

Real Time Scheduling for CPU and Hard Disk Requirements-Based Periodic Task with the Aim of Minimizing Energy Consumption

Vahdaneh Kiani¹, Zeynab Mohseni²

¹Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Iran

^{1,2}These authors have contributed equally to this article as co-first authors

E-mail: amohseni.vkiani@gamil.com

Amir Masoud Rahmani

Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Iran

E-mail: rahmani@srbiau.ac.ir

Abstract—In recent years, with an increasing number of requests, energy, power and temperature have been important keys in embedded systems, which decrease the lifetime of both CPUs and hard disks. The energy consumption is an important issue in computer systems, particularly real-time embedded systems. The frequency and the Revolutions Per Minute are major factors in the reduction of energy consumption in both processors and hard disk drives. Therefore, the main goal of this paper is to present a scheduling mechanism for a real time periodic task that can save more energy. This mechanism is based on increasing, as much as possible, the execution time of the CPU and/or the Read/Write time of the hard disk without passing the task deadline. This will be done by dynamically changing the CPU frequency and/or the RPM of hard disk. Our experimental results demonstrate that the proposed algorithm manages to lower energy consumption by an average of 25% and to reduce the number of missed tasks by 80%.

Index Terms—Energy consumption, periodic real time task, execution time, Read/Write time, Revolutions Per Minute, frequency.

I. INTRODUCTION

Nowadays, Dynamic Frequency Scaling (DFS) is one of the widely-used techniques in real time systems to determine the operating frequency of the processor that ensures all real-time restrictions while minimizing the energy consumption. Dynamic Power Management (DPM) is another major technique for energy conservation that is used to reduce power consumption of off-chip devices such as hard disk drives by switching a device from the active to the energy sleep state. During the last decade, much research has been done in order to find an efficient trade-off between energy consumption and performance. Many studies such as [1-10] have dealt with the tasks with only CPU requests, in which they managed to minimize the energy dissipation using

available techniques such as DVS or DVFS, by dynamically changing the CPU frequency or voltage. Other researchers such as [15], employed another technique known as DPM in order to reduce the power consumption of off-chip devices. In addition, each of these studies [12-14], introduced a power consumption model of a storage server with regards to Read and Write times. In few other studies such as [10, 11], DVS & DPM techniques are combined to save more energy consumption. We applied this method in our study. However, due to the importance and extensive use of hard disk as an off-chip device and with the Revolutions Per Minute (RPM) of hard disks in mind, in this article, we presume a task to specifically contain hard disk requests in addition to those of the CPU. Considering both of CPU and hard disk requests and then presenting an optimizing mechanism can help to save more energy.

This will be achieved via dynamically changing the CPU frequency and/or the RPM of hard disk. This decision is, in turn, primarily based upon which of the two aforementioned devices consumes more energy. In this case, if we need to change the CPU execution time, governor sets the ideal frequency according to the available frequency and the remaining time of task. Furthermore, if the hard disk Read/Write(R/W) time needs to be changed, RPM level will be modified according to the remaining time of the task. Thus we have to reach an efficient trade-off between changing the execution time of the CPU and the R/W time of the hard disk according to their energy consumption. An important point that must be considered is that by reducing the frequency of the CPU or the RPM level of the hard disk, the performance ratio is reduced, but in this work, the reduction of performance ratio which is caused by energy consumption optimization is not significant. Due to the fact that our focus is on scheduling a real time task with CPU and hard disk requests and since some articles such as [10] and [11] merely focus on tasks scheduling algorithm without considering how the execution time of CPU and the R/w time of the hard disk should be altered in order to reduce the energy

consumption of each task, this work is novel in optimizing the energy consumption of a task with both CPU & hard disk requests. The rest of the paper is organized as follows. Section 2 presents related works. In section 3, we describe our algorithm. Performance evaluation and conclusions are presented in Sections 4 and 5, respectively.

II. RELATED WORK

A. Techniques for Saving Energy in CPU

In recent years, a significant number of studies have been done in the field of special-purpose systems for scheduling real time tasks, in which DVFS technique was used in order to improve energy or power consumption. Kamga [1] proposed a mechanism for emulating a precise CPU frequency by using the DVFS management in virtualized environments. In [10] the focus is on a scheduling approach towards the sporadic tasks set in the uni-processor system according to DVFS technique. In [5] they presented a DVFS-based algorithm to reduce energy consumption of processors through efficient use of the generated tasks' slack times by an independent scheduler. Laszewski et al. [6] proposed a scheduling algorithm for DVFS-enabled clusters for executing multiple virtual machines in order to reduce power consumption. In [8] they introduced a mechanism for scheduling of real-time tasks on a non-ideal DVS processor with shared resources in order to obtain better energy efficiency.

B. Techniques for Saving Energy in Hard Disk

Many case studies have been done on the reduction of energy consumption of hard disks. In [12], [13] and [14] they focused on presenting mechanisms for reduction of power and energy consumption of hard disks regardless of them being scheduled for a set of real time tasks. Swaminathan et al. [15] presented the scheduling for a set of hard disk real time tasks with I/O requests based on DPM technique. In this paper we proposed a new algorithm for scheduling a real time periodic task with CPU and hard disk requests, in order to lower energy consumption as much as possible.

Also in Kong et al. [11] they addressed the problem of multi-resource energy minimization for frame-based real-time tasks running on a DVS-capable processor of a discrete set of operational modes with multiple off-chip DPM-capable devices.

III. OPTIMIZING ENERGY CONSUMPTION FOR A PERIODIC REAL TIME TASK WITH CPU AND HARD DISK REQUESTS

A. Phase1: Computing Energy Consumption of CPU and Hard Disk and Comparison of their Respective Values

In this section we will present our new algorithm for computing and comparing hard disk and CPU energy consumption. With the aim of reducing energy consumption, the execution time of the CPU and/or the R/W time of the hard disk up until the deadline will be

increased. Based on which of the hard disk or CPU consumes more energy, the execution time of the task changes with one the items below:

- The R/W time of the hard disk increases.
- The execution time of the CPU increases.
- Both R/W time of the hard disk and CPU execution time increase.

The major parameters of the task are described in Table1.

In Figs.1, 2 and 3 CPU Energy Consumption, HDD Energy Consumption and Decision algorithm are shown, respectively. In the first step of algorithm 1, we have calculated CPU power consumption that is denoted in Eq. (1).

1: **Input:** F_{cpu}, V, λ
 2: **Output:** E_{cpu}
 3: Use Eq. (1) to calculate the power consumption of CPU
 4: Use Eq. (2) to calculate the CPU energy consumption

Fig.1. CPUEC algorithm

1: **Input:** RPM
 2: **Output:** E_{HDD}
 3: Use Eq. (3) to calculate the power consumption of HDD
 4: Use Eq. (4) to calculate the hard disk energy consumption

Fig.2. HDDEC algorithm

Table 1. The major parameters of task.

Parameter	Definition
E_{cpu}	CPU Energy consumption(J)
P	Power consumption
E_{HDD}	Hard Disk Drive Energy consumption(J)
R_t	Ready time(ms)
D_t	Deadline time(ms)
RE_t	Remaining time up until the deadline(ms)
EX_t	Extended period of time from the deadline(ms)
F_{cpu}	Frequency of CPU(GHz)
λ	Effective capacitance
V	Voltage of CPU(V)
$F_{min-cpu}$	Minimum frequency of CPU(GHz)
$F_{max-cpu}$	Maximum frequency of CPU(GHz)
E_{tcpu}	CPU Execution time(ms)
RW_{tHDD}	Read and Write time for hard disk drive(ms)
T_t	Transfer time(ms)
AS_t	Average Seek time in hard disk(ms)
RPM	Revolutions Per Minute in hard disk
E_{task}	The execution time of modified task(ms)

$$P = \lambda \times F_{cpu} \times V^2 \quad (1)$$

Where F, λ and V represent processor's working frequency, the effective capacitance, and processor's working voltage, respectively. In this paper for Intel Pentium M Processor model, the maximum frequency is 1.6 GHz and the minimum frequency is 0.6GHz, so CPU energy consumption is defined as:

$$E_{cpu} = P \times E_{t_{cpu}} \quad (2)$$

For calculating hard disk energy consumption, at first, we must compute power consumption that is defined as:

$$P = \text{Number of platters} \times (\text{RPM})^{2.8} \times (\text{Diameter})^{4.6} \quad (3)$$

For Eq. (3)[17] we used HP 300GB-6G-SAS 15k rpm-SFF(2.5inch) model of hard disk drive with 2 platters, four levels for RPM and 2.5inch diameter, so hard disk energy consumption that is denoted in Eq. (4).

$$E_{HDD} = P \times T_t \quad (4)$$

In Fig.3 at first we compared energy consumption of the hard disk and that of the CPU. If energy consumption of the CPU is greater than that of the hard disk, algorithm 4(First Frequency-Last RPM) will be executed, otherwise algorithm 5(First RPM-Last Frequency) will be executed.

- ```

1: Ecpu = CPUEC algorithm
2: EHDD = HDDEC algorithm
3: If EHDD < Ecpu then
4: FF-LR algorithm
5: else If EHDD > Ecpu then
6: FR-LF algorithm
7: end if

```

Fig.3. Decision algorithm

#### B. Phase2: Computing the Optimal Execution Time of the Task Based on Obtained Ideal Frequency and RPM

In this section, the second phase of our proposed algorithm will be described. In this phase after determining which of the two (CPU or hard disk) consumes more energy, based on the above algorithms, either FF-LR algorithm or FR-LF algorithm in Figs. 4 and 5 will be selected.

##### B.1. Energy Consumption of CPU is More than that of the Hard Disk

In Fig.4 first, the frequency of the CPU is set to minimum and the RPM of the hard disk is set to maximum, in other words, the CPU performs with minimum frequency, which causes the execution time of CPU to increase and the hard disk performs with maximum level of RPM, which causes an decrease of the R/W time of the hard disk. Then the execution time of the CPU and the R/W time of the hard disk are calculated with these new amounts according to Eq. (6) and Eq. (9)[16]. If the sum of the execution time of the CPU and the R/W time of the hard disk is less than  $D_t - R_t$  (task is not missed), it means in addition to prolonging the execution time of the CPU, maybe we can also prolong the R/W time of the hard disk, so the remaining time will be calculated from Eq. (5).

$$RE_t = D_t - (E_{t_{cpu}} + RW_{t_{HDD}}) \quad (5)$$

Based on the sum of the  $RE_t$  from Eq. (5), and the

$RW_{t_{HDD}}$ , the new R/W time of the hard disk is calculated. According to the Eq. (6) and the new R/W time of the hard disk, the rotational delay time will be calculated.

$$RW_{t_{HDD}} = RD_t + AS_t + T_t \quad (6)$$

- ```

1: Input: Rt, Dt, Etcpu, RWtHDD, Fcpu, Fmax-cpu, Fmin-cpu,
   ASt, Tt, RPM
2: Output: Ettask
3: RPM = RPMmax
4: Fcpu = Fmin-cpu
5: Use Eq. (6) to calculate RWtHDD
6: Use Eq. (9) to calculate Etcpu
7: If Etcpu + RWtHDD < Dt - Rt then
8:   Use Eq. (5) to calculate REt
9:   RDt = (RWtHDD + REt) - (ASt + Tt)
10:  Use Eq. (7) to calculate RPM
11:  Compare calculated RPM with 4 levels RPM
12:  Assign the first RPM level that is greater than
   calculated RPM
13:  Use Eq. (6) with ideal RPM level to calculate
   RWtHDD
14: else if Etcpu + RWtHDD > Dt - Rt then
15:  Use Eq. (8) to calculate EXt
16:  CPU frequency changes to Fideal (from Eq. (10))
17:  If Fideal > Fmax-cpu then
18:    Fideal = Fmax-cpu
19:  end if
20:  Use Eq. (9) with ideal Fcpu to calculate Etcpu
21: end if
22: Ettask = Etcpu + RWtHDD

```

Fig.4. FF-LR algorithm

The desired RPM will be obtained from Eq. (7).

$$RPM = (\frac{1}{2} \times 60) / RD_t \quad (7)$$

Then this new RPM will be compared with the levels of RPM and the first RPM level that is greater than the calculated RPM will be considered as the desired RPM value. Finally the new R/W time of the hard disk will be calculated with this RPM.

If the sum of the execution time of the CPU and the R/W time of the hard disk is more than $D_t - R_t$ (missed task), the extended period of time will be calculated from the deadline that is defined as:

$$EX_t = (E_{t_{cpu}} + RW_{t_{HDD}}) - D_t \quad (8)$$

The execution time of the CPU when frequency is minimum will be calculated from Eq. (9).

$$E_{t_{cpu}} = \frac{\sum_{i=1}^N CPI_i}{F_{cpu}} \quad (9)$$

Where N and CPI represent the total number of instructions and Clock cycle Per Instruction, respectively. In addition, the execution time of the CPU can be

calculated by the current execution time of the CPU, the minimum frequency and the current frequency. According to calculated $E_{t_{cpu}}$ and EX_t , the ideal CPU frequency will be obtained (Higher frequency) which is defined as follows:

$$F_{ideal} = \frac{F_{min-cpu} \times E_{t_{cpu}}}{E_{t_{cpu}} - EX_t} \quad (10)$$

If this obtained frequency is more than the maximum CPU frequency, the maximum frequency will be selected as ideal. Consequently the new execution time of the CPU is calculated from Eq. (9) according to the new frequency. Finally according to these obtained execution time of the CPU and R/W time of the hard disk, the new execution time of the task will be calculated.

B.II. Energy Consumption of Hard Disk is more than that of the CPU

```

1: Input:  $R_t, D_t, E_{t_{cpu}}, RW_{t_{HDD}}, F_{cpu}, F_{max-cpu}, F_{min-cpu}, AS_t, T_t, RPM$ 
2: Output:  $E_{t_{task}}$ 
3:  $RPM = RPM_{min}$ 
4:  $F_{cpu} = F_{max-cpu}$ 
5: Use Eq. (6) to calculate  $RW_{t_{HDD}}$ 
6: Use Eq. (9) to calculate  $E_{t_{cpu}}$ 
7: If  $E_{t_{cpu}} + RW_{t_{HDD}} < D_t - R_t$  then
8:   Use Eq. (5) to calculate  $RE_t$ 
9:   CPU frequency changes to  $F_{ideal}$  (from Eq. (11))
10: If  $F_{ideal} < F_{min-cpu}$  then
11:    $F_{ideal} = F_{min-cpu}$ 
12: end if
13: Use Eq. (9) with ideal  $F_{cpu}$  to calculate  $E_{t_{cpu}}$ 
14: else if  $E_{t_{cpu}} + RW_{t_{HDD}} > D_t - R_t$  then
15:   Use Eq. (8) to calculate  $EX_t$ 
16:    $RD_t = (RW_{t_{HDD}} - EX_t) - (AS_t + T_t)$ 
17: Use Eq. (7) to calculate RPM
18: Compare calculated RPM with 4 levels RPM
19: Assign the first RPM level that is greater than calculated RPM
20: Use Eq. (6) with ideal RPM level to calculate  $RW_{t_{HDD}}$ 
21: Use Eq. (5) to calculate  $RE_t$ 
22: If  $RE_t > 0$  then
23:   CPU frequency changes to  $F_{ideal}$  (from Eq. (11))
24:   Use Eq. (9) with ideal  $F_{cpu}$  to calculate  $E_{t_{cpu}}$ 
25: end if
26: end if
27:  $E_{t_{task}} = E_{t_{cpu}} + RW_{t_{HDD}}$ 

```

Fig.5. FR-LF algorithm

In Fig.5 first the RPM of the hard disk is set to minimum and the frequency of the CPU is set to maximum. Then the execution time of the CPU and the R/W time of the hard disk are calculated with these new

amounts according to Eq. (6) and Eq. (9). If the sum of the execution time of the CPU and the R/W time of the hard disk is less than the $D_t - R_t$ (task does not miss), it means in addition to the prolongation of the R/W time of the hard disk, we can also prolong the execution time of the CPU, so this remaining time will be calculated from Eq. (5).

According to calculated $E_{t_{cpu}}$ and RE_t , the ideal CPU frequency will be obtained (lower frequency) which is defined as follows:

$$F_{ideal} = \frac{F_{max-cpu} \times E_{t_{cpu}}}{E_{t_{cpu}} + RE_t} \quad (11)$$

If this obtained frequency is less than minimum CPU frequency, minimum frequency will be selected. Finally the new execution time of the CPU will be calculated according to the new frequency.

If the sum of the execution time of the CPU and the R/W time of the hard disk is more than $D_t - R_t$ (missed task), the extended period of time will be calculated from the deadline according to Eq. (8). Then based on subtraction of the $RW_{t_{HDD}}$ from the EX_t , the new R/W time of the hard disk is calculated. According to the Eq. (6) and the new R/W time of the hard disk, the rotational delay time will be calculated. The desired RPM will be obtained from Eq. (7).

As a result, the new R/W time of the hard disk will be calculated according to the ideal RPM level. Finally, according to obtained execution time of the CPU and R/W time of the hard disk, the new execution time of the task will be calculated.

If the task is not initially missed, after applying our algorithms, we will have a task with an execution time which approaches the deadline, therefore the energy consumption of CPU and/or hard disk is reduced as much as possible, otherwise the execution time of the task is decreased until the deadline.

IV. PERFORMANCE EVALUATION

This section presents the results of our proposed algorithm. Table2 summarizes the specifications of CPU and hard disk that are used in our simulation. We used Matlab software for evaluating our proposed algorithm. 50 tasks were randomly generated of which 10% were initially missed. Fig.6 shows the average execution time of the tasks prior and subsequent applying our algorithm. After applying the algorithm, the average execution time of the tasks increases (because most of them were not initially missed). This algorithm could extend the execution time of the CPU and/or the R/W time of the hard disk up until the deadline. Therefore the energy consumption is reduced by about 25%, which is very significant (Fig.7). Also the execution time of each separate task can be seen in Fig.8 prior and subsequent applying the algorithm.

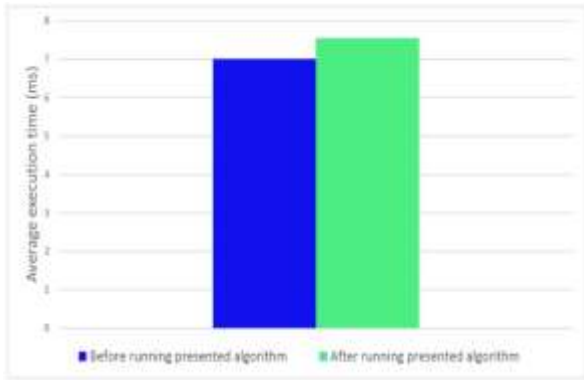


Fig.6. Average execution time of task

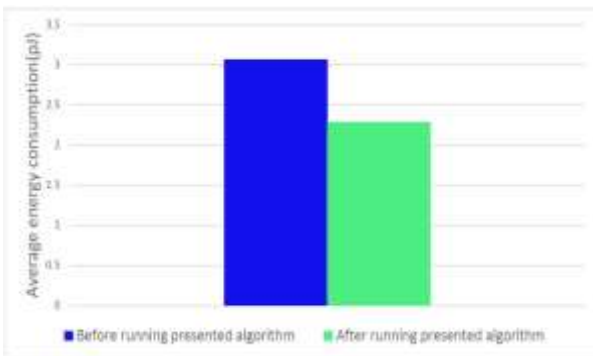


Fig.7. Average energy consumption

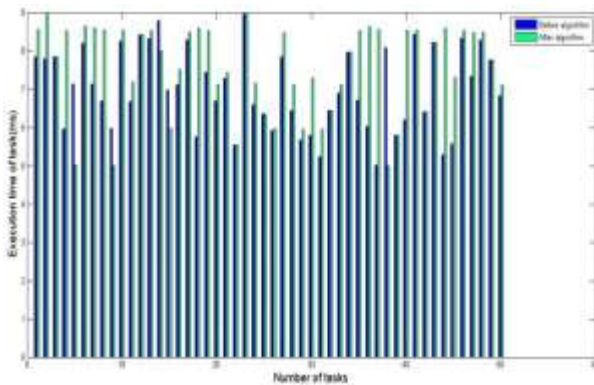


Fig.8. Execution time of tasks

Table 2. Specification of CPU and hard disk drive

HP hard drive	
Name	Value
Capacity	300,000MB
Height	0.62 in (15.6 mm)
Width	2.98 in (75.7 mm)
Depth	4.67 in (118.7 mm)
Interface	SAS
Transfer Rate Synchronous(Maximum)	6 Gb/sec
Physical Configuration	Bytes/Sector 512 Logical Blocks 585,937,500 Rotational Speed: 5400(Min), 7200, 10,000 and 15,000(Max)rpm
Operating Temperature(System Inlet Air Temperature)	50 °to 95 °F (10 °to 35 °C)
Number of Platters	2
Intel Pentium M Processor	
Name	Value
High Mode(HFM) Frequency	Frequency 1.6 GHz Voltage 1.484V TDP(Thermal Design Power) 24.5w
Low Mode(LFM) Frequency	Frequency 600 MHz Voltage 0.956V TDP(Thermal Design Power) 6w

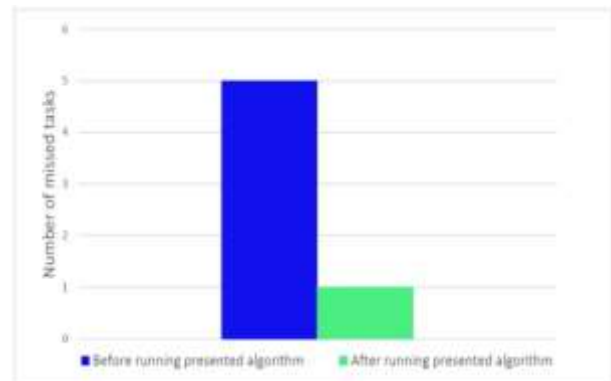
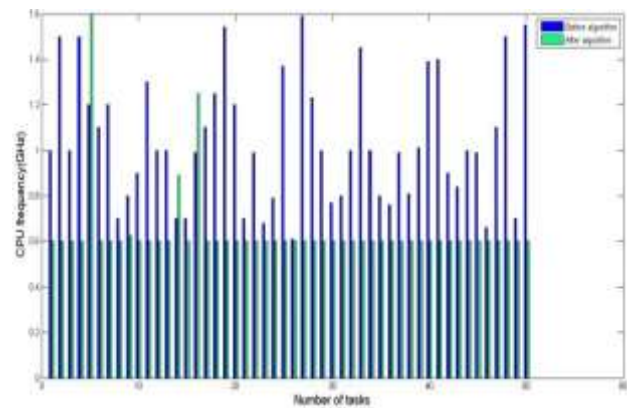
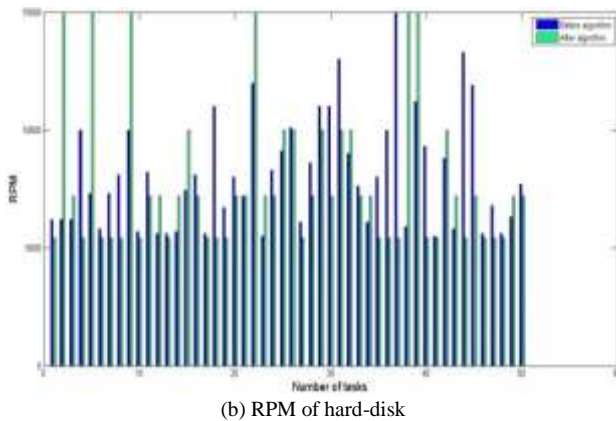


Fig.9. Number of missed tasks

Fig.9 shows the number of tasks that are missed prior and subsequent to the algorithm execution in which the number of the missing tasks is reduced by 80%. The amount of RPMs of the hard disk and frequencies of the CPU are shown in Fig.10. As shown in Fig.10. (a) and Fig.10. (b) the frequency of the CPU and the RPM of the hard disk in most of the tasks reduced to a minimum, respectively. Their respective average values before and after applying our algorithm are also shown in Fig.11. In this figure a reduction in the values of the CPU frequency and the RPM of the hard disk can be observed after running the algorithm.

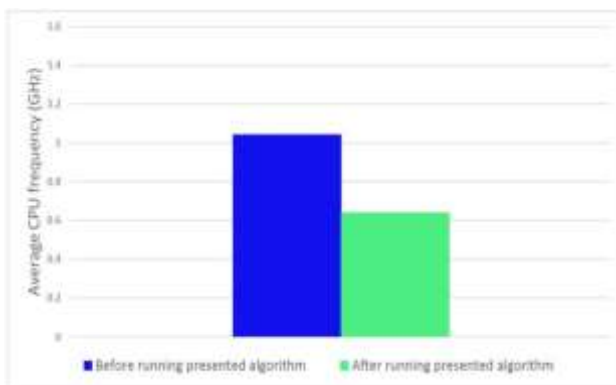


(a) CPU frequency

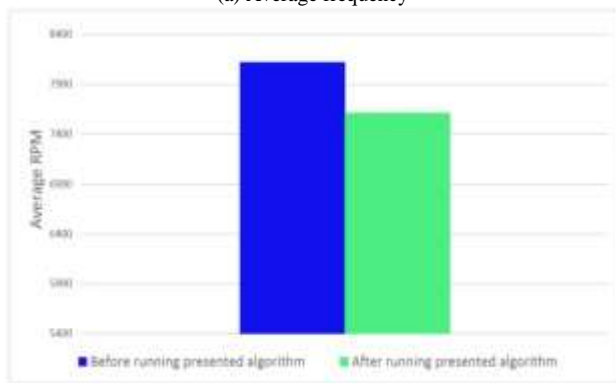


(b) RPM of hard-disk

Fig.10. (a) The values of CPU frequencies and (b) RPM of hard-disk, before and after applying algorithm



(a) Average frequency



(b) Average RPM

Fig.11. (a) Average CPU frequency and (b) Average RPM of hard disk drive before and after applying algorithm

V. CONCLUSION AND FUTURE WORK

In this paper we have presented a novel scheduling algorithm in order to reduce energy consumption in a real time periodic task with CPU and hard disk requests. To achieve this goal, the execution time of the CPU and/or the R/W time of the hard disk must be changed according to the remaining time and consumed energy. In the future study we intend to propose a new algorithm with the aim of minimizing energy consumption with a minimal performance reduction in multitask-multicore processes.

REFERENCES

- [1] C.M. Kanga, "CPU Frequency Emulation Based on DVFS," *In Utility and Cloud Computing (UCC)*, 2012, pp. 367– 374.
- [2] S.J. Cho, S.H. Yun, J.W. Jean, "A power saving DVFS algorithm based on Operational Intensity for embedded systems," *In IEICE Electronics Express*, vol. 12, no. 3, Jan. 2015, pp. 1-7.
- [3] Ch. Da-Ren, Ch. Young-Long, Ch. You-Shyang, "Time and Energy Efficient DVS Scheduling for Real-Time Pinwheel Tasks", in *Journal of Applied Research and Technology*, vol. 12, Issue. 6, Dec. 2014, pp. 1025- 1039.
- [4] Z. Tang, L. Qi, Z. Cheng, K. Li, S.U.Khan, K. Li, "An Energy-Efficient Task Scheduling Algorithm in DVFS-enabled Cloud Environment," in *Journal of Grid Computing*, April. 2015.
- [5] N. Babaii Rizvandi, J. Taheri, A.Y. Zomaya, "Some observations on optimal frequency selection in DVFS-based energy consumption minimization," *In Journal of Parallel and Distributed Computing*, vol. 71, Issue 8, 2011, pp. 1154-1164.
- [6] G.V. Laszewski, L. Wang, A. J.Young, X. He, "Power-Aware Scheduling of Virtual Machines in DVFS-enabled Clusters," *In Cluster Computing and Workshops*, 2009, pp. 1 – 10.
- [7] G.M. Tchamgoue, J. Seo, K.H. Kim, Y.K. Jun, "Compositional Power-Aware Real-Time Scheduling with Discrete Frequency Levels," *In Journal of Systems Architecture*, 2015.
- [8] J. Wu, "Energy-Efficient Scheduling of Real-Time Tasks with Shared Resources," *In Future Generation Computer Systems*, May. 2015.
- [9] X. Zhu, C. He, K.Li, X. Qin, "Adaptive energy-efficient scheduling for real-time tasks on DVS-enabled heterogeneous clusters," *In Journal of Parallel and Distributed Computing*, 2012, pp. 751-763.
- [10] Y. Zhang, R. Guo, "Power-aware fixed priority scheduling for sporadic tasks in hard real-time systems," *In Journal of Systems and Software*, vol. 90, 2014, pp. 128-137.
- [11] F. Kong, Y. Wang, Q. Deng, W. Yi, "Minimizing Multi-Resource Energy for Real-Time Systems with Discrete Operation Modes," *In Real-Time Systems (ECRTS), 2010 22nd Euromicro Conference on*, 2010, pp. 113 –122.
- [12] T. Inoue, A. Aikebaier, T. Enokido, M. Takizawa, "A Power Consumption Model of a Storage Server," *In Network-Based Information Systems (NBIS)*, 2011, pp. 382 – 387.
- [13] A. Hylick, R. Sohan, A. Rice, B. Jones, "An Analysis of Hard Drive Energy Consumption," *In Modeling, Analysis and Simulation of Computers and Telecommunication Systems*, 2008, pp. 1 – 10.
- [14] X. Mountroudou, A. Riska, E. Smirni, "Saving power without compromising disk drive reliability," *In Green Computing Conference and Workshops (IGCC)*, 2011, pp. 1– 6.
- [15] V. Swaminathan, K. Chakrabarty, S.S. Iyengar, "Dynamic I/O Power Management for Hard Real-time Systems," *In Hardware/Software Codesign*, 2001, pp. 237 – 242.
- [16] M. Pedram, K. Choi, "Dynamic Voltage and Frequency Scaling for Energy-Efficient System Design," *In the Association for Computing Machinery*, 2005.
- [17] E. Grochowski, R.D. Halem, "Technological impact of magnetic hard disk drives on storage systems," *In IBM SYSTEMSJOURNAL*, vol. 42, no. 2, 2003.

Author's Profiles



Vahdaneh Kiani received her B.S. in Computer Engineering from South Tehran Branch, Islamic Azad University, Tehran, Iran in 2011 and the M.S. in Computer Engineering from Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran in 2015.

Her research interests are in the areas of embedded and real time systems, scheduling algorithms and Network on Chip.



Zeynab Mohseni received her Associates Degree in Software Computer from Sabzevar Technical and Vocational Collage, Khorasan, Iran in 2007, the B.S. in Computer Engineering from Khorasan Institute of Higher Education, Mashhad, Khorasan, Iran in 2009 and the M.S. in Computer Engineering from

Science and Research Branch, Islamic Azad University, Tehran, Iran in 2015. Her research interests are in the areas of embedded and real time systems, scheduling algorithms and Network on Chip.



Amir Masoud Rahmani received his B.S. in Computer Engineering from Amir Kabir University, Tehran, in 1996, the M.S. in Computer Engineering from Sharif University of Technology, Tehran, in 1998 and the Ph.D. degree in Computer Engineering from IAU University, Tehran, in

2005. He is an assistant professor in the Department of Computer and Mechatronics Engineering at the IAU University. He is the author/co-author of more than 80 publications in technical journals and conferences. He served on the program committees of several national and international conferences. His research interests are in the areas of distributed systems, ad hoc and sensor wireless networks, scheduling algorithms and evolutionary computing.

How to cite this paper: Vahdaneh Kiani, Zeynab Mohseni, Amir Masoud Rahmani, "Real Time Scheduling for CPU and Hard Disk Requirements-Based Periodic Task with the Aim of Minimizing Energy Consumption", *IJITCS*, vol.7, no.10, pp.54-60, 2015. DOI: 10.5815/ijitcs.2015.10.07