Best-Case Response Time Analysis for Precedence Relations in Hard Real-Time Systems

Pierre-Emmanuel Hladik and Anne-Marie Déplanche
IRCCyN, UMR CNRS 6597
44321 Nantes Cedex 3 - France
{hladik | deplanche}@irccyn.ec-nantes.fr

Abstract

We present an approximate computation of the best-case response times of periodic hard real-time tasks with fixed priorities in a single processor environmen. In particular we are interested in the problem of local precedence relations. After exposinf some difficulties to take into account precedence relations, we give a simple way to solve this problem.

1. Introduction

Analysis of fixed priority pre-emptive tasks known as RMA (Rate Monotonic Analysis) has been well developed since the original paper of Liu and Layland in 1973 [LL73]. Now the RMA technique is extended to a large set of systems including task synchronisation, mixture of periodic and aperiodic tasks, distributed systems, etc.

Since Palencia and al. [PGH98] who proposed to improve the computation of the worst-case response time in distributed hard real time systems by computing the best-case one, some proposals have emerged to compute the best-case response time. In [Goo99] and [HKR01] a scenario that produces the best-case response time was exhibited, and in a recent article Redell and al. [RS02] presented an exact method to solve this problem. All these methods consider a model of independent tasks and they propose to take into account precedence relations thanks to jitter terms, but it is pessimistic

This paper is the continuation of a precedent work [HD03] in which a method is presented to compute the worst-case response time for hard real-time tasks with precedence relations in a multiprocessor environment. The method was inspired by [HKL91] where local precedence relations are not only replaced by jitters but also directly integrated in the scheduling analysis. The goal of the present paper is to expose a similar method to compute the best-case response time.

This paper is organised as follows: the next section gives the computational model. In the section 3, we discuss about other methods and problems that address local precedence relations. Section 4 proposes a computation of the best-case response time for our model. Section 5 concludes and investigates some perspectives.

2. The model

The computational model that we consider is equivalent to the linear model of an event-driven system presented in [PGH98]. In this model there is a set of external event sequences that activate tasks. These tasks may generate internal events that activate other tasks, and so on.

The system that we consider is a single processor scheduled under a pre-emptive fixed priority strategy. The application software is composed of *n periodic precedence chains* (figure 1.a) denoted by $\chi_1,\ldots,\chi_n$. Each periodic precedence chain χ_i has a *period* (T_i), a *jitter* (Ji) and is composed of a set of $m(i)$ tasks $\tau_i=\{\tau_{i1},\ldots,\tau_{im(i)}\}$. Each activation of a periodic precedence chain is called a *job*. The relative phasing between the different first jobs of these precedence chains is arbitrary.

Each *task* is characterised by its worst-case computation time (C_{ij}^{w}), its best-case computation time (C_{ij}^{b}), and a priority (P_{ij}) (a task τ_{ij} has a higher priority than a task τ_{ab} iff $P_{ij} > P_{ab}$). Priorities are assumed to be unique.

We state the following rules concerning task activation.

Rule 1: If ϕ_i is the arrival time of the first job of χ_i then the k^{th} instance of τ_{i1} may be released at any time between $\phi_i+(k-1)T_i$ and $\phi_i+(k-1)T_i+J_i$.

Rule 2: The k^{th} instance of τ_{ij}, $2 \le j \le m(i)$ is released when the execution of the k^{th} instance of τ_{ij-1} is

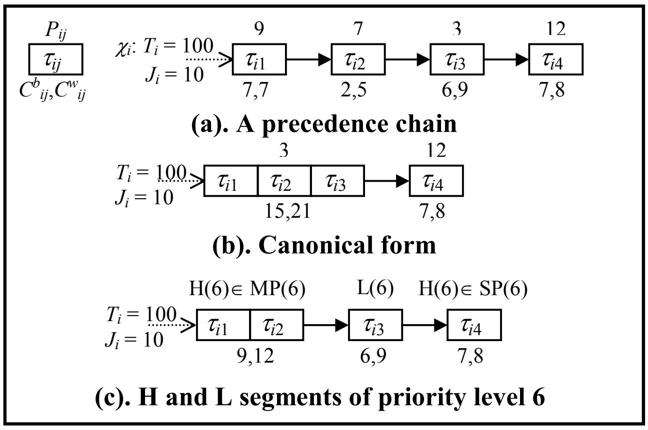

Figure 1. Precedence chain

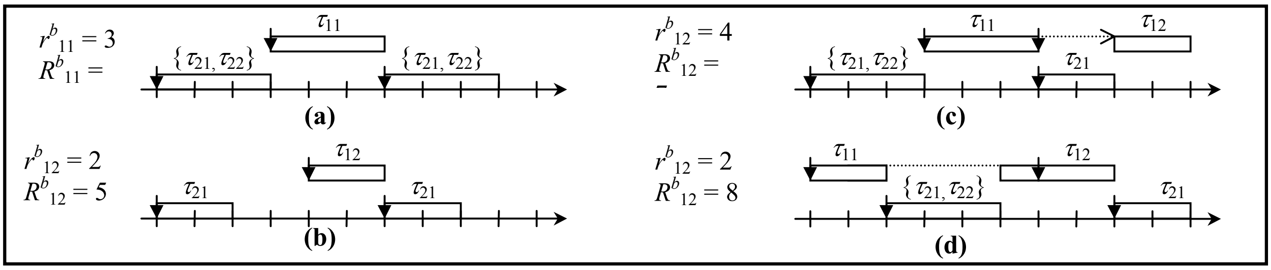

Figure 2. Example of some execution sequences

completed, and only if the execution of the $(k\text{-}1)^{st}$ instance of τ_{ij} is completed. Re-entrance of a periodic precedence chain is impossible.

In [HKL91], Harbour and al. described a transformation on precedence chains, calls *canonical form*, and proved that converting a precedence chain into its canonical form does not change its completion time. Due to lack of space we do not develop this point, we just note that consecutive tasks of a precedence chain in canonical form do not decrease in priority. The figure 1.b illustrates the canonical transformation. We consider from now on that the precedence chain for which we compute the best-case response time, is in canonical form.

For a task τ_{ij}, we call *local* best-case response time, r_{ij}^{b}, the difference between the completion time of τ_{ij} and its release time, and *global* best-case response time, R_{ij}^{b}, the difference between the completion time of τ_{ij} and the arrival time of χ_i.

We call an *execution sequence* a behaviour of the system (release times of all jobs, and execution times of all instances of all tasks).

The best-case response time computation for a task consists in finding an execution sequence, calls *best-case phasing*, for which the task has improved its smallest global response time. To our knowledge, only [PH98] and [RS02] proposed a best-case phasing for a model of independent tasks with jitter. In the next section we will discuss about this.

3. Related work

We will illustrate our matter with the next example: the system is composed of two precedence chains of two tasks, table 1 describes it. Jitters equal zero.

	T_i	τ_{ij}	C_{ij}^{b}	P_{ij}
χ_1	30	τ_{11}	3	1
		τ_{12}	2	3
χ_2	6	τ_{21}	2	4
		τ_{22}	1	2

Table 1: Task set characteristics

We just consider the best-case response time for tasks of χ_1. τ_{11} can only be pre-empted by $\{\tau_{21},\tau_{22}\}$ and τ_{12} by $\{\tau_{21}\}$.

3.1. Redell's hypothesis

In [RS02], Redell and al. proved that, for a set of independent tasks with jitter:

the best-case phasing of a task occurs whenever ***it finishes simultaneously*** *with the release of all its higher priority tasks and these have experienced their maximum release jitter.*

If we consider τ_{11} and τ_{12} as independent tasks (the simplicity of the example allows us to ignore jitter) and observe the rule on best-case phasing, $r_{11}^{b} = 3$ and $r_{12}^{b} = 2$ (figures 2.a and 2.b). Traditionally an approximation of the global response time of a task τ_{ij} ($j > 1$) is:

$$R_{ij}^{b} = R_{ij-1}^{b} + r_{ij}^{b} \tag{1}$$

Thus τ_{12} has a global best-case response time of 5.

But if we now consider precedence relations, the figure 2.c shows the best-case phasing and τ_{12}'s global best-case response time equals 7. Thus, the equation (1) is only a lower bound.

If we try to integrate precedence relations into the rule of best-case phasing, a naïve idea is to "phase" higher priority tasks with the last task of the precedence chain and to build the execution sequence that results from this. Figure 2.d illustrates this case and the global response time of τ_{12} is 8. Thus, it is not a true best-case phasing.

An other idea is to compute the response times for each execution sequence where higher priority tasks are "phased" with one task of the considered precedence chain and find the best. For our example it is easy, there are only two cases (figure 2.c and 2.d) but the problem has an exponential complexity if other precedence chains are added.

3.2. Palencia's hypothesis

In [PGH98] Palencia and al. proposed an approximation of the best-case phasing for a set of independent tasks with jitter:

the best-case phasing of a task occurs whenever ***it is released simultaneously*** *with the completion of all tasks of higher priority and their next releases occur with a maximum delay.*

This rule produces a lower bound of best-case response time because it is impossible that all higher priority tasks just complete simultaneously.

If we consider precedence chains as a set of independent tasks with jitter, the same approximation as Redell's rule is obtained. However, in the next section, we integrate precedence chains in this best-case phasing.

4. Best-case response time

Before getting further into the method, we need to give some definitions.

4.1. H/L segments and MP/SP sets

These definitions come from [HKL91]. Let P denote a priority level. We refer to consecutive tasks in a precedence chain as a segment. An H(P) segment is a segment each task of which has a priority greater than P. In the same way, an L(P) segment refers to a segment each task of which has a priority lower than P.

Given an H segment H_k, let h_k^b be its best-case execution time i.e. the sum of the best-case execution times of tasks in H_k.

Let MP_{ij} denote the set of tasks that can have more than one pre-emption effect on the task τ_{ij}. This set is composed of all H(P_{ij}) segments issued from precedence chains different from χ_i, and that are not preceded by an L(P_{ij}) segment. SP_{ij} is the set of singly pre-emptive tasks on τ_{ij}. It is composed of all H(P_{ij}) segments issued from precedence chains different from χ_i, and that are preceded by an L(P_{ij}) segment.

We denote $MP_{ij}.\chi_m$ the H segment of MP_{ij} belonging to χ_m. $MP_{ij}.\chi_m$ is at the beginning of χ_m and inherits its jitter J_m.

The figure 1.c shows an example of H segments belonging to MP and SP sets.

4.2. Computation of best-case response time

Finding the best-case phasing for a task is similar to exhibit an execution sequence where the interference the task suffers is minimum. Firstly, lemma 1 states that for a task τ_{ij}, there is always an execution sequence where none H segment of SP_{ij} interferes with τ_{ij}. For that we use the concept of τ_{ij}-busy period introduced by Lehoczky in [Leh90]. Using lemma 1, we state that τ_{ij} experiments its best-case response time for an execution sequence where none H segment of SP_{ij} interferes.

Lemma 1. For an execution sequence and the k^{th} instance of τ_{ij}, denoted τ_{ij}^k, if a H segment of SP_{ij} interferes with τ_{ij}^k, then there exists an execution sequence where this H segment doesn't interfere with τ_{ij}^k and where τ_{ij}^k finishes earlier.

Proof. Let H_a be the H segment of SP_{ij} which interferes with τ_{ij}^k, and χ_a its precedence chain. By definition, the activation of H_a can only occur at the beginning of the τ_{ij}-busy period that contains τ_{ij}^k. By delaying χ_a judiciously in the above execution sequence, H_a can no more start the busy-period and thus is rejected out of the busy-period.

Moreover, it's impossible that a new job of χ_a interferes with this busy-period. And the new execution sequence preserves all arrival times for all other precedence chains. Consequently, the length of the busy-period decreases and thus τ_{ij}^k finishes earlier. Indeed, a τ_{ij}-busy period ends with the completion of τ_{ij}. ■

Theorem 1. There is no H segment of SP_{ij} that interferes in the τ_{ij}-busy period that produces the best-case response time of τ_{ij}.

Proof. Let us suppose that an H segment of SP_{ij} interferes in the busy period that produces the best-case response time of τ_{ij}. Lemma 1 proves that we can produce an execution sequence where the response time of τ_{ij} is better. This contradicts the hypothesis. ■

From now on, segments of SP are no more taking into account for the computation of the best-case response time. The best-case phasing problem consists in finding the best-case phasing for the segments of MP. Lemma 2 gives an intermediate result on MP used for the best-case phasing between two precedence chains.

Lemma 2. Let χ_i in canonical form, if we have a segment $H_1 = MP_{ij+1}.\chi_m$ then there exists a segment $H_2 = MP_{ij}.\chi_m$ such $h_2^b \geq h_1^b$.

Proof. By definition, if χ_i is in canonical form then $P_{ij} < P_{ij+1}$. Let $\tau_{ab} \in H_1$, by definition we have $P_{ab} > P_{ij+1} > P_{ij}$, thus $\tau_{ab} \in H_2$. Hence, $H_1 \subset H_2$ and by definition of h_1^b and h_2^b, we have $h_2^b \geq h_1^b$. ■

Theorem 2. Given a precedence chain in canonical form χ_i and another precedence chain χ_j, where $MP_{i1}.\chi_j$ is not empty, the smallest interference of χ_j with χ_i occurs when χ_j is phased such that no work of $MP_{i1}.\chi_j$ is pending when χ_i is released and, all the following activations of χ_j occur with a maximum delay.

Proof. The first part of the proof is strongly inspired from [PGH98]. Let us suppose that χ_i is activated during the execution of an instance of $MP_{i1}.\chi_j$. If we move the activation time of χ_i (the same one as τ_{i1}) further up in time until the completion of $MP_{i1}.\chi_j$, the completion time of τ_{i1} is unchanged. Consequently release times as well as completion times of the other tasks of χ_i remain the same. The global completion time of χ_i will be better since it is counted from a starting date that is later.

The second part of the proof is different because of precedence relations. Let us consider the activations of χ_j that occur after χ_i has been activated, and that interfere with that job of χ_i. By moving theses activations as further up in time as possible, we can only increase the chance of χ_i to finish earlier, because there is more idle time for execution of all tasks of χ_i and because, according to lemma 2, the interference of χ_j on a task of χ_i is less on its successive tasks. ■

Using theorem 2, we extend results of [PGH98] to precedence chains to find a best-case phasing:

the best-case phasing of a precedence chain χ_i occurs whenever ***it is released simultaneously*** *with the completion of all $MP_{i1}.\chi_j$ ($1 \le j \le n$, $j \ne i$) and the next releases of all χ_j occur with a maximum delay.*

As for Palencia, it is impossible that such an execution sequence occurs, because H segments cannot end simultaneously. However we can use this rule to compute a lower bound of the best-case response time.

The computation of the global best-case response time of the precedence chain χ_i requires to compute the global response time for each task of χ_i. The best-case response time of τ_{i1} is given by:

$$R_{i1}^{\,b} = \min\left\{ t > 0 \,\middle|\, \sum_{\substack{H = MP_{i1}.\chi_k \\ k \ne i}} \left\lceil \frac{t - T_k - J_k + R_{k1}^{\,b}}{T_k} \right\rceil_0 h_k^{\,b} + C_{i1}^{\,b} = t \right\}$$

where $R_{k1}^{\,b}$ is the global best-case response time of $MP_{i1}.\chi_k$, and:

$$\lceil x \rceil_0 = \max\left(0, \lceil x \rceil\right)$$

Given the global best-case response time of τ_{ij} ($1 \le j < m(i)$) and, the best-case phasing, it is possible to compute the global best-case response time of τ_{ij+1}:

$$R_{ij+1}^{\,b} = \min\left\{ t > 0 \,\middle|\, R_{ij}^{\,b} + \sum_{\substack{H = MP_{ij+1}.\chi_k \\ k \ne i}} \left(\left\lceil \frac{t - J_k + R_{k1}^{\,b}}{T_k} \right\rceil_0 - \left\lceil \frac{R_{ij}^{\,b} - J_k + R_{k1}^{\,b}}{T_k} \right\rceil_0 \right) h_k^{\,b} + C_{ij+1}^{\,b} = t \right\}$$

The best-case analysis requires the knowledge of the best-case response times of the different segments of MP. In a first time, given an H segment we can consider its global best-case response time equal to the sum of the best-case execution times of all its tasks ($R^b = h^b$).

Improving this estimation by the computation of the best-case response time for all intermediate tasks of a precedence chain requires further work. Nevertheless, the best-case response time of the intermediate tasks of the canonical form are obtained with this approach.

5. Conclusions

We have presented an approach to compute a lower bound of the best-case response time of a precedence chain in a single processor environment. We are now extending this method to distributed systems: a precedence chain distributed over different nodes is considered as independent sub-chains where network delays are translated into jitters attached to sub-chains.

6. References

[Goo99] J. Goossens, "Scheduling of hard real-time periodic systems with various kinds of deadline and offset constraints", PhD Thesis, Université Libre de Bruxelles, 1999.

[HKL91] M. González Harbour, M.H. Klein, and J.P. Lehoczky, "Fixed Priority Scheduling of Periodic Tasks with Varying Execution Priority", in proceeding of Real-Time Systems Symposium, pp. 116-128, San Antonio (Texas), 1991.

[HKR01] W. Henderson, D. Kendall and A. Robson, "Improving the Accuracy of Scheduling Analysis Applied to Distributed Systems", Intl. Journal of Real-Time Systems, Kluwer, 20(1):5-25, January 2001.

[HD03] P.E. Hladik and A.M. Déplanche, "An Extension of Holistic Schedulability Analysis for Precedence Relations in Multiprocessor Systems", 15th Euromicro Work In Progress on Real-Time Systems, 2003.

[Leh90] J.P. Lehoczky, "Fixed Priority Scheduling of Periodic Task Sets with Arbitrary Deadline", IEEE Real-Time System Symposium, 1990.

[LL73] C.L. Liu, and J.W. Layland, "Scheduling Algorithms for Multiprogramming in a Hard Real-Time Environment", Journal of the ACM, 20(1) :46-61, 1973.

[PGH98] J.C. Palencia, J.J. Gutiérrez Garcia, and M. González Harbour, "Best-case Analysis for Improving the Worst-Case Schedulability Test for Distributed Hard Real-Time Systems", 10th Euromicro Workshop on Real-Time Systems, 1998.

[RS02] O. Redell and M. Sanfridson, "Exact Best-Case Response Time Analysis of Fixed Priority Scheduled Tasks", 14th Euromicro Conference on Real-Time Systems, Vienna (Austria), 2002.