Multiprocessor Real-Time Scheduling for Wireless Sensors Powered by Renewable Energy Sources

Conference Paper · October 2018

DOI: 10.1109/AICCSA.2018.8612791

3 authors:

Hussein El Ghor
Lebanese University

Maryline Chetto
University of Nantes

Rola el Osta
University of Nantes

Multiprocessor Real-Time Scheduling for Wireless Sensors Powered by Renewable Energy Sources

Hussein El Ghor
LENS Laboratory
Lebanese University
B.P. 813 Saida, Lebanon
E-mail: hussain@ul.edu.lb

Maryline Chetto
IRCCyN Laboratory
University of Nantes
1 Rue de la Noe, F-44321 Nantes, France
E-mail: maryline.chetto@univ-nantes.fr

Rola El Osta
LENS Laboratory
Lebanese University
B.P. 813 Saida, Lebanon

***Abstract*—Ambient energy harvesting has become a popular solution for battery-operated systems with finite energy supply such as wireless sensor networks. This paper investigates the problem of multiprocessor real-time scheduling in a system whose energy reservoir is replenished by an ambient energy source. In particular, we focus on energy-efficient partitioning for periodic real-time tasks in a homogeneous multi-core platform where both timing and energy requirements are considered. We assume that the optimal scheduler, namely the Earliest Deadline - Harvesting (ED-H) [1], is used on every core of the architecture. Our objective is to find a feasible partitioning solution based on the real-time execution of tasks allocated to the sensor nodes based on the actual energy harvesting data to guarantee the desirable absence of both energy starvation situations and deadline violation. For this sake, we propose an Energy Harvesting-Reasonable Allocation (EH-RA) algorithm that amounts to the traditional bin-packing technique by guaranteeing both timing constraints and energy awareness perspectives. Experimental results show that our approach can achieve a significant gain in performance when compared to EDF.**

I. Introduction

Energy management has become a major problem in wireless sensor networks [2]. Sensor nodes are often equipped with small batteries [3]. Furthermore, in many applications such as submarine, nuclear, medical, the location of these devices render the activity of replacing batteries infeasible or very costly. As a consequence, ambient energy is required to replenish the batteries so as to prolong the lifetime of networks is a widespread concern. There exist some previous work in literature on real-time energy harvesting in embedded systems such as wireless sensors.

The technical challenges to sustain the perpetual operation of the energy autonomy of the system while still meeting all time constraints were initially described in [4], [5]. Among these new challenges of researchers, the energy Harvesting system consists of a monoprocessor unit with unique voltage and frequency.

With today's computational demands and as the miniaturization of integrated circuits reaches its physical limits, multi-core platforms have emerged as a popular solution for different types of applications like mobile computing systems. This is due to the fact that multi-core platforms overcome the limitations of single-core platforms in terms of processing capacity. However, certain challenges should be considered. The first one concerns the scheduling problem under real-time requirements. Uniprocessor scheduling algorithms cannot be applied to multiprocessor platform without loss of performance. Optimal scheduling is a critical design issue as it must be guaranteed that a schedule exists that meets all deadlines of tasks. Energy-efficiency is another central challenge for real-time systems. How to ensure energy availability so as to execute the tasks timely on the platform? This question can reveal very difficult to answer when the cores draw their energy from different ambient energy sources [6].

In our research, we consider multiprocessor real-time systems where both time and energy limitations are to be studied. The objective is to efficiently exploit the effectiveness offered by real-time tasks scheduling methods to face both timeliness requirements and energy starvation situations by using the ED-H scheduler on every single. For that purpose, we propose new partitioned scheduling algorithms based on traditional bin-packing technique to allocate tasks with energy limitation to processors, so as to maximize the percentage of feasible task sets and reduce the number of used processors. To the best of our knowledge, this paper is the first one that studied energy harvesting into partitioning heuristics.

The rest of this paper is organized as follows: The related works are presented in the subsequent section. Section 3 gives the model and terminology. The ED-H algorithm is presented in section 4. In section 5, we propose the Energy Harvesting-Reasonable Allocation (EH-RA) algorithm. Finally, section 6 concludes the paper.

II. Related Works

Multiprocessor real-time scheduling has been an active research topic during the past years [7]. One important approach for the allocation of tasks in multiprocessor systems is to transform it into a classical bin-packing problem [8]. However, this approach ignores energy limitations that may significantly reduce the feasibility of the tasks allocated to each processor. The energy issue has been considered in some works with the objective to reduce the energy consumption by Dynamic Voltage and Frequency Scaling (DVFS) or Dynamic Power Management (DPM) techniques [9] . In such works, the rechargeability of the energy storage unit is always disregarded.

978-1-5386-9120-5/18/$31.00 ©2018 IEEE

A. Multiprocessor Real-Time Scheduling

We first present the related works for classical real-time multiprocessor scheduling with timing as the only constraint. Allocation of tasks to is originally derived based on the bin-packing technique [10], with the task utilization being the *"size"* of the object and the processor utilization bound being the *"capacity"* of the bin.

Considering the intractable nature of the problem, several heuristics were subject of many research papers where common approaches based on the traditional bin-packing methods were proposed, including Worst-Fit (WF), Best-Fit (BF), Next-Fit(NF), and First-Fit (FF) [11].

- **First-Fit (FF):** The FF allocates a new task to the processor according to increasing indices and assigns each task to the lowest index, such that the utilization of the new task along with the utilization of the tasks already allocated to that processor, do not exceed the capacity of the processor. When the current task cannot fit into the capacity of the processor, then FF allocates the task to the next processor.
- **Next-Fit (NF):** The NF algorithm works as follows: Initially all processors are idle, and we start with first processor and first task. If first processor has processor utilization for first task, allocate this task to the current processor, and consider the next task. NF allocates the next task to the current processor, only if it fits into this processor. Otherwise, consider the next processor and same task. Repeat until all tasks are assigned. Note that NF does not check if the task can be allocated in previous processors.
- **Best-Fit (BF):** BF allocates the task to the feasible processor with the lowest residual capacity, in which the task can be allocated. In case there is more than one processor with the same capacity, then BF will choose the processor with the smallest index.
- **Worst-Fit (WF):** Put each task into the lowest capacity processor among those which are already busy. Only start a new processor if the task doesn't fit into processor that's already been used. If there are two or more processor already used, allocate tasks into the processor with the greatest capacity available, in which they can be feasibly allocated.

B. Energy Efficient Real-time Scheduling

Energy management in real-time systems is a multi-faceted optimization problem. A significant number of scheduling techniques have been recently proposed by research community for homogeneous multi-core systems where the static power consumption is considered as negligible [12], [13], [14].

Energy-efficient issues in energy harvesting real-time embedded systems have been studied in the past years, since Aydin et al. in [15] studied the scheduling of the four traditional bin-packing heuristics for homogeneous multi-core systems in which periodic independent tasks are considered. For the aim of reducing the energy consumption, authors proposed a feasible computation method to partition tasks by using variable voltage EDF scheduling. They also demonstrated that when balancing the workload among the processors, like in the Worst-Fit Decreasing (WFD) techniques, they produce the most effective scheduling method. Petrucci et al [16] focused on the problem of allocating a set of independent tasks on a heterogeneous real-time system. An optimization approach is then proposed to find the appropriate energy efficient allocation of tasks having a soft real-time performance and memory bandwidth constraints in a multiprocessor system. To the best of our knowledge, this work is one of the first attempts to incorporate renewable energy to partitioned real-time multiprocessor systems.

III. Models and Terminology

A. Task Model

We consider a set of n independent periodic real-time tasks $\Gamma = \{\tau_1, \tau_2, \cdots, \tau_n\}$. A four-tuple (C_i, D_i, T_i, E_i) is associated with each τ_i. In each request of τ_i, we require a worst case execution time of C_i time units and has a worst case energy consumption (WCEC) of E_i. We consider that the WCEC is independent from its WCET. P_i is referred to the period and D_i the relative deadline. We follow the rule that $0 < C_i \leq D_i \leq T_i$ for each $1 \leq i \leq n$. The processor utilization of task τ_i, denoted by u_{p_i}. We denote $U_p = \sum_{i=1}^{n} u_{p_i}$ to be the total processor utilization of the task set Γ is denoted as . We introduce also the energy utilization of task τ_i as $u_{ei} = \frac{E_i}{T_i}$. We denote U_e, measured in joules/s, as the average power consumed by Γ when executing on the device where $U_e = \sum_{i=1}^{n} u_{ei}$.

Tasks are allocated to m identical processors $\{P_1, P_2, \cdots, P_m\}$ and are independently executed from each other. Once a task is assigned to a processor, it cannot be executed on another processor. The energy source module at a time t harvests the energy from its ambient environment and is then converted into electrical energy at power $P_{rj}(t)$ where $P_{rj}(t)$ is the WCCR (Worst Case Charging Rate). $P_{rj}(t)$ is defined as the instantaneous charging rate that incorporates all losses caused by power conversion and charging process. The harvested energy is then stored in an energy storage unit whose capacity is B_j supplying the processor P_j.

B. Energy Model

Each processor P_j is fed by an ideal energy storage unit (supercapacitor or battery) that has a nominal capacity B_j that corresponds to the maximum stored energy and is expressed in Joules. At each time t, the energy available in the energy reservoir of processor P_j is denoted $E_j(t)$ that cannot exceed the energy storage unit capacity B_j. This means,

$$E_j(t) \leq B_j \ \ \forall t \tag{1}$$

All energy reservoirs are considered to be fully-replenished at time $t = 0$. This means,

$$E_j(0) = B_j \ \ \forall\, 1 \leq j \leq m \tag{2}$$

The harvested energy $Es_j(t_1,t_2)$ by $P_{rj}(t)$ at time interval $[t_1,t_2]$ is given as

$$Es_j(t_1,t_2) = \int_{t_1}^{t_2} P_{rj}(t)dt \tag{3}$$

IV. Earliest Deadline for Energy Harvesting systems (ED-H) Scheduler

In energy constrained systems, dynamic power management plays a crucial role due to its impact on the resulting performance. The dynamic power management rule permits to decide when to put the processor in the active mode and for how long time. The objective of such a policy is to prevent from energy depletion while still preserving the system from deadline violation. Consequently, we presented a novel energy-aware scheduling algorithm ED-H (Earliest Deadline under energy Harvesting) and we proved it to be optimal [1].

The ED-H scheduler executes tasks according to the EDF rule. However, before we authorize the execution of a ready task, the energy storage unit must be sufficient to provide energy for all future occurring tasks. When this condition is not verified, the processor has to be idle in order to enable the storage unit to replenish as much as possible and as long as all the deadlines can still respect despite execution postponement. To formally present the ED-H algorithm, we need to introduce two fundamental concepts: slack time and slack energy.

The slack time of a hard deadline task τ_i at current time t is

$$ST_{\tau_i}(t) = d_i - t - h(t,d_i) \tag{4}$$

Where $h(t,d_i)$ is the total processing demand of uncompleted tasks at t with deadline at or before d_i.

We may then define the slack time of a periodic task set Γ at current time t as follows:

$$ST_\Gamma(t) = min_{d_i>t} ST_{\tau_i}(t) \tag{5}$$

We define the slack energy of τ_i at current time t by equation

$$SE_{\tau_i}(t) = E(t) + E_s(t,d_i) - g(t,d_i) \tag{6}$$

Where $g(t,d_i)$ represents the total energy required by tasks on the time interval $[t,d_i)$. It concerns both tasks which are ready at t but not completed at d_i and future tasks, with deadline less than or equal to d_i. In addition, $E_s(t,d_i)$ is the amount of energy that is produced by the renewable energy source between t and d_i.

Hence, the slack energy of the periodic task set Γ at current time t represents the maximum energy surplus that the system could consume instantaneously at t. The slack energy at t is

$$SE_\Gamma(t) = max_{t<d_i} SE_{\tau_i}(t) \tag{7}$$

Let $Q_r(t)$ be the list of uncompleted tasks ready for execution at time t. $SE(t)$ and $ST(t)$ are respectively the slack energy and the slack time of the system at time t. The ED-H scheduling algorithm obeys the following rules:

- **Rule 1:** The EDF priority order is used to select the future running task in $Q_r(t)$.
- **Rule 2:** The processor is imperatively idle in $[t,t+1)$ if $Q_r(t)=\phi$.
- **Rule 3:** The processor is imperatively idle in $[t,t+1)$ if $Q_r(t)\neq\phi$ and either $E(t)=0$ or $SE(t)=0$.
- **Rule 4:** The processor is imperatively busy in $[t,t+1)$ if $Q_r(t)\neq\phi$ and either $E(t)=B$ or $ST(t)=0$.
- **Rule 5:** The processor can equally be idle or busy if $Q_r(t)\neq\phi$, $0<E(t)<B$, $ST(t)>0$ and $SE(t)>0$.

V. Energy Harvesting-Reasonable Allocation Algorithm

The allocation is carried out using Reasonable Allocation algorithms (RA). The uniprocessor utilization bound for ED-H scheduling of periodic and independent tasks is one ($U_p=1$). Consequently, a task of utilization factor u_{p_i} fits into processor P_j , which already has m_j tasks allocated to it with total processor utilization U_{pj} , if the (m_j+1) tasks are timely schedulable, that is if $U_{pj} \leq 1 - u_{p_i}$.

Moreover, a task τ_i is energy schedulable if its energy utilization u_{ei} fits into processor P_j , which already has m_j tasks allocated to it with total processor utilization U_{ej}. In other words, the (m_j+1) tasks are energy schedulable if $U_{ej} \leq P_{rj} - u_{ei}$.

Hence, in order to assign the task τ_i, we have to find the lowest j index such that a temporally allocation of task τ_i along with m_j tasks already allocated to processor P_j is still feasible, and the value of $P_{rj} - U_{ej}$ is the smallest possible.

A. Illustrative Example

We consider in this example a multi-core platform consisting of two processors ($m=2$), and a task set Γ of five tasks with same parameters as in table I. Each task is now characterized by its worst case energy consumption (table I). We assume that the level of the energy storage capacity for each processor is $B_1=B_2=16$. For ease of simplicity, we assume that the rechargeable power for each processor, P_{rj}, is constant along time and equals 3.

TABLE I
A task set with five real-time periodic tasks with energy consumption

Task τ_i	C_i	D_i	T_i	E_i	u_{ei}	u_{p_i}
1	2	6	8	8	1.0	0.25
2	2	8	10	10	1.0	0.20
3	6	16	20	20	1.0	0.30
4	8	18	20	20	1.0	0.40
5	8	32	40	28	0.7	0.20

When we schedule the task set Γ on two processors (figure 1) according to the bin-packing based approaches that allocate real-time tasks solely based on their energy utilization factors, we verify that the used heuristics can successfully schedule the

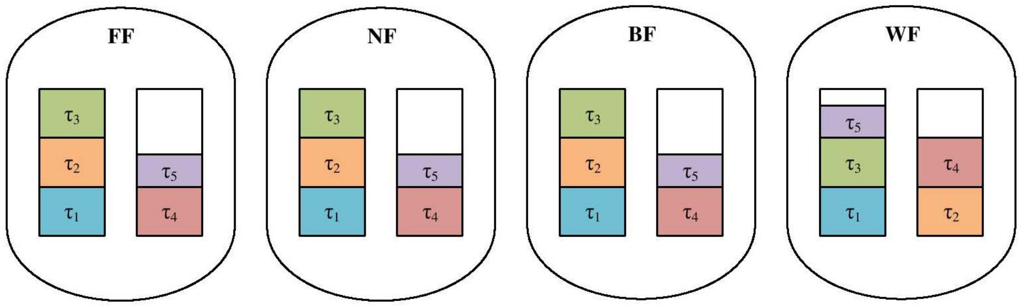

Fig. 1. Assignment of tasks under energy constraints according to FF, NF, BF and WF in EH.

Algorithm 1 Energy Harvesting-Reasonable Allocation Algorithm $(EH - RA)$

Input: Task set Γ of n tasks $\Gamma = \{\tau_1, \tau_2, \cdots, \tau_n\}$ and a set of m processors $\{P_1, P_2 \cdots, p_m\}$, Processor P_j has a battery with capacity B_j and source power $P_{rj}(t)$.
Output: $EH - RA$ Schedule.

$i := 1$; $j := 1$; /*i = ith task, $j = j^{th}$ processor */
while $i \le n$ **do**
  /* Schedule task τ_i according to Reasonable Allocation algorithm, here Worst-Fit is used */
  Sched ← FALSE
  for $1 \le j \le m$ **do**
    if $(U_{p_j} + u_{p_i}) \le 1$ **then**
      if $(U_{ej} + u_{ei}) \le P_r$ and $P_{rj} - U_{ej}$ is the smallest possible **then**
        $U_{ej} = U_{ej} + u_{ei}$;
        $U_{p_j} = U_{p_j} + u_{p_i}$;
        Sched ← TRUE
        break;
      end if
    end if
  end for
  $i := i + 1$;
end while

tasks in Table I. If we consider the same assignment of tasks as in the Worst-Fit case and according to the processor utilization (figure 1), tasks τ_1, τ_3 and τ_5 are assigned to processor P_1 and tasks τ_2 and τ_4 are assigned to processor P_2.

By scheduling the tasks in processor P_1 according to EDF, we obtain a non feasible schedule since the battery B_1 becomes empty at $t = 16$. As the system immediately stops, we conclude that the deadline miss ratio during the first hyperperiod is about 37.5%.

In contrast, when scheduling the same tasks according to ED-H, note that neutral operation is guaranteed and all tasks are executed at the end of the hyperperiod.

VI. Experiments and Results

In this section, we present a discussion of our experimental evaluations for the proposed partitioned scheduling algorithms. We first introduce the experimental setup used in our evaluation.

A. Experimental Setup

We conducted two sets of experiments to study the performance of our scheduling algorithm EH-RA against the non-idling scheduler such as EDF. The operation of the proposed study is measured by calculating for different schedulers the *success ratios*, which means the number of feasible tasks over the number of total tasks generated. We tested the above schedulers with respect to the processor utilization and the number of tasks. The task periods are randomly generated within $[10, 500]$. For each testing point, we generated 1000 task sets. A random number generator enables us to produce, for every quantum of time, a power energy profile with minimum value 1 and a maximum value, here 9, as an input of the simulator.

B. Experiment 1: Performance Evaluation by Varying the Processor Utilization

In this experiment, we evaluated the performance of the proposed schedulers with respect to the processor utilization. We varied the processor utilization from 0.5 to 1 with an increment of 0.05. The experimental results were collected and plotted in Figure 2.

We can observe that our proposed algorithms (EH-WF and EH-FF) outperforms the others (EDF-WF and EDF-FF). For example, when the processor utilization is 0.8, EH-WF can achieve a success ratio around 0.77, with an improvement of around 69% and 72% over EDF-WF and EDF-FF respectively. From figure 2, we can observe that EH-WF can guarantee the feasibility of any task set with processor utilization below 0.7. The success ratio by EDF-WF and EDF-FF drop sharply when the processor utilization is above 0.7. This is because that while EH-WF can guarantee any task sets with utilizations no more than 0.7, it rejects any task set that cannot pass the energy availabilty checking condition. We can also see that

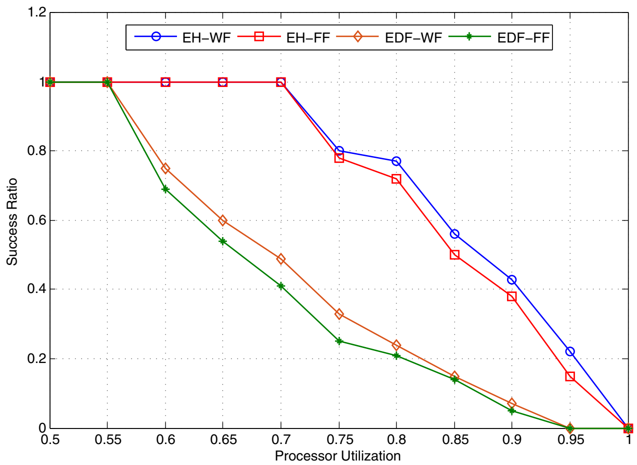

Fig. 2. Performance v.s. Processor Utilization.

when the processor utilization is 0.85, EH-WF can achieve a success ratio 4 times of that by EDF-WF.

Compared with EDF-BF, the improvement of our proposed algorithm (i.e. EH-WF) comes from the fact that, instead of executing ready tasks if only fits the timing constraints, EH-WF takes also energy constraint into consideration, and finds the optimal of allocation of tasks to processors such that tasks are timely scheduled on the processor and without energy starvation. Thus, our proposed EH-WF algorithm always outperforms the traditional EDF scheduler.

C. Experiment 2: Performance Evaluation by Varying the Nember of Tasks

In this experiment, we evaluated the performance of the schedulers with respect to the number of tasks. We varied the number of tasks from 4 to 20. The experimental results were collected and plotted in Figure 3.

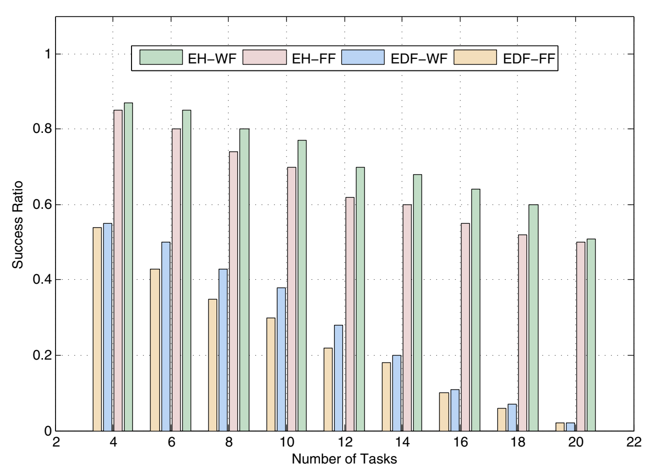

Fig. 3. Performance v.s. number of tasks.

EDF-WF offers good performance at small number of tasks, though its feasibility performance degrades rapidly when the number of tasks increases. From Figure 3, we can observe that EH-WF can achieve success ratios significantly better than the other three approaches. When the number of tasks is 10, the success ratio of EH-EDF is around 51% and 62% more that EDF-WF and EDF-FF, respectively. As the number of tasks increases the gain in success ratio increases till it reaches 96% when the number of tasks is 20. The reason is that tasks are allocated to processors according to the energy demand that must fit the ED-H scheduler on the selected processor.

VII. Conclusion

This work takes a real-time multiprocessor based on energy harvesting systems as a scheduling object, adds the energy consumption model for a homogeneous platform, pursues the goal of reducing energy consumption, and maximizing the number of feasible task set. From this characterization, we propose a flexible solution, Energy Harvesting-Reasonable Allocation (EH-RA) algorithm, based on the traditional bin-packing technique to assign tasks with energy constraints to processors so as to guarantee deadline violation and energy starvation. We use the optimal scheduler, namely ED-H, on every core of the architecture. Simulation experiments show that our proposed scheduler increases the percentage of feasible task sets by an average of 41% and 60% when respectively varing the processor utilization and the number of tasks as compared to EDF.

The future work will focus on a complete simulation analysis for the performance of EH-RA relative to nonidling scheduler such as EDF that is also based on the bin-packing technique.

References

[1] M. Chetto. *Optimal Scheduling for Real-Time Jobs in Energy Harvesting Computing Systems*. IEEE Transactions on Emerging Topics in Computing, 2014.

[2] Weizhe Zhang, Boyu Song, and Enci Bai. *A Trusted Real-Time Scheduling Model for Wireless Sensor Networks*. Journal of Sensors, 8 pages, 2016.

[3] V. Kianzad and S. S. Bhattacharyya. *Efficient techniques for clustering and scheduling onto embedded multiprocessors*. IEEE Transactions on Parallel and Distributed Systems, vol. 17, no. 7, pp. 667-680, 2006.

[4] A. Kansal, J. Hsu, S. Zahedi, M.B. Srivastava. *Power Management in Energy Harvesting Sensor Networks*. ACM Transactions on Embedded Computing Systems, vol. 6, No. 4, 2007.

[5] A. Kansal, J. Hsu,. *Harvesting aware Power Management for Sensor Networks*. Proc. of ACM/IEEE Design Automation Conference, pp. 651-656, 2006.

[6] Steck, J.B. and Rosing, T.S. *Adapting Task Utility in Externally Triggered Energy Harvesting Wireless Sensing Systems*. Sixth International Conference on Networked Sensing Systems (INSS), 2009.

[7] Ming Fan. Real-Time Scheduling of Embedded Applications on Multi-Core Platforms. PhD thesis, Florida International University, 2014.

[8] Ming Fan, Gang Quan. *Harmonic-Fit Partitioned Scheduling For Fixed-Priority Real-Time Tasks On the Multiprocessor Platform*. Ninth IEEE/IFIP International Conference on Embedded and Ubiquitous Computing, 2011.

[9] Gang Chen, Kai Huang, and Alois Knoll. *Energy optimization for real-time multiprocessor system-on-chip with optimal dvfs and dpm combination*. ACM Transactions on Embedded Computing Systems (TECS), 2014.

[10] K. Shin and P. Ramanathan. *Real-Time Computing: A New Discipline of Computer Science and Engineering*. Proc. IEEE, 82(1), pp. 6-24, Jan. 1994.

[11] E. G. Coffman, G. Galambos, S. Martello and D. Vigo, *Bin Packing Approximation Algorithms: Combinational Analysis*. Kluwer Academic Publishers, Ed. D. Z. Du and P.M. Pardalos, 1998.

[12] J.-J.Chen and C.-F.Kuo. *Energy-efficient scheduling for real-time systems on dynamic voltage scaling (dvs) platforms*, in RTCSA, 2007.

[13] Ying LiJianwei, Niu Jianwei, Niu Mohammed, Atiquzzaman Mohammed and Atiquzzaman Xiang Long. *Energy-aware scheduling on heterogeneous multi-core systems with guaranteed probability*. Journal of Parallel and Distributed Computing, 2016.

[14] Siddharth Garg. *Dynamic Power Management for Dark Silicon Multicore Processors*, Advances in Computers, Volume 110, Pages 171-216, , 2018.

[15] H. Aydin and Q. Yang. *Energy-aware partitioning for multiprocessor real-time systems*, in Proceedings of the 17th International Symposium on Parallel and Distributed Processing (IPDPS), 2003.

[16] V. Petrucci, O. Loques, D. Mosse, R. Melhem, N. A. Gazala, and S. Gobriel. *Thread assignment optimization with real-time performance and memory bandwidth guarantees for energy-efficient heterogeneous multi-core systems*, in Proceedings of the 17th IEEE International Conference on Real-Time and Embedded Technology and Application Symposium (RTAS), 2012.