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

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:

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

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: