MULTIPROCESSOR REAL TIME SCHEDULING BASED ON TIME-CONSTRAINT AND ENERGY-CONSTRAINT

RAKOTOARISOA Hasina Patrick[1]- RATSIMBA Mamy Nirina[2] – RASTEFANO Elisée[3]

[1]*PhD Student, Embedded System - Instrumentation - Electronic System Modeling Laboratory,*

[2]*PhD, Embedded System, Instrumentation and Electronic System Modeling,*

[3]*Professor, Embedded System, Instrumentation and Electronic System Modeling,*

Doctoral School of Science and Technics of Engineering and Innovation

Université of Antananarivo, Madagascar

ABSTRACT

This paper deals with a scheduling algorithm for the optimization of the execution time and the energy consumption of a multiprocessor architecture. This algorithm is divided into two parts: (i) real time scheduling based on time constraint, and (ii) scheduling based on energy constraint. The first algorithm gives the allocation and the scheduling of tasks on the processors, while the second algorithm determines the speed that each task will have to be executed to obtain the tradeoff between task execution time and processor power consumption. The result is obtained by determining the optimum execution speeds of each task on the processors because the execution time and the energy consumption are related to the speed of execution.

Keyword: *real time computation scheduling; energy consumption of multiprocessor computer; task execution time*

---

1. INTRODUCTION

Scheduling is the process of allocating resources like processor to the tasks to be executed at a specific time. Real-time scheduling is therefore the selection of real-time tasks. The real time scheduling involves some objectives and in this paper, we use to have a high performance of system with minimum energy consumption. Multiprocessor real-time scheduling is based an association of the tasks with the processors that make up the architecture. One of the major problems of embedded systems is energy autonomy. This autonomy depends on the execution of the tasks on the processors. This article gives a real-time multiprocessor scheduling algorithm that respects the time deadline (hard deadline) and consumes less energy.

2. RELATED WORK

Several works already dealt with multiprocessor real-time scheduling. There are several multiprocessor architectures depending on the nature of the processors, the nature of the communications between processors, and the type of memory [1] [2]. But the general principles of scheduling on multiprocessor architectures are almost the same. We will consider a set of *m* identical processors P_1, P_2,. ., P_m. And we will do the scheduling a set of n periodic/sporadic tasks $\tau = \{\tau_1, \tau_2, ..., \tau_n\}$ on a parallel platform composed of m identical processors: P_1, P_2, ..., P_m.

Scheduling on a multiprocessor architecture must respect the following constraints:

- a processor executes at most one task at every moment;
- a task runs on at most one processor at any given time [1].

There is also work that deals with real-time scheduling of tasks combined with energy consumption. These works are mainly concerned with scheduling on a processor [3] [4].

3. METHODOLOGY

First, the scheduling problem is based on the allocation of tasks on the processors. In this first part, one will have to respect the constraints of time of execution of the tasks. Hence the name of the first algorithm "real time scheduling based on time-constrained".
Then, the second algorithm determines the speed that each task will have to be executed to have the compromise between the execution time of the tasks and the energy consumption of the processors. The result is obtained by determining the optimal execution speeds of each task on the processors because the execution time and energy consumption are related to the execution speed.

3.1 Method for determining the execution time and optimal energy for a task

It is a question of choosing among all the scheduling which maximize the flow, those which spend the least energy. Feasible scheduling of the same set of tasks on multiprocessor machines can lead to different optimal energy consumptions, given the number of processors. It is therefore necessary to find not only the appropriate scheduling, but also the number of processors that gives the minimum energy consumption among the optimal consumptions of the same set of tasks.
According to the existing theories and according to ROHÁRIK VILCU [3], the dissipated power P is a convex and strictly increasing function. It is approximated by a polynomial which is at least of the second degree. We thus take in this memory the value of the power as a function of the speed of the tasks:

$$P(v) = v^{\alpha}$$

where α is the maximum degree of the polynomial, P is the dissipated power, *v* is the speed of the tasks.

So, we have the following two equations to find the minimum execution time and the minimum energy consumed for a task:

$$\begin{cases} C_i = w_i/v \leq D_i & (Eq.1) \\ E_i = v^{\alpha}.w_i/v & (Eq.2) \end{cases}$$

Where C_i is the Computation time associated with the task τ_1; w_i the relative importance of τ_1; D_i the deadline time; and E_i the Energy consumed

According to the first equation, each task must be executed until its deadline to have the minimum energy because the speed decreases according to the execution time. The second equation calculates the energy consumed by each task during its execution.

3.2 Conclusion

In Figure 1 below, the shape of the curve in blue represents the influence of speed on the energy consumed by a task and a red curve on the execution time of a task. We see that the execution time and energy depend of speed execution of task. Here we take the dissipated power : $P(v) = v^{\alpha}$.

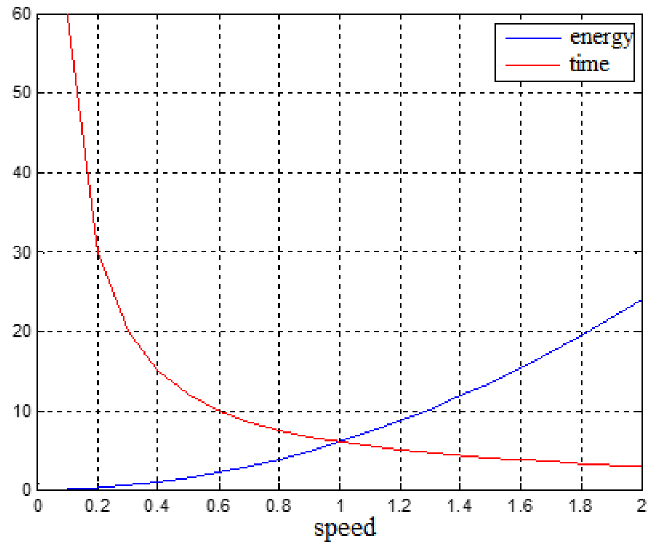

Fig -1 : Characteristics of execution time – computing speed and characteristics of consumed energy of a task – computing speed.

Then, to attain the objectives:

- The execution time must not exceed the time deadline (Eq.1), and.
- the energy consumption is the smallest possible (Eq.2).

We make $C_i = D_i$ (**the execution time must be equal to the time deadline**) and we have the speed to execute a task in the processor.

4. SCHEDULING ALGORITHM

The algorithm is obtained from the optimal scheduling algorithm based on time-constrained. Therefore, it is necessary to use either the EDF algorithm for the case of dynamic priorities, or the RM algorithm for the case of fixed priorities. This scheduling only takes into account the execution time of the tasks.
For the energy consumption, it is necessary to change the speed of execution of tasks. The calculation of this speed constitutes the criterion of scheduling based on energy stress. This change can be a reduction or an increase depending on the timing of each task and also the needs of the application. So, if we reduce the speed, we will gain in energy consumption but the task becomes slow. On the other hand, if one increases the speed, the execution time becomes small but the task consumes more energy.
Finally, the optimization of the execution time of the tasks and the energy consumption of the processors is obtained from these two orderings: scheduling based on time constraint and scheduling based on energy-constrained. The real-time multiprocessor scheduling algorithm based on time-constrained and energy-constrained is done in three steps: the initial allocation of tasks, multiprocessor scheduling and the determination of the new execution time of the tasks and the energy consumption:

Algorithm: Execution time and energy consumption optimization for multiprocessor architecture

1: *Initialize the set of task to schedule* $\boldsymbol{\Gamma = \{\tau_i(C_i, T_i), i = 1, \dots, n\}}$

% schedulability analysis of all tasks on multiprocessor architecture

2: ***if*** τ_i *is schedulable* ***then***

%Allocation of tasks on the processors

3: *Allocate* τ_i *on a processor*

4: *Determine **the speed** that each task will have to be executed using* $\boldsymbol{C_i = D_i - O_i}$

5: *Then rescheduling the tasks with the new computation time.*

6: ***else***

7: τ_i *is not schedulable*

8 : ***fi***

5. APPLICATION

The tasks described in Table 1 below are considered. These tasks are to be scheduled on two processors. In this table, O_i is the start time of the task, C_i the execution time, D_i the time deadline, T_i the period of task, w_i the relative importance of the task i.

Table -1: Table showing the characteristics of the tasks to be scheduled.

Tasks	O_i	C_i	D_i	T_i	w_i
τ_1	0	3	5	5	6
τ_2	5	6	13	13	24
τ_3	0	11	13	13	55

These tasks are schedulable on two processors since the following relationships are verified:

$$\begin{cases} U(\tau) \approx 1.98 \\ U_{max}(\tau) = 0.769 \end{cases}$$

Taking the power dissipation function P (v) = v^3, for the initial allocation of tasks (without optimization of execution time and energy consumption), we have:

- $v_1 = 2$; $v_2 = 4$ et $v_3 = 5$
- On the processor P1 :
  E1 = 24 + 384 = 408 nW.s
- On the processor P2 :
  E2 = 1375 nW.s
- On all processors (two processors) :
  E = E1 + E2 = 1783 nW.s

Then, the execution time of these tasks is t = 11 ns and the energy consumed is E = 1783 nW.s. As a result, we can see the optimal scheduling of three tasks on two processors based on execution constraints, as shown in the Figure 2;

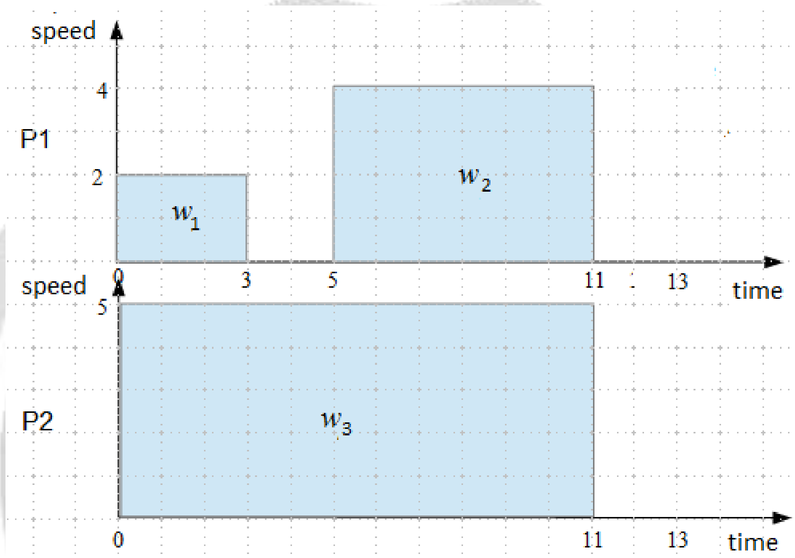

Fig -2 : Optimal scheduling for three tasks on two processors based only the execution time constraint.

Applying our optimization of execution time and energy consumed on these tasks, we have the Figure 3. In this figure the execution time is the same as the time deadline. The dissipated power function remains the same P (v) = v^3.

Then, the execution time of these tasks is t = 13 ns and the energy consumed is E = 1208.74 nW.s.

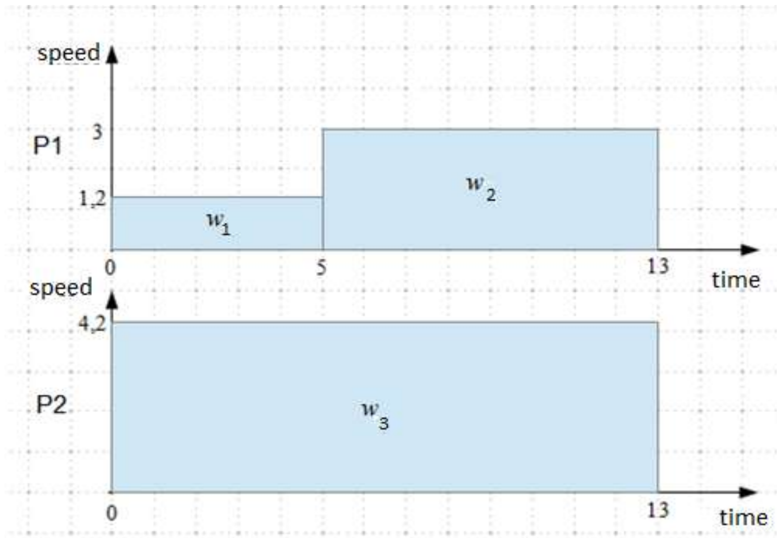

Fig -3 : Optimal scheduling for three tasks on two processors based on time and energy constraints.

6. DISCUSSION AND CONCLUSION

The theory shows that we can reduce energy consumption by using a good scheduling algorithm. To arrive at the objective, it is necessary to consider the execution time of the tasks and as well as its speed of execution on the processors. Since the result obtained has no influence on the behavior of the system (the deadlines are respected), then our algorithm is valid but other aspects can be considered also to reach the minimization of energy consumption of the embedded systems.

The work presented in his paper is multiprocessor real time scheduling based on time and energy constraints. The energy consumed is proportional to the speed of execution of the tasks. Therefore, it is shown that the execution time of a task must be equal to its deadline so that the energy consumed is the minimum possible. And then, the execution time and the energy consumed are optimized.

7. REFERENCES

[1]. Joël GOOSSENS, « *Ordonnancement multiprocesseur* », Université Libre de Bruxelles, ETR'07 - 6 septembre 2007.
[2]. Mohamed MAROUF, *« Ordonnancement temps réel dur multiprocesseur tolérant aux fautes appliqué à la robotique mobile »*, Thèse de Doctorat, Ecole Nationale Supérieure des mines de Paris, 01 Juin 2012.
[3]. Dana - Mihaela BOHARIK VÎLCU, « *Ordonnancement optimal de tâches pour la consommation énergétique du processeur»,* Thèse de Doctorat, Université Paris XII - Val de Marne, 19/02/2004.
[4]. Vincent NELIS, *« Ordonnancement de processus dans les systèmes temps réel monoprocesseurs sous contraintes de temps et d'énergie »,* Université Libre de Bruxelles.