Optimal Real-Time Scheduling Algorithm for harvesting energy environment

Mona Kumari and Ajitesh Kumar
GLA University, Mathura
Ajitesh.kumar@gla.ac.in

Abstract. Now a day, most of the electronic devices are battery operated and the performance of these devices is depend upon battery power. For the concern of feasible scheduling in real time operated system, need to manage battery power. The energy consumption in mobile devices is depends upon power management. In this article we proposed an optimal real time scheduling algorithm with constraint of harvesting energy, where every task set having flexible speed to manage the energy consumption. The experimental result shows that this optimal algorithm allows the task to execute on assigned speed that will reduce the overall energy consumption. Proposed algorithm give better management of harvesting energy and also gives 10- 15% better result in term of available energy after execution of task in compare to existing DVFS algorithm.

1. Introduction
In every real time system all task should be completed within their deadlines and produce the correct result for an example anti-locking break system in a vehicle is a best example of hard real time system in this time computing system the real data is not produce expected result within their deadlines it may cases system fail. Whatever real time system is there may be hard soft or firm is demand to achieve energy neutral operation if its execution requirement can be supported forever despite energy limitations [10]. In today’s environment every devices executing on that platform which have their own battery power and needs real data within a time period and also do not depend upon the power of outside because of the nature of mobility and most of the devices remains beyond the recharge point due to mobility, for example video streaming applications require light weight device that can be movable across the world [11]. So in this type of system those are always in mobility and not to be recharge every time because of unavailability of recharge point it require a strong power management system to increase the battery backup period for ensuring the feasibility of mixed task set. However, in some application like wireless sensor node deployed in remote areas for environment surveillance, the replacement and recharging battery is not possible due to remote areas so such type of system also require harvesting energy for recharge their battery power and power management for battery to increase their life time. For enhancing the battery power the harvesting feature techniques is incorporated in battery [5]. Harvesting technique is very useful where the battery replacement or recharging is costlier. The harvesting energy is the process of producing electrical energy by using renewable energy resources.
available in environment. There are a lot of resources available in nature like solar equipped energy, kinetic energy system thermal energy generator etc. Therefore, the energy generated by such resources is sufficient for any battery operated system and there is no need of battery replacement, only requires best power management policies.

1.1 Some Characteristics of Harvesting Real Time Application:

Those system have the ability to execute the task at lowest standby current to maximize the battery power storage, and in this system it also having the switch on and off ability instantly and having the analog capability for sensor communication and measurement. This system also has to operate on lowest voltage range to maximize the harvest power backup of the battery.

1.2 Harvesting Energy real time System Limitations:

1. Renewable energy resource is not available always it’s depend upon the nature.
2. Energy intensity varying and depends upon the nature of environment like in day time the intensity it too high and in night it may be zero in case of solar energy.
3. Energy storage must be limited and depends upon the battery capacity.

Considering all above constraint, to maximize the possible number of mixed task set to be schedule so that it requires efficient power management.

In this paper we are focusing on scheduling mix periodic task with deadline constraint on single processor with variable speed assignment by using renewable energy resources with having limited power.

In this article we proposed approach for those systems that are always in mobility and having the problem of energy consumption. By using harvesting constraint this paper successfully implements the algorithm where the task executes with low level energy consumption and recharges their battery whenever the battery goes down from their threshold value.

This research article organized as, in section 2, related work with motivational example are discuss, Section 3 shows that the system model and assumption and in section 4 as main section of this paper is present the proposed algorithm with an example and in next section 5 simulation results are discussed and finally in last section we summarize the articles with some references.

2. Related Work and Motivation

Now a day the consumption of energy plays a vital role for mobile devices that depends upon only battery power. There is a lot of work done in this area and most of the researchers focus on either minimizing the energy consumption or somehow maximizing the system performance such as the target achieved with the energy constraints. In such cases, the ability of the recharge energy storage unit is always disregarded with time [9][18].

Rechargeable system has gained a little attention for scheduling real-time tasks on a single processor. The question is arrived how much amounts of energy to reach the feasibility schedule that complete their execution before the task deadline i.e. without wasting of battery power.

Moser et al. [12] emphasizing on scheduling task set with given deadlines for mix task set, that execute on a single preempted processor system that is managed by a rechargeable battery system. The battery power shows availability to be predictable but time-varying in nature. In this paper the author proposed lazy scheduling algorithm (LSA) and for that algorithm the experimental evaluation shows it to be optimal algorithm for the concern of deadline miss ratio of the task set. After deep evaluation we find that this algorithm is variation of the famous earliest deadline first scheduling algorithm, in this
concept the execution of task is start only if the task is ready to execute and having the less deadline time period among all ready tasks and the system is able to run until the deadline of the task to be meet with maximum power consumption.

The actual drawback of this concept is that the algorithm concept says that execute all task with full speed of the system and that can reduce the stored energy in the battery and coming tasks will be missed the deadlines because of unavailability of the battery power.[13] So that we always remember that it’s not to consume total energy for executing task set is proportional to its execution time.

In the article [6] the author gives a scheduling algorithm that is EDF (Earlier Deadline with energy guarantee constrains) where the available tasks are scheduling for execution according to the algorithm it refuses the energy generator in future at the arrival time of tasks.

This algorithm implies on the two basic characteristics of the real time system one is available slack time period and slack energy which are help to the task in the future. The main idea behind this algorithm is to execute the tasks set according to the EDF.[14]

Before schedule a task to be execute, they must ensure that the energy storage is sufficient to execute all future coming tasks. When such type of condition is not verified, the processor has to idle that time so that the battery recharges as much as possible and meet to threshold value. The drawback of the proposed algorithm is the overflow of the battery because of constant speed of the processor, or we say that sometimes underflow condition shown by battery execute a particular task set.

In the paper Allavena et al. [2] shows an off-line scheduling algorithm that uses voltage and frequency selection variables (DVFS) that execute the frame based real time task. In this algorithm the processor allowed to be reducing power consumption by slow down task set execution under their deadline constraints.

2.1. Motivational Example

We have three mixed tasks TS1 (0, 3, 12), TS2 (1, 3, 10), TS3(0, 2, 8) that are required to execute with their deadline constrain. And the system that we consider to schedule this set of mixed tasks is managed by a harvesting energy resources like solar energy. The attribute of tasks TS1, TS2 and TS3 are having release time r(t), worst case execution time and deadline.

So we are going to be assuming that the harvesting energy with a constant rate during in day and which is approximately equal to 2J/sec and the processor that we consider for scheduling the set of tasks can allow three different frequencies or say that speed levels Slow, Si, and Smax. At Slow task is consumed 1J of energy per second, at Si task may be consume 3.5J/sec and at Smax task consume 7.5J/sec. Suppose that the total capacity of energy storage in battery (BC) = 28J, and energy available in store at a starting time t=0 EC (t) = 22J. At t = 0, there are only two task T1 and T3 is ready to execute and as the deadline of T1 is earlier than T3 that are given, so according to scheduling algorithm EDF, priority of T1 is higher than T3. So, at t = 0 T1 start its execution. But before that, we must be check that is there sufficient energy available in battery to execute the task T1 at its maximum speed until its total execution.

\[ ET1(Smax) = 7.5 \times 2 = 15J < 22J = EC(t=0) \]

As the requirement of T1 to complete its execution at maximum speed is less than the available energy, so that T1 execute at Smax until its completion. At t=1 another task T2 is released and as the deadline of T2 is earlier than T1 , therefore T2 preempt T1 .Execute T3 , So that before start of execution firstly checks that energy availability.
EC (t=1) = 22 – 7.5 + 1 = 15.5J

So that the need of energy to run task T2 at maximum speed Smax that is 15.5J which may be lesser than stored energy in battery, so T2 run at Smax up to the completion of task set, so that it complete its execution time at t = 3. Therefore, at t = 3

EC(t) = 22 - 16 + 2 = 8J

Now, T1 task resumes its previous execution, and we find the available energy require to execute without missing its deadline at Smax is 8J which is less than the available energy, therefore, T1 execute at Smax until its completion, and completed at t = 4. So that

At t = 4 EC(t) = 9 - 7 + 1 = 3J

Now the task T3 starts its execution, and as the energy requirement of T3 at Smax is 24J, which is more than the available energy in the battery. Therefore, T3 execute at slow level, T2 takes 12s to complete its execution, which is more than its deadline time.

3. Harvesting Energy Model and Assumptions

For a real time, uni processor system that contains a set of independent Preemptable mix task set T1, T2, T3 ......Tn. For each task Ti has a different attribute:
- Arrival time of task set
- worst case execution time of task set at defined speed
- Energy required computing the task
- Deadline of mixed task set is given

For assignment of the priority of the task set here we use the real time dynamic priority scheduling algorithm that is EDF so according to EDF rule of scheduling the task is assigned the priority of the task on the basis of deadline, lesser the deadline higher the priority and schedule according to the set of independent mix task set.

Here we also use DVS processor that are capable to execute the task set in variable speed levels like Vmin, Vmid and Vmax with the corresponding speed level Smin, Smid and Smax and any task set run at any speed level between Smax and Smin.

\[
\text{Power of Energy Consumption} = C \times S^3 \quad \text{(1)}
\]

Here \( C \) is the capacity of battery and \( S \) is the assigned Speed

The worst case execution time of the mixed task set at different speed assigned level is the sum of execution time interval period and the time taken by its higher priority tasks which one is preempted.

3.1 Source of Harvesting Energy Generation

How we generate the energy? Is a biggest issue for mobile devices so here we use some harvesting energy resources?

The harvesting energy resource is depending upon behavior of environment like solar and wind etc, such type of the system highly varying according to time.

\[
\text{Energy generated by resource} = \text{time interval} \times \text{rate of generation of energy} \quad \text{(2)}
\]
RTS system will have the rate of energy generation that is harvesting energy generation. That is may be equal to 1.5J/sec and may be it should be constant in day time and rate will be zero in night time due to solar system.

3.2 Storage of Harvesting Energy

We assume that we are having maximum storage energy capacity C and if no task is executed it may be reaches to maximum capacity and shows overflow

Consumed energy is always lying in between zero energy and capacity of battery........(3)

For running the task, battery power at a time and the respective energy $E_d$ in between time interval is consumed from the battery where to be stored. That we have following assumption that energy stored at t time will be varied time to time depends upon the energy consumptions.

3.3 Energy Consumption

Energy consumption means battery power may be consumed and depend upon the processor speed to execute the particular task set.

The Energy consumption in time interval is depending upon the power consumption at the same time interval.

Energy consumed maximum when process run at maximum level of processor speed and when it works low speed it consume lesser energy. [14]

4. Proposed Mythology with Example

Harvested energy system mix task set will be executed at lowest possible level processor speed that is require to meet its deadline time period and saved energy should be used for coming task higher level if it requires to any task to meet their deadline or to be avoid to energy overflow due to recharging of the battery.

Every mix task is executed at minimum level of energy even the battery has full power and computes the task at full speed of the processor. Whenever any mix task set should be arrived.

4.1 Two different situations with respect of energy consumption:

4.1.1 First Situation:

If the mix task set requirement is in maximum speed is lesser than the storage available battery energy C, then we start to calculate slack time period for mix task and if slack is greater than zero, means slack is available then reserve a time slot from next start time of a task to deadline and energy equals to the energy required by task set to complete its execution at maximum speed. Now we calculate the lowest possible speed for execution of a task set

For that the available energy in battery will be added with the harvested energy and reduce that consumption energy which are less than the available energy of the battery. The execution must be completed before the deadline of the task set. After executing the mixed task set with assigned lowest speed of the processor for completion of mixed task and available slack time period whatever is earlier. So if mixed task execution is not complete before the next coming task set, then we start execution of remaining task with full of speed on basis of reserved energy in reserve slot.
4.1.2 Second Situation:
If require power for executing task is lesser than the available energy at highest level of speed then we go for lowest possible and feasible speed for a task set until its reach their deadlines.

4.2 Improved Harvesting real Time Scheduling Algorithm (IHA-RTS)

This improved algorithm is helping out the process of battery recharge by using harvesting energy system so that the battery will recharge after its threshold value reaching.

**Step 1:** Ready Queue and reserve queue for harvesting energy should be initialized by zero

**Step 2:** Schedule the task on the basis of EDF algorithm

**Step 3:** If the ready queue is empty then process the task

**Step 4:** If available energy in the battery is greater than zero and the task execution with highest speed is less or equal to the available energy then calculate the slack availability

**Step 5:** If available slack is zero then this task should be entering in the ready queue and run the task in higher speed of the processor. And if the slack is available

**Step 6:** Execute the task as per the scheduling algorithm

**Step 7:** Otherwise, if any other task arrived at a time which is the less than running task

**Step 8:** Their dead line is also less than the running task

**Step 9:** Then we update the ready queue and coming task preempt the running task

**Step 10:** Now we update the energy available with battery by available subtracted by consumed energy

**Step 11:** Now we continue to execute the task in assigned lower speed

**Step 12:** If the energy required by the coming task to execute is lesser or equal to available

**Step 13:** then switch to step 5

**Step 14:** otherwise step 22

**Step 15:** Now we continue to execute the task in assigned lower speed

**Step 16:** And if the task is complete before the deadline then remove the task set from ready queue and reserve queue

**Step 17:** Now saved energy to be freeze and time slot for next coming task

**Step 18:** Now the available battery power is added the reserve and available

**Step 19:** otherwise remaining portion of task set should be run at higher speed level of the processor

**Step 20:** And now consumed energy should be less from the available energy

**Step 21:** now available energy in the battery is C

**Step 22:** Now check the available energy is not zero and energy consumed at higher speed level is greater than available energy then

**Step 23:** Compute the lower possible speed to execute the task by using AFSA (Algorithm for speed assignment)

**Step 24:** Now execute the task at assigned possible speed at the end of the task set

**Step 25:** if available energy is less than or equal to zero

**Step 26:** Then calculate the common available slack period to execute the task in ready queue when energy will be zero

**Step 27:** And if the energy is less and slack availability is also less than zero

**Step 28:** Now recharge the battery and wait for that

**Step 29:** and if the time is greater than coming task then insert the mix task set in the ready queue

**Step 30:** Stop all process
4.3 Algorithm for speed assignment for mix task set (AFSA)

This algorithm will help the IHA-RTS for assignment of the speed level so that we can save the maximum power by adjusting the speed level of the processor to consuming the energy of the battery. This algorithm can also be useful for any real time application for assigning of the speed level to execute the mixed task set to accomplish the deadline parameter of the task.

Step 1: Start by preparing the possible speed level in the process
Step 2: By increasing order of the speed limit it may sort
Step 3: if assigned speed at any level is equal to n level
Step 4: then calculate the energy level of the battery with harvested energy rate is subtracted by consumed energy to execute the task at this level
Step 5: And the energy consumed by this speed level the task should complete the deadline then assign this speed level to the processor
Step 6: Otherwise we increase the one step size on the speed level
Step 7: And again go to the step 4
Step 8: End

This assignment technique works for both periodic as well as aperiodic task set for ant real time battery application.

5. Evaluation and discussion

This approach is implemented and executed by using RTOSX. For verifying this approach, we are going to see this approach performance by using an example. This example is tested and shows the result and compare with the existing approach that shows the improvement over the performance in different parameters. For performance evaluation we are having the remaining energy available in the battery after successfully completion of the task set.

And the second parameter for performance evaluation is acceptance ration of the task set.
Here we check the variation in mix task set load in the system to saving the reaming energy of the battery and meet all requirement of the system.

We are having a three mix task set those are preemptive in nature are T1 with arrival time 0, execution time 2 and their deadline is 10, as same Task T2 having value 0, 3 and with 15 deadline and last task set T3 having attributed 1, 2 and 6. This task is going to be schedule using battery power and that battery rechargeable and managed by harvesting energy resource, solar energy.

Now we assumed some parameters like rate of recharge of battery in day time is 1J/Sec and in night time it may be zero due to solar power.

And we also assume that we are having three level of speed that are the requirement of the AFSA.
And we also assumed that the highest level of the frequency used energy by a constant rate 1J/sec and at any middle level that are uses the 3J/sec and at higher speed level that is 8J/sec.

So that higher speed level used the highest energy consumption and consumes more n more battery power.

So in this case we also assumed the total power availability of the battery is 32J. By using such assumption, we are going to schedule and execute the above mix task set. As we see in our assumption that we are having the sufficient battery backup to execute the whole task set in full of speed level that is 8J/sec but according to our algorithm. If the slack period is available, we never go with higher speed level of the processor, and reserve the time slot of the task with their assigned speed by AFSA algorithm. In all aspect we schedule this task set as;
Fig. 1. Schedule of Mixed Task Set

Fig. 1 Schedule of the Mixed Task set with energy constraint at different level of speed assignment. This task set should be schedule without missing of any task deadline and save the battery power up to 3J that can be used in the future coming tasks that is the motive of this proposed algorithm.

Now we compare the result of our proposed approach IHA-RTS with the Energy aware dynamic voltage and frequency selection (EA-DVFS) [16]. We compare the result in different parameter like acceptance ration of the task and saved remaining energy on the battery. We see in the figure 3 and 4 how much our proposed algorithm gives better performance in compare to exiting algorithm. See in figure 3 we have drawn a comparison by using factor mix set periodic load varying from 0.1 to 1.5 and average remaining battery backup after task set execution varying from 0 to 100.

Fig. 2: Remaining Battery Energy after Task Execution

We observe in this figure after a time if we increase the system load then we decrease the saving battery power average.
Fig. 3: Shows the schedule of Proposed Algorithm and DVFS algorithm

We observe that average saving battery energy better than EA DVFS in fewer loads and same in when we have more task load in the system.

Fig.4. shows the load with acceptance ratio

This experimental result shows that our proposed algorithm is also better in acceptance ration that shows in figure 4 that is varies 0 to 0.5 and 1. Our proposed algorithm gives better performance result that shows in this experiment and the performance decrease in some point when system load increases.

5. Conclusion

In this paper we present an optimal algorithm that works in harvesting energy environment. The experimental result shows that this algorithm allows the task to execute on decided operating speed that will be reduce the overall energy consumption as well as never miss their deadline by using switching speed of the processor and help the recharging of the battery by solar energy system. The example and experimental result shows that this algorithm improves the storage capacity of the battery as well as the acceptance ratio of the task set. This approach performance slowdown when we increase system load. So for future goal is to improve this algorithm so that if we increase the system load then the performance ratio should not be slow down.
References

[1] Agrawal, S., Yadav, R. S. and Ranvijay, 2009. A Pre-emption Control Approach for Energy Aware Fault Tolerant Real Time System, International Journal of Recent Trends in Engineering, 381-386.

[2] El Osta, R., Chetto, M., & El Ghor, H. (2020). Optimal Slack Stealing Servicing for Real-Time Energy Harvesting Systems. The Computer Journal, 63(10), 1537-1546.

[3] Pavić, I., & Džapo, H. (2020, May). Energy-Aware Real-Time Scheduling for Energy-Harvesting Sensor Nodes. In 2020 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) (pp. 1-6). IEEE.

[4] Allavena, A. and Mosse, D., 2001. Scheduling of frame-based embedded systems with rechargeable batteries, Workshop Power Manage Real-time Embedded System.

[5] Bertogna, M. and Baruah, S., 2010. Limited Preemption EDF Scheduling of Sporadic Task Systems, IEEE Transactions on Industrial Informatics, 579 - 591.

[6] Chetto, M. and Zhang, H., 2010. Performance Evaluation of Real-Time Scheduling Heuristics for Energy Harvesting Systems, EEE/ACM Int’l Conference on Cyber, Physical and Social Computing (CPSCom),Green Computing and Communications (GreenCom), 398 - 403.

[7] Dehghan and Maryam, 2010. Adaptive checkpoint placement in energy harvesting real-time systems , 18th Iranian Conference on Electrical Engineering (ICEE), 932 - 937.

[8] Hussein, E. L. G., Chetto, M. and Chehade, R.H., 2011. EH-EDF: An On-line Scheduler for Real-Time Energy Harvesting Systems, 18th IEEE International Conference on Electronics Circuit and System (ICECS), 776-779.

[9] Hussein, E. L. G., Chetto, M. and Chehade, R.H., 2011. A real-time scheduling framework for embedded systems with environmental energy harvesting, Elsevier on Computers and Electrical Engineering 37.

[10] Liu, S., Lu, J., Wu, Q. and Qiu, Q., 2011. Harvesting-Aware Power Management for Real-Time Systems With Renewable Energy, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 1 - 14.

[11] Liu, S., Lu, J., Wu, Q. and Qiu, Q., 2010. Load-matching adaptive task scheduling for energy efficiency in energy harvesting real-time embedded systems, ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED), 325 – 330.

[12] Liu, S., Qiu, Q. and Wu, Q., 2008. Energy Aware Dynamic Voltage and Frequency Selection for Real-Time Systems with Energy Harvesting, conference on Design, Automation and Test in Europe, 236 - 241.

[13] Lu, J. and Qiu, Q, 2011. Scheduling and mapping of periodic tasks on multi-core embedded systems with energy harvesting, Journal, Computers and Electrical Engineering archive, 498 - 510.

[14] Moser, C., Brunelli, D., Thiele, L. and Benini, L., 2007. Real-time scheduling for energy harvesting sensor nodes, Real-Time Syst ; 37(3), 233-260.

[15] Moser, C., Thiele, L., Brunelli, D. and Benini, L., 2007. Adaptive Power Management in Energy Harvesting Systems ,Design, Automation and Test in Europe Conference and Exhibition, 1 - 6.

[16] Niu, L. and Quan, G. 2006. System-Wide Dynamic Power Management for Portable Multimedia Devices, Eighth IEEE International Symposium on Multimedia, 97 -104.

[17] Niu, L. and Quan, G., 2006. Energy minimization for real-time systems with (m,k)-guarantee, IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 717-729.

[18] Paul, A. A. and Pillai, A.S. B., 2011. Reducing the Number of Con- text Switches in Real Time Systems, International Conference on Process Automation, Control and Computing (PACC), 1 - 6.