Exact best-case response time analysis of real-time tasks under fixed-priority pre-emptive scheduling for arbitrary deadlines
Bril, R.J.; Cucu-Grosjean, L.; Goossens, J.

Published in:

Published: 01/01/2009

Document Version
Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the author's version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

Citation for published version (APA):
Abstract

In this paper, we present a conjecture for exact best-case response times of periodic released, independent real-time tasks with arbitrary deadlines that are scheduled by means of fixed-priority pre-emptive scheduling (FPPS). We illustrate the analysis by means of an example. Apart from having a value on its own whenever timing constraints include lower bounds on response times of a system to events, the novel analysis allows for an improvement of existing end-to-end response time analysis in distributed systems, i.e. where the finalization of one task on a processor activates another task on another processor.

1. Introduction

Real-time systems are systems that provide correct and timely responses to events in their environment. The term timely means that the timing constraints imposed on these responses must be met. The real-time software of these systems is typically designed as a set of tasks and a scheduling algorithm that determines the order in which the tasks are executed. In such a setting, the timing constraints on the responses of the system give rise to derived timing constraints on the responses of the tasks. In this paper, we consider fixed-priority pre-emptive scheduling (FPPS), which is currently considered to be a de-facto standard for real-time scheduling in industry. Typically, timing constraints are interpreted as upper bounds on response times of a system and its tasks to events, i.e. responses should not be too late. Accordingly, the vast majority of books and papers addressing systems based on FPPS focus on methods for worst-case analysis in general and worst-case response time analysis in particular.

Whenever timing constraints include lower bounds on response times of a system to events, i.e. when responses should not be too early, methods for best-case analysis become important as well. A well-known example is an airbag, which must neither be inflated too early nor too late upon a collision. Another example is WiseMAC [4], where information must be sent in intervals of time during which the receiver is awake. Notably, the seminal work on response time analysis for FPPS by Harter [7, 8] already covers both worst-case and best-case response time analysis. The need for best-case response time analysis has later also been identified by others in the area of (finalization) jitter of periodic tasks in general and in the area of distributed systems in particular [3, 9, 10, 17, 18].

Worst-case response time analysis for FPPS has been addressed extensively in the literature, and many restrictions of the original scheduling model [13] have been lifted in later work. As examples, [16] introduced the notion of a sporadic task next to a periodic task, [12] address (worst-case) relative deadlines smaller than periods and [11, 22] (worst-case) relative deadlines larger than periods, [20] lifted independent tasks to tasks with mutual access to
shared resources (other than the processor) by presenting the priority ceiling protocol, [1, 22] address tasks with activation jitter, [6, 14, 15, 21] consider tasks with a specific phasing rather than arbitrary phasing, [5] introduced FPPS with varying priorities, and [19, 23] address scheduling with pre-emption thresholds. The scheduling models for best-case response time analysis [3, 8, 18] are considerably less advanced, however. Compared to the original scheduling model, the following advancements are facilitated: (worst-case) relative deadlines are also allowed to be smaller than periods and tasks can have activation jitter.

In this paper, we improve existing analysis by presenting a conjecture for exact best-case response time analysis for tasks with arbitrary deadlines. We illustrate the analysis by means of an example.

This paper is organized as follows. We present our scheduling model for FPPS in Section 2 and we briefly recapitulate existing best-case response analysis in Section 3. Our conjecture for exact best-case response analysis for arbitrary deadlines is the topic of Section 4. In Section 5, we present an example illustrating our novel analysis. The paper is concluded in Section 6.

2. A basic scheduling model for FPPS

We assume a uniprocessor system and a set $T$ of $n$ periodically released, independent tasks $\tau_1, \tau_2, \ldots, \tau_n$ with unique, fixed priorities. At any moment in time, the processor executes the highest priority task that has work pending, i.e. tasks are scheduled using FPPS.

Each task $\tau_i$ generates an infinite sequence of jobs $t_{ik}$ with $k \in \mathbb{Z}$. The inter-activation times of $\tau_i$ are characterized by a (fixed) period $T_i \in \mathbb{R}^+$ and an (absolute) activation jitter $AJ_i \in \mathbb{R}^+ \cup \{0\}$, where $AJ_i < T_i$. Moreover, $\tau_i$ is characterized by a best-case computation time $BC_i \in \mathbb{R}^+$, a worst-case computation time $WC_i \in \mathbb{R}^+$, where $BC_i \leq WC_i$, a phasing $\phi_i \in \mathbb{R}$, a (relative) worst-case deadline $WD_i \in \mathbb{R}^+$, and a (relative) best-case deadline $BD_i \in \mathbb{R}^+ \cup \{0\}$, where $BD_i \leq WD_i$. The set of phasings $\phi_i$ is termed the phasing $\phi$ of the task set $T$. The deadlines $BD_i$ and $WD_i$ are relative to the activations.

Figure 1. Basic model for a periodic task $\tau_i$ with (absolute) activation jitter $AJ_i$.

Note that the activations of $\tau_i$ do not necessarily take place strictly periodically with period $T_i$, but somewhere in an interval of length $AJ_i$ that is repeated with period $T_i$. The activation times $a_{ik}$ of $\tau_i$ satisfy $\sup_{\ell}(a_{ik}(\phi_i) - a_{ik}(\phi_i) - (k - \ell)T_i) \leq AJ_i$, where $\phi_i$ denotes the start of the interval in which job zero is activated, i.e. $\phi_i + kT_i \leq a_{ik} \leq \phi_i + kT_i + AJ_i$. A task with activation jitter equal to zero is termed a strictly periodic task.

The active interval of job $t_{ik}$ is defined as the time span between the activation time $a_{ik}$ of that job and its finalization time $f_{ik}$, i.e. $[a_{ik}, f_{ik})$. The response time $R_{ik}$ of job $t_{ik}$ is defined as the length of its active interval, i.e. $R_{ik} = f_{ik} - a_{ik}$.

Figure 1 illustrates the above basic notions for an example job of a periodic task $\tau_i$.

We assume that we do not have control over the phasing $\phi$, so we assume that any arbitrary phasing may occur. We also assume other standard basic assumptions [13], i.e. tasks are ready to run upon their activation and do no suspend themselves, tasks will be preempted instantaneously when a higher priority task becomes ready to run, a job of task $\tau_i$ does not start before its previous job is completed, and the overhead of context switching and task scheduling is ignored. Finally, we assume that the deadlines are hard, i.e. each job of a task must be completed after its best-case deadline and before its worst-case deadline. Hence, a set $T$ of $n$ tasks can be scheduled if and only if

$$BD_i \leq R_{ik} \leq WD_i$$

for all $i = 1, \ldots, n$ and all $k \in \mathbb{Z}$.

For notational convenience, we assume that the tasks are given in order of decreasing priority, i.e. task $\tau_1$ has highest priority and task $\tau_n$ has lowest priority.

3. Existing best-case response time analysis

The best-case response time $BR_i$ of a task $\tau_i$ is the smallest (relative) response time of any of its jobs, i.e.

$$BR_i \overset{\text{def}}{=} \inf_{\phi \in \mathbb{R}} R_{ik}(\phi).$$

For worst-case deadlines at most equal to periods minus activation jitter, i.e. $WD_i \leq T_i - AJ_i$, the best-case response
time $BR_i$ is given by the largest $x \in \mathbb{R}^+$ that satisfies
\[ x = BC_i + \sum_{j<i} \left( \left( \frac{x - AJ_j}{T_j} \right) - 1 \right)^+ BC_j. \] (3)

Here, the notation $w^+$ stands for $\max(w, 0)$, which is used to indicate that the number of preemptions of tasks with a higher priority than $\tau_i$ can not become negative. To calculate $BR_i$, we can use an iterative procedure based on recurrence relationships, starting with an upper bound, e.g. the worst-case response time $WR_i$ of task $\tau_i$.

As described and illustrated in [18], the largest solution of (3) is a lower bound for worst-case deadlines larger than periods minus activation jitter, i.e. $WD_i > T_i - AJ_i$. For $T_i - AJ_i \geq WR_i$, we know that a job of task $\tau_i$ can never delay a next job, and the existing best-case response time analysis therefore remains exact.

4. A conjecture for arbitrary deadlines

When the worst-case relative deadline $WD_i$ of a task $\tau_i$ is larger than its period $T_i$ minus its activation jitter $AJ_i$, the execution of a job of $\tau_i$ may be delayed by the previous job. The longest interval of time in which jobs of a task can delay subsequent jobs is the worst-case length $WL_i$ of a so-called level-$i$ active period [2], which is found for the smallest $x \in \mathbb{R}^+$ that satisfies the following equation
\[ x = \sum_{j<i} \left[ \frac{x + AJ_j}{T_j} \right] WC_j. \] (4)

Such a smallest value exists when either (i) the utilization factor $U^T$ is smaller than 1 or (ii) $U^T$ is equal to 1, the activation jitter of all tasks of $T$ are equal to zero, and the least common multiple of all tasks of $T$ exists [2]. To calculate $WL_i$, we can use an iterative procedure based on recurrence relationships. The maximum number $w \ell_i$ of jobs of task $\tau_i$ in a level-$i$ active period is given by
\[ w \ell_i = \left\lfloor \frac{WL_i + AJ_i}{T_i} \right\rfloor. \] (5)

For best-case response time analysis of tasks under FPPS, we only need to consider the last job of a task $\tau_i$ in a level-$i$ active period, because that job is the only job in the active period with a response time at most equal to $T_i$ [2]. We now determine the best-case response time of a task $\tau_i$ by reusing (3) for $w \ell_i$ fictive tasks $\tau_i'$ with best-case computation times $(k + 1) \cdot BC_i$, where $0 \leq k < w \ell_i$.

**Conjecture 1** The best-case response time $BR_i$ of task $\tau_i$ with $T_i - AJ_i < WD_i$ is given by
\[ BR_i = \max_{0 \leq k < w \ell_i} \left( BR_i((k + 1) \cdot BC_i) - \begin{cases} 0 & k = 0 \\ kT_i + AJ_i & k > 0 \end{cases} \right), \] (6)

where $w \ell_i$ is the worst-case number of jobs of $\tau_i$ in a level-$i$ active period, and $BR_i((k + 1) \cdot BC_i)$ is the best-case response time of a fictive task $\tau_i'$ with a best-case computation time $BC_i' = (k + 1)BC_i$, a period equal to its worst-case deadline, i.e. $T_i' = WD_i$, and a worst-case deadline $WD_i'$ equal to
\[ WD_i' = WD_i + \begin{cases} 0 & k = 0 \\ kT_i - AJ_i & k > 0 \end{cases}, \] (7)

and a best-case deadline $BD_i'$ equal to $BD_i + k \cdot BC_i$. We can start the calculation with $k = w \ell_i - 1$ and use $WL_i$ as initial value for the iterative procedure to determine $BR_i((w \ell_i \cdot BC_i))$. For next steps, we can use the previously found $BR_i'$ value as initial value, obviating the need to determine $WR_i'$ for each fictive task $\tau_i'$. Note that for $w \ell_i = 1$, (6) becomes equal to the solution of (3). Hence, the conjecture therefore applies for tasks with arbitrary deadlines.

5. An example

For illustration purposes, we use an example task set $T_1$ with characteristics given in Table 1, and determine the best-case response time $BR_3$ of task $\tau_3$. In this example, best-case computation times are equal to worst-case computation times. The processor utilization $U^T_1 = \frac{17}{20} < 1$, hence the smallest value of (4) exists for all three tasks of $T_1$. The worst-case length $WL_3$ of the level-3 active period is equal to 20, and we therefore find $w \ell_1 = \left\lfloor \frac{WL_3 + AJ_1}{T_1} \right\rfloor = \left\lfloor \frac{20.6}{4} \right\rfloor = 3$.

Using Conjecture 1, we get $BR_3 = \max(17 - (14 + 0.6), 9 - (7 + 0.6), 2) = \max(2.4, 1.4, 2) = 2.4$. A time-line for $T_1$ with a best-case response time $BR_3 = 2.4$ for task $\tau_3$ is shown in Figure 2.

Using (3) of the existing analysis for this example yields a value $BR_3 = 2$, which is pessimistic, i.e. too small. Hence, our novel analysis for best-case response times can improve end-to-end response time analysis in distributed systems [3, 9, 10, 17, 18].

6. Conclusion

In this paper, we presented a conjecture for exact best-case response time analysis for periodically released, independent real-time tasks with arbitrary deadlines that are
ever timing constraints include scheduled using FPPS, and illustrated the analysis by means of an example. Apart from having a value on its own whenever timing constraints include lower bounds on response times of a system to events, our novel analysis allows for an improvement of existing end-to-end response time analysis in distributed systems. A formal proof of our conjecture is currently under study.

References