

Energy performance evaluation for dynamic frequency scaling on rate monotonic and earliest deadline first scheduling algorithm



Sharizal Fadlie Sabri, Noor Azurati Ahmad *, Shamsul Sahibuddin, Salwani Mohd Daud, Kamilia Kamardin

Advanced Informatics School, University Technology of Malaysia, Kuala Lumpur, Malaysia

ARTICLE INFO

Article history:

Received 31 August 2017

Received in revised form

7 November 2017

Accepted 1 December 2017

Keywords:

Rate monotonic

Earliest deadline first

Dynamic voltage and frequency scaling

Simulator

SimSo

ABSTRACT

Dynamic Voltage and Frequency Scaling (DVFS) algorithm is widely used to reduce processor's energy consumption in mixed-criticality systems. Existing works show that there are more DVFS algorithms were developed based on Earliest Deadline First (EDF) scheduling scheme compared to Rate Monotonic (RM) scheduling scheme. To understand the reason why EDF is prevalent for DVFS algorithms, simulation can be done to get the power consumption of both scheduling schemes. However, there is no simulation tools available dedicated for this purpose. This research aims to investigate power consumption of both scheduling schemes under different processor speed levels using SimSo simulator. The processor's utilization and schedulability are determined using a set of task in SimSo. Power consumption for Least Common Multiple of tasks period is then calculated based on the proposed power model. The result shows that EDF performs better than RM and able to reduce 30% of energy at processor speed of 0.7. In addition, the capability of SimSo to be used in DVFS algorithm creation also had been observed. It is found that SimSo can be used for further research with additional codes.

© 2017 The Authors. Published by IASE. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

1. Introduction

In the last decade, battery-powered embedded systems have become more important not only in the industries but also in our daily life. The advancement of microprocessor technology has improved processing performance and dramatically improves the size and capabilities of an embedded system. This has increased the energy requirement of such systems. In the applications such as wireless sensors network, wearable devices and mobile robots, it may be impossible to replace the battery to prolong the system's lifetime. The growth of battery density is also much slower than other technologies as shown in Fig. 1. Thus, the system's hardware and software need to be designed for energy efficiency.

There are two existing algorithms commonly used to reduce the energy consumption of embedded systems: Dynamic Voltage and Frequency Scaling (DVFS) and Dynamic Power Management (DPM). DVFS (Florence et al., 2016; Xu et al., 2012; Moreno and De Niz, 2012; Chen et al., 2013) reduces the processor energy usage by reducing processor

frequency. This will slow down the processing speed and stretch the execution time as much as possible. Meanwhile, DPM (Deng et al., 2016; Awan and Petters, 2011; Huang et al., 2009) aims to reduce energy dissipation of the processor during idle time by shutting down the processors. This research is focusing on DVFS to understand how it can be applied efficiently.

From the literatures, it can be observed that most of DVFS algorithms are created based on EDF scheduler. This research tries to investigate the behavior of RM and EDF scheduling policies when the processor's speed is scaled to few different levels.

It is important to understand which scheduling policy is performed better for DVFS implementation in term of energy efficiency. The investigation is done using an open-source simulator, SimSo (Chéramy et al., 2014) to observe the schedulability of the task set. Then, energy consumption is determined based on the result. It is also important to see the feasibility of SimSo for DVFS research. This article is organized as follows: Section 2 discusses on the existing works related on the DVFS scheduling. The simulator, power model and task model are described in Section 3. Subsequent section will discuss on the result of the experiment. Finally, Section 5 concludes this article.

* Corresponding Author.

Email Address: azurati@utm.my (N. A. Ahmad)

<https://doi.org/10.21833/ijaas.2018.01.019>

2313-626X/© 2017 The Authors. Published by IASE.

This is an open access article under the CC BY-NC-ND license

(<http://creativecommons.org/licenses/by-nc-nd/4.0/>)

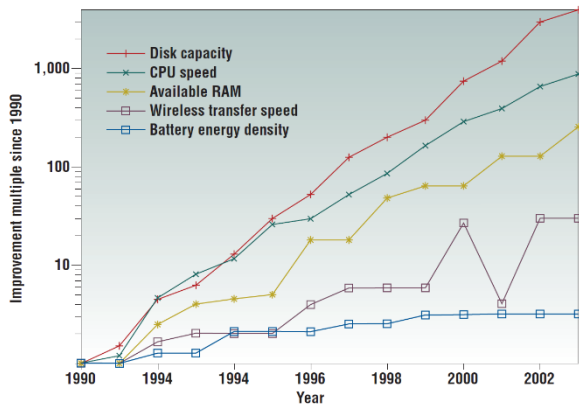


Fig. 1: Growth of battery energy density compared to other technologies (Paradiso and Starner, 2005)

2. Related works

DVFS can be applied to static and dynamic priority scheduler. Rate Monotonic (RM) and Earliest Deadline First (EDF) scheduler (Liu and Layland, 1973) has been widely used for DVFS implementation. RM scheduling is a static priority scheduling scheme that gives highest priority to the task with the shortest period. Contrary, EDF is a dynamic priority scheduling that gives highest priority to the task with imminent deadline. EDF will keep sorting the tasks by its deadline and assigns the priority.

The first paper on DVFS, published in 1995, introduced three algorithms based on EDF scheduler (Yao et al., 1995). Those three algorithms were designed for aperiodic task. One of the algorithms that is known as Average Rate Heuristic, set the speed of processor equal to the sum of the worst-case utilization of the active jobs. It may affect the processing speed when there are few active tasks causing them finished after deadlines if additional task arrives. Pillai and Shin (2001) introduced five algorithms considering both EDF and RM schedulers. All the algorithms only consider periodic task and set the speed at the lowest speed that guaranteed execution time of the task.

Aydin is a researcher who presented five algorithms focusing on periodic tasks. Interestingly, all the algorithms are based on EDF scheduler. Aydin et al. (2001) proposed an $O(n^2 \log n)$ algorithm to compute optimal speed values for task execution. It proves that every task can execute at constant speed without affecting the energy optimality. Aydin et al. (2004) proposed another three algorithms which are offline algorithm, online algorithm that dynamically reclaim energy and an algorithm that is using online, adaptive and speculative speed adjustment mechanism. The researcher continues his work (Aydin et al., 2006) by proposing an algorithm with more general power model that includes leakage power, frequency-dependent power and frequency-independent power components. In recent years, Bambagini et al. (2011) introduced enhanced algorithms, BSDVFS and BSDVFS* which are based on EDF scheduler. The algorithms consider

switching speed overhead by checking the period of dynamic slack.

3. Methodology

This section explains the power model used in the experiment to determine processor's power consumption. Then, the workload model is presented. At the end of the section, the SimSo simulator and the experiment are described.

3.1. Power model

This experiment considers a periodic real-time task set running on a single processor system. The processor is assumed to have DVFS mechanisms allowing it to dynamically adjust its operating frequency. It has several speed levels that represents the total number of frequencies supported by the processor. Consider a real-time task with execution time, t running on a processor at maximum frequency, f_{max} . The processor can be scaled to run at a several frequency levels, f where $0 < f \leq f_{max}$. So, the relative processor speed level s can be defined as f/f_{max} .

Processor's power consumption is composed of static and dynamic power. Static power consumption is the power dissipated during by processor during no activities. While, dynamic power consumption is the power consumed during task executions. This experiment only considers the dynamic power consumption which can be managed by the DVFS mechanism. Most of early DVFS algorithms assumed dynamic power is generally in proportion to the cubic of the speed, s . Therefore, energy consumption, E for a given execution time, t is defined by

$$E = \alpha \cdot \frac{t}{s} \cdot s^3 \quad (1)$$

$$E = \alpha \cdot t \cdot s^2 \quad (2)$$

where α is a coefficient (Ali et al., 2015). For this experiment, α is assumed equals to 1 for the sake of simplicity.

3.2. Task model

The task model for the experiment consists of independent tasks which are characterized by a 3-tuple of parameters: a period P , a deadline D , and a Worst-Case Execution Time (WCET) C . Tasks are released simultaneously after processor start execution. The task set used in the experiment (Bambagini et al., 2016) is shown in Table 1.

The utilization of the tasks, U_i can be calculated using the following formula (Buttazzo, 2005):

$$U_i = C/P \quad (3)$$

and processor's utilization, U_P is defined as

$$U_P = \sum_{i=1}^n U_i \quad (4)$$

where U_P is equal to 0.7.

3.3. Simulation

The experiment was done using a simulator called SimSo (Simulation of Multiprocessor Scheduling with Overheads) which is developed by the researchers in Laboratory for Analysis and Architecture of System, (LAAR) in France. SimSo is an open-source simulator developed using Python programming language. It means that researchers are able to extend its functionality based on their preferences.

The task set in Table 1 is inserted into the SimSo. All tasks are set to be aborted if they missed their deadlines. Fig. 2 shows the tasks that have been keyed in. Simulation is then set to run in five processor speed level: 0.6, 0.7, 0.8, 0.9 and 1.0. Speed level below 0.5 is not tested because it is not energy efficient (Bambagini et al., 2016) and increases the risk of missing deadline.

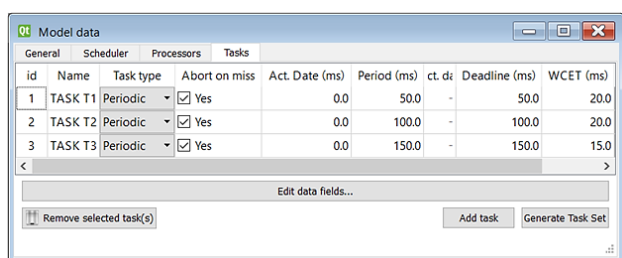


Fig. 2: Task set in SimSo

Simulation is done using RM and EDF scheduling for each speed level. Duration of simulation is the Least Common Multiple of periods that is 300ms. Processor’s utilization is determined and energy consumption for every case is calculated. Finally, the functionality and capability of SimSo for DVFS algorithm have been analyzed.

Table 1: Task set

Task	Period, P	Deadline, D	WCET, C
T1	50	50	20
T2	100	100	20
T3	150	150	15

4. Results and analysis

The results are divided into three components. First, result on simulation of RM and EDF at maximum processor speed. Second, result on simulation at different speed level. Finally, a comparison with the result of DVFS algorithm is shown.

4.1. RM and EDF at maximum speed

Simulation shows that both scheduling scheme successfully shows the processor’s utilization similar as calculated. Fig. 3 shows the simulation result of RM and EDF which is exactly similar.



Fig. 3: Simulation result of RM and EDF at maximum speed

4.2. RM and EDF at different speed level

Simulation shows that both scheduling scheme perform similarly at speed 0.8 and 0.9. However, RM fails to schedule at speed lower than that. EDF still able to schedule at speed 0.7 at maximum processor’s utilization and energy consumption is reduced by 30%. Table 2 summarizes the result of simulation. Fig. 4 and Fig. 5 show the results for both scheduling schemes at speed 0.7 respectively.

result. Researcher only knows whether the task set is schedulable or not. To have this information, SimSo needs to be amended using the Python codes.

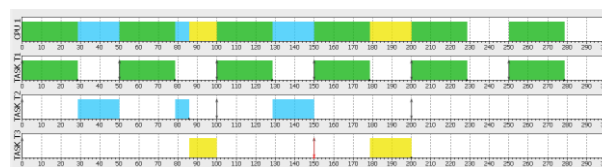


Fig. 4: RM fails to schedule Task 3 at speed 0.7

4.3. DVFS algorithm using SimSo

SimSo provides two built-in DVFS algorithms: Static EDF and Cycle Conserving EDF (CC-EDF). Simulation for both algorithms depicts the similar result as shown as Fig. 5. Task allocation of the algorithms is shown in Fig. 6. However, there is no information on processor speed in the simulation

5. Conclusion and future work

This paper investigates the processor’s utilization of RM and EDF algorithms using SimSo simulator. Based on the task set, it is observed that EDF performs better than RM at a lower speed. At the minimum schedulable speed, energy consumption of EDF is reduced by 30% compared to RM which is

only reduced 20%. Result also shows the advantage of EDF that allows full processor utilization, which implies a more efficient exploitation of computational resources. SimSo also able to schedule based on DVFS algorithm but amendment

needs to be done to the codes so that more information can be displayed to users. As a conclusion, SimSo can be used as simulator for a research on designing a DVFS algorithm with additional coding.

Table 2: Simulation result for different speed level

Processor Speed	Rate Monotonic Scheduling			Earliest Deadline First		
	Energy Consumption	Utilization	Schedulability	Energy Consumption	Utilization	Schedulability
0.6	107.9998	1.000	No	107.9998	1.000	No
0.7	125.9999	0.8571	No	146.9998	1.000	Yes
0.8	168	0.8750	Yes	168	0.8750	Yes
0.9	188.9998	0.7778	Yes	188.9998	0.7778	Yes
1.0	210	0.700	Yes	210	0.7	Yes

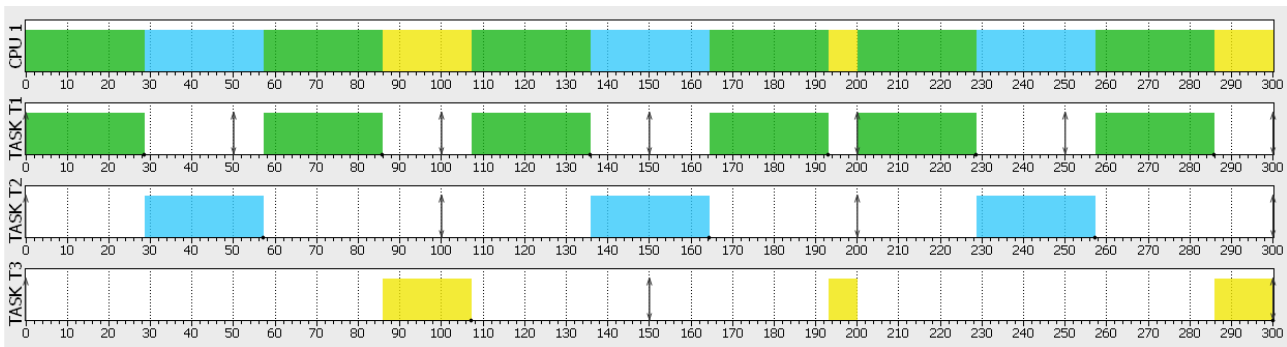


Fig. 5: EDF utilize 100% of processor at speed 0.7

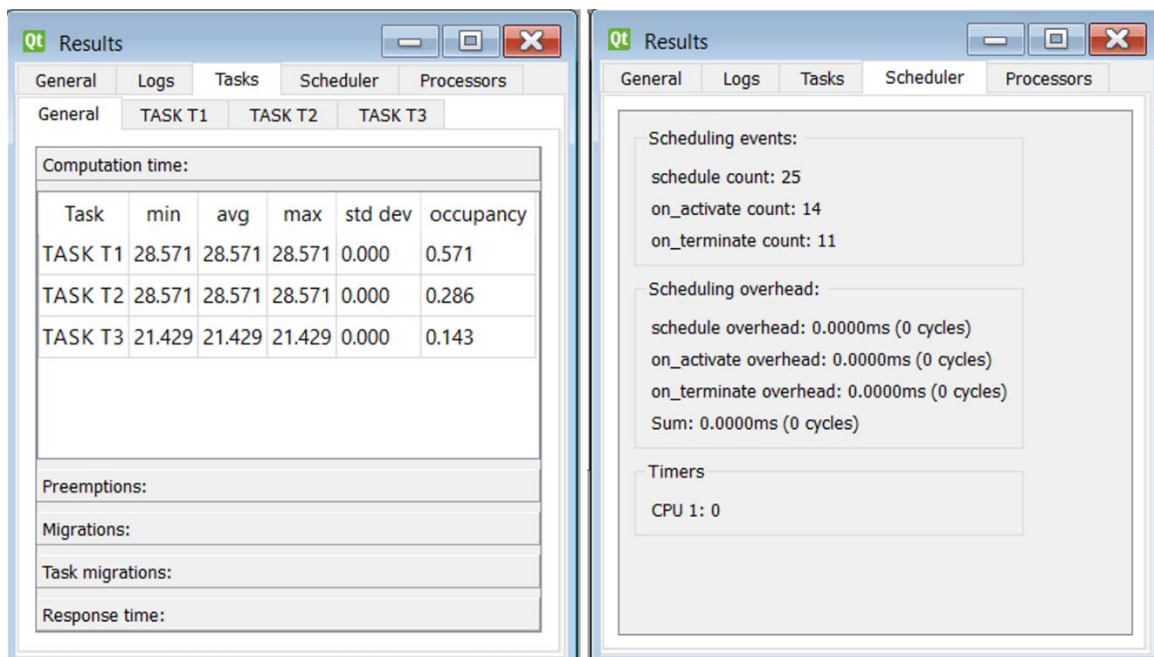


Fig. 6: Task allocation of Static-EDF and CC-EDF

In the future, the author will work on an EDF-based DVFS algorithm to reduce energy consumption of mixed-criticality systems. SimSo may provide a good foundation for author as a task simulator. It can help author to save time from creating new simulator from scratch. However, the real hardware implementation is important to prove the proposed algorithms.

Acknowledgement

The work presented in this paper was supported by the UTM GUP Tier 1 grant, Q.K130000.2538.17H44.

References

Ali I, Seo JH, and Kim KH (2015). A dynamic power-aware scheduling of mixed-criticality real-time systems. In the IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, IEEE, Liverpool, UK: 438-445. <https://doi.org/10.1109/CIT/IUCC/DASC/PICOM.2015.63>

Awan MA and Petters SM (2011). Enhanced race-to-halt: A leakage-aware energy management approach for dynamic priority systems. In the 23rd Euromicro Conference on Real-Time Systems, IEEE, Porto, Portugal: 92-101. <https://doi.org/10.1109/ECRTS.2011.17>

Aydin H, Devadas V, and Zhu D (2006). System-level energy management for periodic real-time tasks. In the 27th IEEE

- International Real-Time Systems Symposium, IEEE, Rio de Janeiro, Brazil: 313-322. <https://doi.org/10.1109/RTSS.2006.48>
- Aydin H, Melhem R, Mossé D, and Mejía-Alvarez P (2001). Determining optimal processor speeds for periodic real-time tasks with different power characteristics. In the 13th Euromicro Conference on Real-Time Systems, IEEE, Delft, Netherlands: 225-232. <https://doi.org/10.1109/EMRTS.2001.934038>
- Aydin H, Melhem R, Mossé D, and Mejía-Alvarez P (2004). Power-aware scheduling for periodic real-time tasks. IEEE Transactions on Computers, 53(5): 584-600.
- Bambagini M, Marinoni M, Aydin H, and Buttazzo G (2016). Energy-aware scheduling for real-time systems: a survey. ACM Transactions on Embedded Computing Systems (TECS), 15(1): 7.1-7.34.
- Bambagini M, Prosperi F, Marinoni M, and Buttazzo G (2011). Energy management for tiny real-time kernels. In the International Conference on Energy Aware Computing, IEEE, Istanbul, Turkey: 1-6. <https://doi.org/10.1109/ICEAC.2011.6136687>
- Buttazzo GC (2005). Rate monotonic vs. EDF: Judgment day. Real-Time Systems, 29(1): 5-26.
- Chen G, Huang K, and Knoll A (2013). Energy optimization for real-time multiprocessor system-on-chip with optimal DVFS and DPM combination. In the IEEE 11th Symposium on Embedded Systems for Real-time Multimedia, IEEE, Montreal, QC, Canada: 40-40. <https://doi.org/10.1109/ESTIMedia.2013.6704501>
- Chéramy M, Hladik PE, and Déplanche AM (2014). SimSo: A simulation tool to evaluate real-time multiprocessor scheduling algorithms. In the 5th International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems, Madrid, Spain.
- Deng C, Guo R, Wang H, and Peng A (2016). A dynamic power management algorithm for sporadic tasks in real-time embedded systems. In the IEEE Trustcom/BigDataSE/I SPA, IEEE, Tianjin, China: 2073-2078. <https://doi.org/10.1109/TrustCom.2016.0318>
- Florence AP, Shanthi V, and Simon CB (2016). Energy conservation using dynamic voltage frequency scaling for computational cloud. The Scientific World Journal, 2016: Article ID 9328070, 13 pages. <https://doi.org/10.1155/2016/9328070>
- Huang K, Santinelli L, Chen JJ, Thiele L, and Buttazzo GC (2009). Periodic power management schemes for real-time event streams. In the 48th IEEE Conference on Decision and Control, 2009 Held Jointly with the 28th Chinese Control Conference, IEEE, Shanghai, China: 6224-6231. <https://doi.org/10.1109/CDC.2009.5400034>
- Liu CL and Layland JW (1973). Scheduling algorithms for multiprogramming in a hard-real-time environment. Journal of the ACM (JACM), 20(1): 46-61.
- Moreno GA and De Niz D (2012). An optimal real-time voltage and frequency scaling for uniform multiprocessors. In the IEEE 18th International Conference on Embedded and Real-Time Computing Systems and Applications, IEEE, Seoul, South Korea: 21-30. <https://doi.org/10.1109/RTCSA.2012.51>
- Paradiso JA and Starner T (2005). Energy scavenging for mobile and wireless electronics. IEEE Pervasive Computing, 4(1): 18-27.
- Pillai P and Shin KG (2001). Real-time dynamic voltage scaling for low-power embedded operating systems. In the 18th ACM symposium on Operating Systems Principles, ACM, Banff, Alberta, Canada, 35(5): 89-102. <https://doi.org/10.1145/502034.502044>
- Xu H, Kong F, and Deng Q (2012). Energy minimizing for parallel real-time tasks based on level-packing. In the IEEE 18th International Conference on Embedded and Real-Time Computing Systems and Applications, IEEE, Seoul, South Korea: 98-103. <https://doi.org/10.1109/RTCSA.2012.10>
- Yao F, Demers A, and Shenker S (1995). A scheduling model for reduced CPU energy. In the 36th Annual Symposium on Foundations of Computer Science, IEEE, Milwaukee, WI, USA: 374-382. <https://doi.org/10.1109/SFCS.1995.492493>