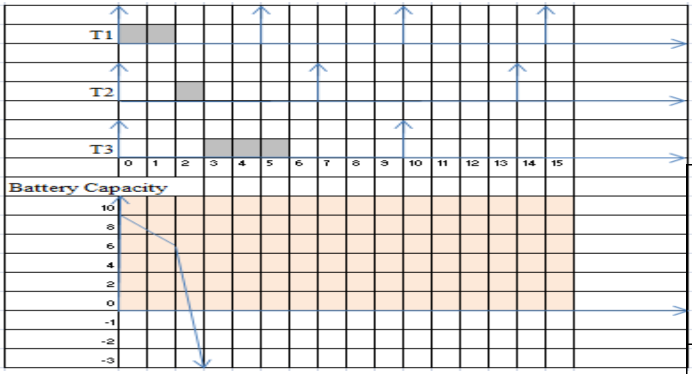

Figure1. Task schedule using EDF with pr=1 and Emax=10 without idle time.

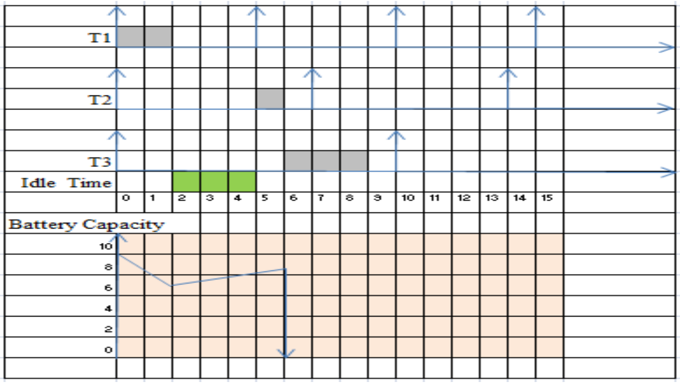

Figure 2. Task schedule using EDF with pr=1 and Emax=10 with idle time.

The figure 1 shows the execution of three periodic tasks with charging rate pr=1 and Emax=10 schedule by EDF without idle time. After energy calculation of task T1, E (0) = 10 – 7 + 1(2) = 5 and for T2 at time 2, E (2) = 5 – 9 + 1(1) = - 3. At the beginning energy is calculated at time 0 for task T1, and it is sufficient and hence task T1 is successfully executed. But for task T2 at time 2, it is found that energy is not sufficient i.e. -3 hence task is simply discarded. In this case idle time is not inserted for additional energy gain for the task. Now if the idle time is inserted at time at time 2 and task T2 is postpone for the execution without missing its deadline as shown in figure 2. Energy is gain during this period and task T2 is successfully executed at time 5, but it empties the battery at low level zero, which discarded the task T3 at time 6.

Average Idle Time: It is the time duration when processor is not performing any scheduling job. This parameter gives the average idle time of the processor. The effect of inserted idle time on several tasks with various processor utilization is shown in Table 1.

Table 1. Performance of EDF Scheduling algorithm on Average Idle Time

SN	Number of Task	Processor Utilization	Hyper Period	Energy Capacity		
				Min (25)	Avg (50)	Max (100)
1	10	0.98	1800	111.00	53.85	114.00
2	20	0.78	1800	55.3	53.85	65.4
3	30	0.40	4800	156.33	156.16	154.6
4	40	0.31	24000	590.8	590.12	590.67
5	50	0.58	27000	539.42	539.21	539.18

IV. CONCLUSION

Adjusting the unused processor space i.e. idle time in the system plays very important role. Normally real time systems are tightly coupled with time and dealing with task execution this idle time is well suitable for gaining the additional energy in the system. The real time task may be extended up to its deadline to decide on additional idle time in the system. This additional idle time is useful for further efficient execution of the system.

V. REFERENCES

[1] Minming Li,"Approximation algorithm for variable voltage processors: Min energy, max throughput and online heuristics", *Theoretical Computer Science,* ELSEVIER, 412, pp. 4074-4080, 2011.

[2] Meikang Qiu, Laurence T. Yamg, Zili Shao, and Edwin H.-M. Sha,, "Dynamic and Leakage Energy Minimization with Soft Real-Time Loop Scheduling and Voltage Assignment ", *IEEE*

transaction on very large scale integration(VLSI) system, Vol. 18, No. 3.March 2010.

[3] Clemens Moser , Davide Brunelli , Lothar Thiele ,Luca Benini, “Real-time scheduling for energy harvesting sensor nodes”. *Published online: 17 July 2007 © Springer Science+Business Media, LLC 2007 Real-Time Syst, 37:* pp 233–260, 2007.

[4] Macro Severini, Stefano Squartini, Francesco Piazza, “Energy-aware lazy scheduling for energy-harvesting sensor nodes”, *Neural Computing and Applications 23:*1899-1908, Springer , 2013.

[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] Yasmina Abdeddaïm, Younès Chandarli, Damien Masson,“The Optimality of PFPASAP Algorithm for Fixed-Priority Energy-Harvesting Real-Time Systems” *Published in ECRTS France,* pp 1-17, 2013.

[7] M. Chetto-Silly,” The EDL server for scheduling periodic and soft aperiodic tasks with resource constraints”, *The International Journal of Time-Critical Computing Systems, 17,* pp 87–111, 1999.

[8] Clemens Moser , Davide Brunelli , Lothar Thiele ,Luca Benini, “Real-time scheduling for energy harvesting sensor nodes”, *Published online: 17 July 2007 © Springer Science+Business Media, LLC 2007 Real-Time Syst,* 37: pp 233–260, 2007.

[9] H Ghor, M Chetto, R Chehade, “ A Nonclairvoyant Real-Time Scheduling for Ambient Energy Harvesting Sensors ”. *Hindawi publishing corporation, International Journal of Distributed Sensors Networks*, Article 732652, 11 pages, volume 2013.