On the correctness of "Gang EDF Scheduling of Parallel Task Systems" (Kato, Ishikawa, RTSS'09)

Pascal Richard(1), Joël Goossens(2)

(1) *LIAS/University of Poitiers, France*
(2) *Université libre de Bruxelles (ULB), Belgium*

Abstract

This short note describes two correctness problems in the paper entitled "Gang EDF Scheduling of Parallel Task Systems", Shinpei Kato and Yutaka Ishikawa, presented at RTSS'09.

Key words: Real-time Gang Scheduling, Schedulability Test, Gang EDF Algorithm

1 Introduction

This note focuses on real-time Gang Scheduling of real-time tasks upon multiprocessor platforms. We raise two correctness issues in the schedulability test presented in [2]: "Gang EDF Scheduling of Parallel Task Systems", Shinpei Kato and Yutaka Ishikawa, presented at RTSS'09.

This paper deals with preemptive scheduling of parallel real-time tasks on identical multiprocessor platforms. In this parallel task model, job parallelism is allowed meaning that a job is simultaneously executed on different processors. The task parallelism is said rigid since the tasks always simultaneously use a predefined number of processors. This model is also named Gang scheduling in the literature. The classical task model is a particular case of the Gang scheduling one in which every task uses exactly one processor.

The paper [2] presents a Gang scheduling algorithm named Gang EDF and its schedulability test named [**KAT**] in the following. This schedulability test

Email address: `pascal.richard@univ-poitiers.fr`; `joel.goossens@ulb.ac.be` (Pascal Richard(1), Joël Goossens(2)).

extends the techniques used in Global EDF schedulability [1] (named [**BAR**] in the following) to Gang real-time tasks. The very attractive idea behind the Gang test [**KAT**] is: if all tasks use exactly one processor, then the Gang test is equivalent to [**BAR**].

In this note, we show through counterexamples that the test is not correct on its principle. The test is based on a necessary condition for a task to miss a deadline that is erroneous. Furthermore, the time interval considered to check the schedulability of a Gang task is not valid when some tasks simultaneously use several processors. From our understanding, [**KAT**] is valid and fully equivalent to [**BAR**] when all the tasks use a single processor at a time (i.e., in the classical task model).

In the following, Section 2 recalls the task model and Section 3 summarizes the main principles of [**KAT**]. Section 4 presents a counterexample showing that (i) the necessary condition to check if a task misses its deadline is not correct and (ii) the time interval considered in the test is not valid in the general case in which tasks are allowed to use simultaneously several processors. Lastly, Section 5 provides a short discussion for concluding.

2 The Gang scheduling model

[**KAT**] considers rigid parallel tasks, named hereafter Gang, to be executed upon m identical processors.

2.1 Task model

Let $\tau = \{\tau_1, \ldots, \tau_n\}$ be a set of n sporadic tasks with $\tau_i = (v_i, C_i, D_i, T_i)$ is characterized by the number v_i of used processors, a worst-case execution time C_i when executed in parallel on v_i processors, a minimum inter-arrival time T_i and a constrained relative deadline D_i (i.e., $D_i \leq T_i$). The utilization of τ_i is $U_i = C_i/T_i$.

Each task generates an infinite sequence of jobs. The execution of a job of τ_i is represented as a $C_i \times v_i$ rectangle in time $\times$ processor space. Every job must be completed by its deadline.

2.2 Gang EDF

Gang EDF is a priority scheme which applies EDF (Earliest Deadline First) basically in the same way as the classical Global EDF: jobs with earlier deadlines are assigned higher priorities. At most m processors are used by Gang tasks at any time. Clearly, Gang EDF is equivalent to Global EDF if every task uses a single processor (i.e., $v_i = 1, 1 \leq i \leq n$).

3 [KAT] schedulability test

In this section, we present the main principles of the Gang EDF schedulability test [**KAT**]. The presented materials come from [2].

3.1 Test principles

[**KAT**] is a generalization of [**BAR**]. [**BAR**] is a schedulability test based on a time demand analysis for Global EDF. Basically, the test exhibits a *necessary schedulability condition* on the parameters of all the tasks that must be satisfied in order to exhibit a deadline miss for a given job (i.e., the problem job). Then, the contrapositive yields a *sufficient schedulability condition* for the considered scheduling algorithm.

[**KAT**] test considers any legal sequence of job request of task system τ on which a deadline is missed by Gang EDF. Assume that τ_k is generating the problem job at time t_a that must be completed by its deadline at time $t_d = t_a + D_k$. Let t_0 be the latest time instant before or at t_a at which at least v_k processors are idled and $\Delta_k = t_d - t_0$. A necessary condition for the problem job to miss its deadline is: higher priority tasks are blocking τ_k for strictly more that $D_k - C_k$ in the interval $[t_a, t_d)$. Since τ_k requires v_k processors, it is blocked while $m - v_k + 1$ processors are busy. This minimum interference necessary for the deadline miss is defined by the *interference rectangle* whose width w_k and height h_k are respectively given by:

$$w_k = \Delta_k - C_k \tag{1}$$
$$h_k = m - v_k + 1 \tag{2}$$

Let $I_k(\tau_i, \Delta_k)$ be the worst-case interference against the problem job over $[t_0, t_d)$, meaning that it blocks the problem job over $[t_a, t_d)$ and is executed over $[t_0, t_a)$.

If the problem job misses its deadline, then it is necessary that the total amount of work that interferes over $[t_0, t_d)$ exceeds the interference rectangle:

$$\sum_{\tau_i \in \tau} I_k(\tau_i, \Delta_k) > w_k \times h_k \tag{3}$$

Notice h_k is fixed while w_k is not (since Δ_k is not determined). Thus, the previous condition must be checked for all values of Δ_k. Δ_k will be proved to be bounded in practice (described next in Section 3.2).

The interference must take into account carry-in jobs in the interference rectangle who arrive before t_0 and have not completed execution by t_0. Following [**BAR**], the [**KAT**] test distinguishes the interference coming from tasks without or with a carry-in job, respectively denoted $I_1(\tau_i, \Delta_k)$ and $I_{carry-in}$. Several bounds of these interference contributions are defined (Section 4.1 and 4.3 in [2]). The schedulability test [**KAT**] checks a task system by using the following result:

Theorem 1 *It is guaranteed that a task system τ is successfully scheduling by Gang EDF upon m processors, if the following condition is satisfied for all tasks $\tau_k \in \tau$ and all $\Delta_k \geq D_k$:*

$$\sum_{\tau_i \in \tau} I_1(\tau_i, \Delta_k) + I_{carry-in} \leq w_k \times h_k \tag{4}$$

Several bounds have been proposed in [2] to evaluate the accumulated interference $\sum_{\tau_i in \tau} I_k(\tau_i, \Delta_k)$. We limit ourselves to use the first proposed bound (named Simple Bounds and defined by Equation (6) in [2]):

$$I_1(\tau_i) = \min(\text{hbf}(\tau_i, \Delta_k), w_k) \times \min(v_i, h_k) \qquad \text{if } i \neq k$$
$$I_1(\tau_i) = \min(\text{hbf}(\tau_i, \Delta_k) - C_k, A_k) \times \min(v_i, h_k) \qquad \text{if } i = k$$

with $A_k = \Delta_k - D_k$ that defines the maximum contribution of τ_k in the feasibility interval $[t_0, t_d)$. We recall that the horizontal demand bound function defined in Equations (1-2) in [2]:

$$\text{dbf}(\tau_i, L) = \max\left(0, \left\lfloor \frac{L - D_i}{T_i} \right\rfloor + 1\right) \times C_i \times v_i$$
$$\text{hbf}(\tau_i, L) = \text{dbf}(\tau_i, L) \times \frac{1}{v_i}$$

Consequently, we can simplify as: $\text{hbf}(\tau_i, L) = \max\left(0, \left\lfloor \frac{L-D_i}{T_i} \right\rfloor + 1\right) \times C_i$.

This result stated in Theorem 1 defines the basis of the [KAT] schedulability test and it is not formally proved in the paper [2].

3.2 Bounding feasibility interval

In order to use the test (Theorem 1), Δ_k must be bounded to define a finite time interval to test possible values for Δ_k. The upper bound presented in [2] and presented hereafter leads to a pseudo-polynomial schedulability test.

Theorem 2 *If Condition (4) is to be violated for any Δ_k, then it is violated for some $\Delta_k \geq D_k$ satisfying Condition (5), where $C_{carry-in}$ denotes $\sum_{\tau_i \in \tau_{carry-in}} C_i$.*

$$\Delta_k \leq \frac{h_k C_k - \sum_{\tau_i \in \tau}(D_i - T_i)U_i \times \min(v_i, h_k) + C_{carry-in}}{h_k - \sum_{\tau_i \in \tau} U_i \times \min(v_i, h_k)} \quad (5)$$

where $\tau_{carry-in}$ is the set of tasks with a carry-in job.

The proof provided in [2] comes from a classical linearization of the floor functions starting from the schedulability condition:

$$\sum_{\tau_i \in \tau} \text{hbf}(\tau_i, \Delta_k) \times \min(v_i, h_k) + C_{carry-in} > w_k \times h_k \quad (6)$$

4 Correctness problems

4.1 Necessary Condition (3) is not valid

We next show that the necessary condition stated in the Inequality 3 is not valid. Let us consider the task set defined in Table 1 and a platform with 2 processors. A Gang EDF schedule is depicted in Figure 1 (the synchronous periodic case) and this task set is feasible for all possible task release scenarios. Please notice that the task set schedulability is not mandatory for exhibiting the correctness problem.

We analyze the task τ_2 in task set of Table 1: $k = 2$. τ_2 is the problem job. The feasibility interval is delimited by: $t_0 = t_a = 0$ and $t_d = D_2 = 4$; $\Delta_2 = t_d - t_0 = 4$ and $A_2 = \Delta_2 - D_2 = 0$. The scenario $\Delta_2 = D_4$ is the first scheduling point considered in Theorem 1. There is no carry-in job in this example since we analyze the first job of τ_2. The interference rectangle is:

Tasks	v_i	C_i	D_i	T_i
τ_1	1	2	4	4
τ_2	2	2	4	4
τ_3	1	1	4	4

Table 1
Gang task set for counter-example 1 to be executed upon a 2-processor platform

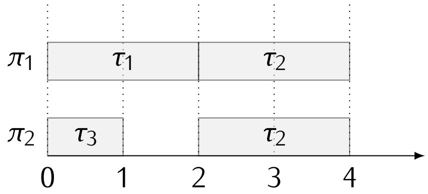

Fig. 1. Gang EDF schedule for the task set defined in Table 1

$$w_2 = \Delta_2 - C_2 = 4 - 2 = 2$$
$$h_2 = m - v_2 + 1 = 2 - 2 + 1 = 1$$

Let us now compute the demand bound functions for the tasks in the counter examples:

$$\text{hbf}(\tau_1, \Delta_2) = \max\left(0, \left\lfloor \frac{4-4}{4} \right\rfloor + 1\right) \times 2 = 2$$
$$\text{hbf}(\tau_2, \Delta_2) = \max\left(0, \left\lfloor \frac{4-4}{4} \right\rfloor + 1\right) \times 2 = 2$$
$$\text{hbf}(\tau_3, \Delta_2) = \max\left(0, \left\lfloor \frac{4-4}{4} \right\rfloor + 1\right) \times 1 = 1$$

Now we compute the interference bounds:

$$I_1(\tau_1, \Delta_2) = \min(2, 2) \times \min(1, 1) = 2$$
$$I_2(\tau_2, \Delta_2) = \min(2 - 2, 4 - 4) \times \min(2, 1) = 0$$
$$I_3(\tau_3, \Delta_2) = \min(1, 2) \times \min(1, 1) = 1$$

Hence, the total cumulative interference is:

$$\sum_{\tau_i \in \tau} I_1(\tau_i) = 3 > w_2 \times h_2 = 2$$

According to the previous inequation, τ_2 necessarily misses its deadline, which is not correct.

4.2 The feasibility interval is not valid

From our understanding, all mathematical derivations performed in the proof of Theorem 2 in [2] are correct. Nevertheless, we will exhibit a problem that comes from the starting assumption defined in Inequality (6). The reader can refer to [2] for the definition of $C_{carry-in}$ [1]. Next, we will only use the fact that $C_{carry-in} \geq 0$ (i.e., a carry-in job can only define an interference greater than or equal to zero to the problem job).

In Theorem 2, the numerator is always a positive value since $D_i \leq T_i, 1 \leq i \leq n$, and all used values are positive or zero. Thus, the numerator is always a positive value. Using a counterexample, we will see that the denominator is not always positive. As a consequence, this raises a correctness issue for applying the schedulability test over a time interval which has a negative length.

Consider the task set $\tau = \{\tau_1, \tau_2, \tau_3\}$, $\tau_1 = (1, 3, 4, 4)$, $\tau_2 = (2, 1, 4, 4)$ and $\tau_3(1, 2, 4, 4)$ to be executed upon $m = 2$ processors. For testing task τ_2 using Theorem 1, we need to bound Δ_2 using Theorem 2. We prove hereafter that such a bound is negative for the counterexample.

Consider the denominator of Inequality (5): $h_k - \sum_{\tau_i \in \tau} U_i \times \min(v_i, h_k)$. For task τ_2, we first compute $h_2 = m - v_2 + 1 = 2 - 2 + 1 = 1$; this implies that $\min(v_i, h_2) = 1, 1 \leq i \leq n$ and thus $h_2 - \sum_{\tau_i \in \tau} U_i = 1 - (\frac{3}{4} + \frac{1}{4} + \frac{2}{4}) = -\frac{1}{2}$. As a consequence the denominator is negative. Thus, the upper bound computed by Theorem 2 of the time interval while checking the schedulability of a task has a negative length which is not valid in with respect to the schedulability test defined in Theorem 1.

5 Conclusion

We exhibited two correctness problems in this note. The first one concerns necessary condition (3) and the arguments used to compute the contribution to the called *interference rectangle*. The second one concerns the feasibility interval. As previously said, the mathematical derivations in the proof of the Theorem 2 are valid. Nevertheless, the Inequality (6) used as a schedulability condition for computing an upper bound of Δ_k leads to a correctness problem of Theorem 2. Currently, we have not been able to fix these problems to repair the test.

Notice that if every task verifies $v_i = 1, 1 \leq i \leq n$, then [**KAT**] is the equivalent

[1] based on an a knapsack-based heuristic for selecting worst-case interference generated by the tasks — those tasks are in the set $\tau_{carry-in}$ in Theorem 2

to [**BAR**]. But, the generalization proposed in [2] seems not to be as simple as expected to cope with general gang task systems.

References

[1] Baruah, S. Techniques for multiprocessor global schedulability analysis. In *28th IEEE International Real-Time Systems Symposium, 2007* (December 2007), pp. 119–128.

[2] Kato, S., and Ishikawa, Y. Gang EDF scheduling of parallel task systems. In *30th IEEE Real-Time Systems Symposium, 2009* (December 2009), pp. 459–468.