Journal of Systems Architecture 65 (2016) 59–63

Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

New response time analysis for global EDF on a multiprocessor platform

Jinkyu Lee

Department of Computer Science and Engineering, Sungkyunkwan University (SKKU), Republic of Korea

ARTICLE INFO

Article history:
Received 7 December 2015
Revised 6 March 2016
Accepted 13 April 2016
Available online 16 April 2016

Keywords:
Response Time Analysis
Global EDF (Earliest Deadline First)
Real-time system
Multiprocessor platform

ABSTRACT

Time-predictability is the most important requirement for a real-time system, and researchers have therefore paid attention to the duration between the arrival and completion of a real-time task, called *response time*. RTA (Response Time Analysis) studies, however, rely on the same technique, yielding room for further improvement, especially regarding multiprocessor platforms. For this paper, we investigated the properties of an existing utilization-based schedulability analysis for global EDF (Earliest Deadline First) on a multiprocessor platform, and developed a new RTA technique based on the corresponding properties, which calculates the response times of tasks in task sets deemed schedulable by the existing analysis. We demonstrated through simulations that our proposed RTA technique not only calculates response times that are less pessimistic than those of the existing approach, but also successfully derives response times that cannot be obtained by the existing approach.

1. Introduction

In real-time systems, it is important, even in the worst-case scenarios, to make the systems predictable. The *response time analysis* (RTA), which calculates an upper bound on the time duration between the release time and the completion time of a task, has therefore been widely studied—especially regarding real-time control applications for which the input-output delay and jitter are critical. As multi-core architectures become popular, the RTA technique that accounts for the interference of higher-priority tasks on a uniprocessor platform [1] has been extended to global scheduling algorithms on a multiprocessor platform [2–5]. However, most (if not all) of the existing RTAs for global scheduling algorithms are only sufficient and rely on the same technique used in [2], meaning it is worthwhile to develop of a new RTA technique that can find tighter upper bounds on the response time.

In this paper, we develop a new RTA technique for global EDF (Earliest Deadline First) [6]; for this purpose, we revisit a utilization-based schedulability analysis for global EDF [7] called GFB, and then develop a new RTA technique based on GFB, which calculates the response times of tasks in task sets deemed schedulable by GFB. We demonstrate via simulations that our proposed RTA can result in smaller response times for some tasks, when compared to those derived by the existing RTA [2] and its improved version [5]. We also show that our proposed RTA technique successfully calculates the response times of some tasks that cannot be obtained by the existing RTA [2] and [5].

System model. In this paper we focus on a sporadic task model [8]. In this model, we specify a task τ_i in a task set τ as (T_i, C_i), where T_i is the minimum separation (as well as the relative deadline), and C_i is the worst-case execution time requirement when τ_i is exclusively executed on a unit-capacity processor. A task τ_i invokes a series of jobs, each separated from its predecessor by at least T_i time units, whereby each job of τ_i should finish its execution within T_i time units after its release. We consider a platform with m identical unit-capacity processors, and assume that a single job of a task cannot be executed in parallel. Our target scheduler is global EDF in which m jobs with the earliest deadlines are chosen for execution.

2. New Response Time Analysis for Global EDF

In this section, we first recapitulate an existing utilization-based schedulability analysis for global EDF called GFB, as well as its properties [7]. Then, we develop a new RTA technique, based on GFB with the properties.

2.1. *GFB schedulability analysis with its properties*

In GFB, the amount of execution by global EDF on a platform with m identical unit-capacity processors is compared with that by (ideal) fluid execution. The former and latter are formally expressed in the following definitions.

E-mail address: jinkyu.lee@skku.edu

http://dx.doi.org/10.1016/j.sysarc.2016.04.003
1383-7621/© 2016 Elsevier B.V. All rights reserved.

Definition 1. Let $\mathcal{W}_i(t_1, t_2, \tau, m)$ denote the amount of execution that is performed by jobs invoked by $\tau_i \in \tau$ in $[t_1, t_2)$, when τ is scheduled by global EDF on a platform with m identical unit-capacity processors.

Definition 2. Let $\mathcal{L}_i(t_1, t_2, \tau_i)$ denote the amount of execution performed by jobs invoked by τ_i $[t_1, t_2)$, when τ_i is exclusively scheduled on a single processor with $\frac{C_i}{T_i}$-capacity.

Then, we present the schedulability analysis of GFB and its related properties by using the above definitions in the following lemma.

Lemma 1 GFB [7]**.** *Suppose the following condition holds for a task set τ:*

$$\sum_{\tau_i \in \tau} \frac{C_i}{T_i} \le m - (m-1) \cdot \max_{\tau_i \in \tau} \frac{C_i}{T_i}. \tag{1}$$

Then, the following (i) and (ii) hold, when τ is scheduled by global EDF on a platform with m identical unit-capacity processors.

(i) There is no deadline miss for τ; and
(ii) The following condition holds for all $t \ge 0$:

$$\sum_{\tau_i \in \tau} \mathcal{W}_i(0, t, \tau, m) \ge \sum_{\tau_i \in \tau} \mathcal{L}_i(0, t, \tau_i). \tag{2}$$

Note that Eq. (2) is presented Lemma 1 in [7] with different notations.

Proof. Although the proof is given in [7], we briefly prove the lemma for completeness.

Suppose that Eq. (2) is violated, and t_0 denotes the time instant when the inequality violated at the first time. Then, there exists a job of a task τ_k that satisfies

$$\mathcal{W}_k(a, t_0, \tau, m) < \mathcal{L}_k(a, t_0, \tau_k), \text{ and} \tag{3}$$

$$\sum_{\tau_i \in \tau} \mathcal{W}_i(a, t_0, \tau, m) < \sum_{\tau_i \in \tau} \mathcal{L}_i(a, t_0, \tau_i), \tag{4}$$

where a is the release time of the job of τ_k.

Let x and y denote the amount of time when all m processors are busy and at least one processor is idle in $[a, t_0)$, respectively. Then, by definition, the amount of execution by τ in $[a, t_0)$ is at least $m \cdot x + y$, and therefore, from Eq. (4), the following inequality holds:

$$m \cdot x + y < \sum_{\tau_i \in \tau} \frac{C_i}{T_i} \cdot (x + y). \tag{5}$$

Also, since the job of τ_k executes at least y amount of time in $[a, t_0)$, the following inequality holds from Eq. (3):

$$y < \max_{\tau_i \in \tau} \frac{C_i}{T_i} \cdot (x + y). \tag{6}$$

If we add Eq. (5) to $(m-1)\cdot$ Eq. (6), we can conclude the contradiction of Eq. (1). □

In the next subsection, we develop a new RTA technique using the properties of Lemma 1.

2.2. New Response Time Analysis

To develop a new RTA for global EDF using Lemma 1, we first derive the following properties regarding $\mathcal{W}_i(\cdot)$ and $\mathcal{L}_i(\cdot)$ that will be used for the new RTA:

P1. Suppose there is no deadline miss until t_0 for τ. If $t\ (<t_0)$ belongs to an interval between the deadline of a job of τ_i and the release time of the next job of τ_i, $\mathcal{W}_i(0, t, \tau, m) = \mathcal{L}_i(0, t, \tau_i)$ holds; and

P2. $\mathcal{L}_i(t_1, t_2, \tau_i) \le (t_2 - t_1) \cdot \frac{C_i}{T_i}$ holds for all of $t_2 \ge t_1 \ge 0$.

Both properties trivially hold. For P1, considering there is no deadline miss, $\mathcal{W}_i(0, t, \tau, m)$ is the same as $\mathcal{L}_i(0, t, \tau_i)$, as long as t does not belong to the execution window (an interval between the release time and deadline) of any job of τ_i. For P2, since the processor has a $\frac{C_i}{T_i}$-capacity, the amount of execution cannot exceed the value of $\frac{C_i}{T_i}$ multiplied by the interval length. By using the GFB properties, we develop a new RTA for global EDF in the following theorem:

Theorem 1. *Suppose a task set τ is scheduled by global EDF on a platform with m identical unit-capacity processors. If Eq. (1) holds, the response time of $\tau_k \in \tau$ is upper-bounded by R_k, where the following applies:*

$$R_k = T_k \cdot \frac{\sum_{\tau_i \in \tau - \{\tau_k\}} \frac{C_i}{T_i}}{m} + C_k. \tag{7}$$

Proof. By Lemma 1, if Eq. (1) holds, (i) and (ii) also hold. We now derive Eq. (7) using (i) and (ii), and then P1 and P2.

Let x denote the release time of a job of τ_k (called J_k). We now calculate the upper-bound of the sum of J_k's higher-priority execution of other tasks $\tau_i \in \tau - \{\tau_k\}$ in $[x, x + T_k)$, i.e., a time interval between the release time and deadline of J_k.

Let $x + l_i$ denote the earliest deadline of a job invoked by τ_i after x (i.e., $l_i > 0$). We considered two cases: $l_i > T_k$ and $l_i \le T_k$. Since $x + T_k$ is the deadline of J_k and the scheduler is global EDF, the amount of J_k's higher-priority execution by jobs of τ_i in $[x, x + T_k)$ is zero if $l_i > T_k$. We now investigate the case of $l_i \le T_k$.

We considered the following two intervals: $[x, x + l_i)$ and $[x + l_i, x + T_k)$. All of the jobs of τ_i executed in $[x, x + l_i)$ have a higher priority than J_k, and the amount of the higher-priority execution is $\mathcal{W}_i(0, x + l_i, \tau, m)$ - $\mathcal{W}_i(0, x, \tau, m)$, which is the same as $\mathcal{L}_i(0, x + l_i, \tau_i) - \mathcal{W}_i(0, x, \tau, m)$, by P1. In $[x + l_i, x + T_k)$, there are at most $\lfloor \frac{T_k - l_i}{T_i} \rfloor$ jobs of τ_i, which have a higher-priority than J_k, i.e., jobs of τ_i with release times and deadlines within $[x + l_i, x + T_k)$. Then, the amount of higher-priority execution is upper-bounded by $\lfloor \frac{T_k - l_i}{T_i} \rfloor \cdot C_i$; therefore, the total amount of execution by the jobs of a higher-priority than J_k in $[x, x + T_k)$ is upper-bounded according to the following:

$$\begin{aligned}
&\sum_{\tau_i \in \tau - \{\tau_k\}} \Big(\mathcal{L}_i(0, x + l_i, \tau_i) - \mathcal{W}_i(0, x, \tau, m) \\
&\quad + \left\lfloor \frac{T_k - l_i}{T_i} \right\rfloor \cdot C_i \Big) \\
&= - \sum_{\tau_i \in \tau - \{\tau_k\}} \mathcal{W}_i(0, x, \tau, m) \\
&\quad + \sum_{\tau_i \in \tau - \{\tau_k\}} \Big(\mathcal{L}_i(0, x + l_i, \tau_i) + \left\lfloor \frac{T_k - l_i}{T_i} \right\rfloor \cdot C_i \Big).
\end{aligned} \tag{8}$$

Applying P1, i.e., $\mathcal{W}_k(0, x, \tau, m) = \mathcal{L}_k(0, x, \tau_k)$, Eq. (2) can be changed as follows:

$$\begin{aligned}
&\sum_{\tau_i \in \tau - \{\tau_k\}} \mathcal{W}_i(0, x, \tau, m) \ge \sum_{\tau_i \in \tau - \{\tau_k\}} \mathcal{L}_i(0, x, \tau_i) \\
&\iff - \sum_{\tau_i \in \tau - \{\tau_k\}} \mathcal{W}_i(0, x, \tau, m) \le - \sum_{\tau_i \in \tau - \{\tau_k\}} \mathcal{L}_i(0, x, \tau_i).
\end{aligned} \tag{9}$$

Then, we re-arrange Eq. (8), resulting in the following upper-bound:

The RHS of Eq. (8)

$$\leq -\sum_{\tau_i \in \tau - \{\tau_k\}} \mathcal{L}_i(0, x, \tau_i)$$

$$+ \sum_{\tau_i \in \tau - \{\tau_k\}} \left(\mathcal{L}_i(0, x + l_i, \tau_i) + \left\lfloor \frac{T_k - l_i}{T_i} \right\rfloor \cdot C_i \right)$$

$$= \sum_{\tau_i \in \tau - \{\tau_k\}} \left(\mathcal{L}_i(0, x + l_i, \tau_i) - \mathcal{L}_i(0, x, \tau_i) + \left\lfloor \frac{T_k - l_i}{T_i} \right\rfloor \cdot C_i \right)$$

$$= \sum_{\tau_i \in \tau - \{\tau_k\}} \left(\mathcal{L}_i(x, x + l_i, \tau_i) + \left\lfloor \frac{T_k - l_i}{T_i} \right\rfloor \cdot C_i \right).$$

After applying P2,

$$\leq \sum_{\tau_i \in \tau - \{\tau_k\}} \left(l_i \cdot \frac{C_i}{T_i} + \left(\frac{T_k - l_i}{T_i} \right) \cdot C_i \right)$$

$$= T_k \cdot \sum_{\tau_i \in \tau - \{\tau_k\}} \frac{C_i}{T_i}. \tag{10}$$

As a result, $T_k \cdot \sum_{\tau_i \in \tau - \{\tau_k\}} \frac{C_i}{T_i}$ amount is the maximum of the higher-priority execution in $[x, x + T_k)$. Since m higher-priority executions are required to prevent the execution of J_k in any interval, J_k can be blocked by a maximum of $\frac{T_k \cdot \sum_{\tau_i \in \tau - \{\tau_k\}} \frac{C_i}{T_i}}{m}$ time units, thereby proving the theorem. □

3. Evaluation

In this section, we compare our RTA with the existing RTA technique in [2,5], in terms of response time and time-complexity.

To compare the average performance of the response time, we need to generate task sets by modifying the popular technique used in [9–11]. The following two input parameters are employed: (i) number of processors m (two or four) and (ii) individual task-utilization (C_i/T_i) distribution (i.e., the five bimodal and five exponential distributions in [11]). For each task, T_i is uniformly distributed in $[1, T_{max} = 1000]$, and C_i is chosen based on the individual task utilization distribution. For each combination of (i) and (ii), we repeat the following procedure from [10] in order to generate 10,000 task sets that results in 100,000 task sets for any given m.

1. Initially, a set of $m + 1$ tasks is generated.
2. To exclude unschedulable task sets, we check whether the generated task set satisfies a necessary and sufficient feasibility condition according to [12].
3. If the check of Step 2 fails, we discard the generated task set and return to Step 1; otherwise, we include this set for evaluation. This set then serves as a basis for the next new set, followed by the creation of a new set through the addition of a new task to this old set and a return to Step 2.

Among all of the generated task sets, we focused on the task sets that satisfy Eq. (1), and calculated the individual tasks' response times by using our RTA technique in Eq. (7) and the existing RTA technique in [2]; these equations are annotated by Ours and BeCi, respectively. For every individual task's response time, we calculated and sorted the ratio of BeCi to Ours, and the cumulative percentages of the ratios are plotted in Fig. 1.

As shown in Fig. 1(a) with $m = 2$, while BeCi results in smaller response times for 65.9% tasks (between 0 and 65.9), the same occurs for Ours for 34.1% tasks (between 65.9 and 100.0); moreover, for 15.6% tasks (between 84.4 and 100.0), BeCi cannot compute response times (because BeCi concludes that at least one task in each task set has its response time larger than its relative deadline), whereas Ours successfully calculates response times. The following example shows superiority of Ours in terms of response time calculation.

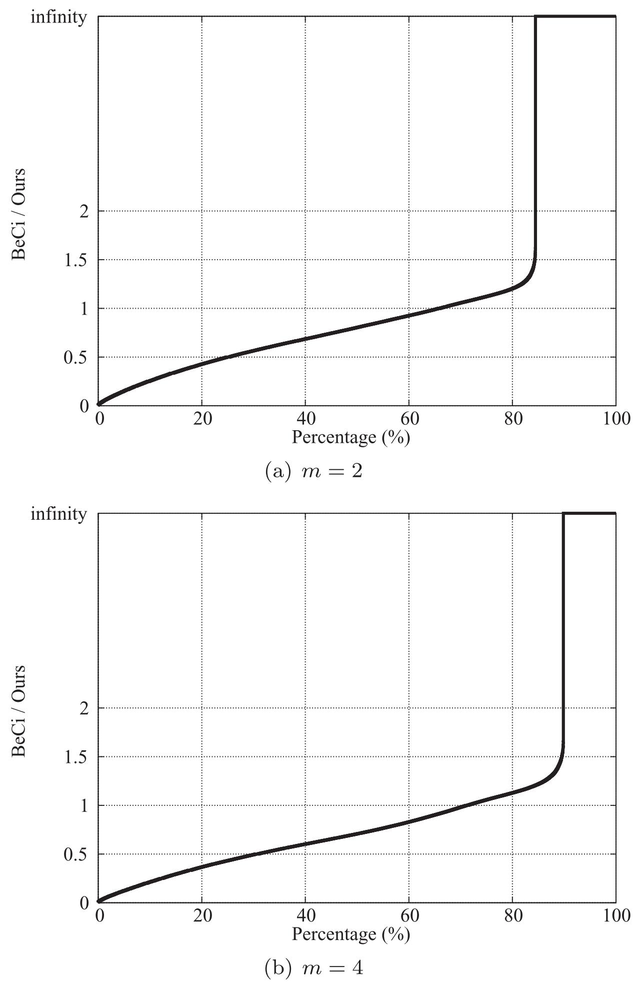

Fig. 1. Cumulative percentage of the ratio of BeCito Ours(BeCi/Ours).

Example 1. Consider a task set $\tau = \{\tau_1(T_1 = 100, C_1 = 40), \tau_2(80, 40), \tau_3(60, 30)\}$ scheduled on two processors. While the response times of tasks in τ calculated by BeCi are 100, 80, and larger than 60 (unscheduable) for τ_1, τ_2, and τ_3, respectively, those calculated by Ours are 90, 76, and 57. Here, BeCi calculates $R_1 = 100$, $R_2 = 80$, and $R_3 > 60$ in the first iteration; since there is no slack update (i.e., no task satisfies $D_i - R_i > 0$), BeCi finishes the response time calculation without any further iterations.

Likewise, when $m = 4$ in Fig. 1(b), we observed similar behaviors, whereby Ours reduces the response times of 28.7% tasks and calculates the response times, which cannot be computed by BeCi, of 10.2% tasks.

Regarding the overhead, Ours requires only $O(|\tau|)$ time-complexity for the response time of each task, where $|\tau|$ is the number of tasks in τ. However, the time-complexity of BeCi in [2] has been observed as pseudo-polynomial.

We also compare response times calculated by BeCi with those calculated by the most recent work [5] (published just before this paper's submission), which is denoted by SuLi. SuLi incorporates the technique that yields less pessimistic interference calculation presented in [13], to the response time calculation framework of BeCi in [2]. It is known that SuLi performs better than

BeCi. With the same task sets, Ours yields smaller response times than SuLi for 2.7% and 3.8% tasks for $m = 2$ and $m = 4$, respectively. Note that the time-complexity of SuLi is much higher than BeCi (pseudo-polynomial), while Ours requires only $O(|\tau|)$ time-complexity.

In summary, Ours, which requires a lower time-complexity, not only reduces the pessimism of the calculation of the response times for a number of tasks, but also calculates a number of additional response times that cannot be obtained by BeCi and SuLi. Note that the response times derived by both Ours and SuLi (and BeCi) are safe upper-bounds, and we can therefore choose the minimum among the all to obtain smaller response times.

4. Conclusion

In this paper, we derived a new RTA technique for global EDF on a multiprocessor platform. We also demonstrated via simulations that our RTA technique not only results in smaller response times, but also calculates the response times of additional tasks that cannot be obtained by using the existing approach.

In order to derive the new RTA technique in Theorem 1, we used the properties of (i) and (ii). In the future, we intend to find scheduling algorithms that satisfy some properties similar to (i) and (ii), and then develop further new RTA techniques. Another direction of future work is to develop ways to apply the proposed scheme to other existing EDF schedulability tests [14].

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2014R1A1A1035827). This work was also supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIP) (No.R0190-15-2071, Open PNP platform for diversity of autonomous vehicle based on cloud map).

References

[1] N. Audsley, A. Burns, M. Richardson, A. Wellings, Hard real-time scheduling: the deadline-monotonic approach, in: Proceedings of the IEEE Workshop on Real-Time Operating Systems and Software, 1991, pp. 133–137.

[2] M. Bertogna, M. Cirinei, Response-time analysis for globally scheduled symmetric multiprocessor platforms, in: Proceedings of IEEE Real-Time Systems Symposium, 2007, pp. 149–160.

[3] N. Guan, M. Sitgge, W. Yi, G. Yu, New response time bounds for fixed priority multiprocessor scheduling, in: Proceedings of IEEE Real-Time Systems Symposium, 2009, pp. 387–397.

[4] R. Davis, A. Burns, Priority assignment for global fixed priority pre-emptive scheduling in multiprocessor real-time systems, in: Proceedings of IEEE Real–Time Systems Symposium, 2009, pp. 398–409.

[5] Y. Sun, G. Lipari, Response time analysis with limited carry-in for global earliest deadline first scehduling, in: Proceedings of IEEE Real-Time Systems Symposium, 2015, pp. 130–140.

[6] C. Liu, J. Layland, Scheduling algorithms for multi-programming in a hard-real-time environment, J. ACM 20 (1) (1973) 46–61.

[7] J. Goossens, S. Funk, S. Baruah, Priority-driven scheduling of periodic task systems on multiprocessors, Real-Time Syst. 25 (2–3) (2003) 187–205.

[8] A. Mok, Fundamental Design Problems of Distributed Systems for the Hard-Real-Time Environment, Massachusetts Institute of Technology, 1983 Ph.D. thesis.

[9] T. Baker, An analysis of EDF schedulability on a multiprocessor, IEEE Trans. Parallel Distrib. Syst. 16 (8) (2005) 760–768.

[10] M. Bertogna, M. Cirinei, G. Lipari, Schedulability analysis of global scheduling algorithms on multiprocessor platforms, IEEE Trans. Parallel Distrib. Syst 20 (2009) 553–566.

[11] J. Lee, A. Easwaran, I. Shin, Maximizing contention-free executions in multiprocessor scheduling, in: Proceedings of IEEE Real-Time Technology and Applications Symposium, 2011, pp. 235–244.

[12] S. Baruah, N.K. Cohen, C.G. Plaxton, D.A. Varvel, Proportionate progress: a notion of fairness in resource allocation, Algorithmica 15 (6) (1996) 600–625.

[13] S. Baruah, Techniques for multiprocessor global schedulability analysis, in: Proceedings of IEEE Real-Time Systems Symposium, 2007, pp. 119–128.

[14] M. Bertogna, S. Baruah, Tests for global EDF schedulability analysis, J. Syst. Archit. 57 (5) (2011) 487–497.

Jinkyu Lee is an assistant professor in Department of Computer Science and Engineering, Sungkyunkwan University (SKKU), Republic of Korea, where he joined in 2014. He received the BS, MS, and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology (KAIST), Republic of Korea, in 2004, 2006, and 2011, respectively. He has been a research fellow/visiting scholar in the Department of Electrical Engineering and Computer Science, University of Michigan until 2014. His research interests include system design and analysis with timing guarantees, QoS support, and resource management in real-time embedded systems and cyber-physical systems. He won the best student paper award from the 17th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS) in 2011, and the Best Paper Award from the 33rd IEEE Real-Time Systems Symposium (RTSS) in 2012.