On utilization bounds for a periodic resource under rate monotonic scheduling

André M. van Renssen, Stefan J. Geuns, Joost P.H.M. Hausmans, Wouter Poncin, and Reinder J. Bril

Technische Universiteit Eindhoven (TU/e), Department of Mathematics and Computer Science,
Den Dolech 2, 5600 AZ Eindhoven, The Netherlands
a.m.v.renssen@student.tue.nl, r.j.bril@tue.nl

Abstract

This paper revisits utilization bounds for a periodic resource under the rate monotonic (RM) scheduling algorithm. We show that the existing utilization bound, as presented in [8, 9], is optimistic. We subsequently show that by viewing the unavailability *of the periodic resource as a deferrable server at highest priority, existing utilization bounds for systems with a deferrable server [3, 11] can be reused. Moreover, using this view, the utilization bound presented in [7] for hierarchical fixed-priority scheduling turns out to be similar to the bound in [3].*

1. Introduction

Today, fixed-priority pre-emptive scheduling (FPPS) is a de-facto standard in industry for scheduling systems with real-time constraints. A major shortcoming of FPPS, however, is that temporary or permanent faults occurring in one application can hamper the execution of other applications. To resolve this shortcoming, the notion of *resource reservation* [6] has been proposed. Resource reservation provides *isolation* between applications, effectively protecting an application against other, malfunctioning applications.

In a basic setting of a real-time system, we consider a set of independent applications, where each application consists of a set of independent, periodically released, hard real-time tasks that are executed on a shared resource. We assume two-level hierarchical scheduling, where a *global* scheduler determines which application should be provided the resource and a *local* scheduler determines which of the chosen application's tasks should execute. Although each application could have a dedicated scheduler, we assume FPPS for every application. For temporal protection, each application is associated a dedicated reservation. We assume a *periodic resource model* [8] for reservations.

In this paper, we consider least upper bounds for schedulability of an application given a periodic resource, where the local scheduler uses the rate monotonic (RM) scheduling algorithm. We show that the existing utilization bound, as presented in [8, 9], is optimistic. We subsequently show that by viewing the *unavailability* of the periodic resource as a deferrable server at highest priority, we can reuse existing utilization bounds for systems with a deferrable server [3, 11]. We briefly discuss (*i*) two errors identified in the latter utilization bounds, (*ii*) the similarity between the bounds in [7] for hierarchical FPPS and the bound in [3], and (*iii*) a novel utilization bound as presented in [10].

This paper is organized as follows. In Section 2, we briefly recapitulate the system model described in [8] and the utilization bound for the RM algorithm. An example refuting that bound is presented in Section 3. In Section 4, we show how to reuse existing utilization bounds for systems with a deferrable server at highest priority. We discuss utilization bounds in Section 5 and conclude the paper in Section 6.

2. Recapitulation of existing results

This section briefly recapitulates the system model and the utilization bound for the RM algorithm of [8].

We consider a *workload model* W, which describes the applications, a *periodic resource model* Γ, which describes the available resources, and a shared resource, i.e. a single processor. For the workload model, we assume the periodic task model of Liu and Layland [4]. Hence, we assume n periodically released, independent tasks τ_1, τ_2, …, τ_n with unique, fixed priorities, that do not suspend themselves, and have arbitrary phasing. Each task τ_i is characterized by (p_i, e_i), where p_i is the period and e_i is the computation time. We assume that the tasks are given in order of decreasing priority, i.e. task τ_1 has highest priority and task τ_n has lowest priority. We use the rate monotonic (RM) scheduling algorithm to schedule the tasks, i.e. we assume $p_1 \leq p_2 \leq \cdots \leq p_n$.

A periodic resource model $\Gamma(\Pi, \Theta)$ characterizes a partitioned resource that guarantees allocations of Θ time units

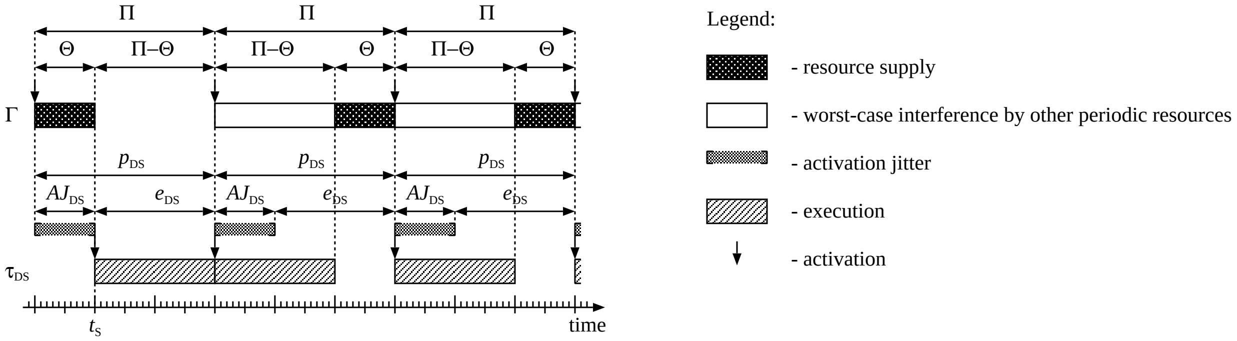

Figure 1. A situation with a worst-case (i.e. minimum) resource supply of a periodic resource Γ in an interval starting at time t_S, and a task τ_{DS} modeling the *unavailability* of the periodic resource Γ.

every Π time units, where a resource period Π is a positive integer and a resource allocation time Θ is a real number in $(0, \Pi]$. Figure 1 illustrates a situation with a worst-case (i.e. a minimum) resource supply of a periodic resource Γ in an interval starting at time t_s. From this figure, we derive that the *longest* interval without any resource supply from Γ has a length of $2(\Pi - \Theta)$.

Given a periodic resource Γ, the *utilization bound* $UB_\Gamma(RM)$ of the RM scheduling algorithm is defined as a number such that a periodic workload set W is schedulable if

$$\sum_{\tau_i \in W} \frac{e_i}{p_i} \leq UB_\Gamma(RM). \qquad (1)$$

The following theorem from [8] provides a utilization bound for the RM scheduling algorithm.

Theorem 1 (Utilization Bound for RM Algorithm ([8])) *Given a periodic resource $\Gamma(\Pi, \Theta)$, a utilization bound $UB_\Gamma(RM)$ of the RM scheduling algorithm for a set of m periodic workloads is*

$$UB_\Gamma(RM) = \frac{\Theta}{\Pi}\left(m(\sqrt[m]{2}-1) - \frac{\sqrt[m]{2}(\Pi - \Theta)}{p^*}\right), \qquad (2)$$

where p^* is the shortest period of W.

3. Utilization bound from [8] refuted

Consider a periodic resource $\Gamma(\Pi, \Theta)$ and a periodic workload set W consisting of 2 tasks τ_1 and τ_2, which are characterized by $(100, 1)$ and $(150, 1)$, respectively. Hence, the processor utilization U_W of W is given by $U_W = \frac{e_1}{p_1} + \frac{e_2}{p_2} = \frac{1}{100} + \frac{1}{150} = \frac{1}{60}$. Let $2(\Pi - \Theta) = p_1$, i.e. the worst-case length $2(\Pi - \Theta)$ of an interval of time without any resource supply from Γ is equal to the period p_1 of task τ_1. Hence, task τ_1 is *not* schedulable. According to Theorem 1, the workload W is schedulable when $U_W \leq UB_\Gamma(RM)$, i.e. for

$$\frac{1}{60} \leq \frac{\Pi - 50}{\Pi}\left(2(\sqrt[2]{2}-1) - \frac{\sqrt[2]{2}(50)}{100}\right).$$

We can rewrite this latter relation to

$$\Pi \geq \frac{50(\frac{3}{2}\sqrt{2}-2)}{(\frac{3}{2}\sqrt{2}-2) - \frac{1}{60}}.$$

The right-hand side of the relation is approximately 58. According to Theorem 1, the workload W is therefore schedulable for $\Pi = 60$ and $\Theta = 10$, which is obviously wrong; see also Figure 2.

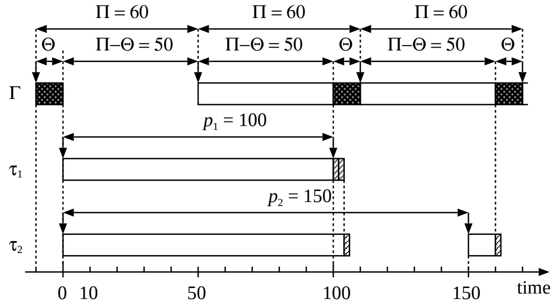

Figure 2. A timeline with a deadline miss for task τ_1 at time $t = 100$.

4. Reusing existing utilization bounds

In this section, we show how to reuse existing utilization bounds for systems with a deferrable server at highest priority for a periodic resource. To that end, we first show that the *unavailability* of a periodic resource can be modeled as a Deferrable Server (DS) task at highest priority. Next, we show how to apply existing results.

4.1. Unavailability of the periodic resource

Figure 1 shows that for worst-case analysis purposes, the *unavailability* of the periodic resource Γ can be modeled as a (DS) task τ_{DS} with a (fixed) period $p_{\mathrm{DS}} = \Pi$, a (fixed) computation time $e_{\mathrm{DS}} = \Pi - \Theta$, and an activation jitter $AJ_{\mathrm{DS}} = \Theta$; see also [1]. Similar results can be found for the worst-case response time analysis of tasks with an associated sporadic server as presented in [7].

4.2. A utilization bound

In [3, 11], least upper bounds on schedulability are presented for the rate monotonic scheduling algorithm for task sets have a so-called Deferrable Server (DS) task τ_{DS} executing at highest priority and n ordinary periodic tasks $\tau_1, \tau_2, \ldots, \tau_n$. These papers make the following general assumption

$$p_{\mathrm{DS}} \leq p_1 \leq p_2 \leq \cdots \leq p_n. \qquad (3)$$

In [11], it is shown that $\frac{p_1}{p_{DS}} < 2$ and $\frac{p_n}{p_{DS}} < 2 + U_{DS}$ are necessary conditions for a task set to be a worst-case task set. Next, for $p_n \in [kp_{\mathrm{DS}} + e_{\mathrm{DS}}, kp_{\mathrm{DS}} + 2e_{\mathrm{DS}})$, where $k \geq 1$, it is shown that p_n can be increased to $p'_n = kp_{\mathrm{DS}} + 2e_{\mathrm{DS}}$. The analysis is subsequently carried out by considering three distinct cases:

1. $p_{\mathrm{DS}} \leq p_1 \leq \ldots \leq p_n < p_{\mathrm{DS}} + e_{\mathrm{DS}}$;
2. $p_{\mathrm{DS}} < p_{\mathrm{DS}} + e_{\mathrm{DS}} \leq p_1 \leq \ldots \leq p_n < 2p_{\mathrm{DS}} + e_{\mathrm{DS}}$;
3. $p_{\mathrm{DS}} \leq p_1 \leq \ldots \leq p_k < p_{\mathrm{DS}} + e_{\mathrm{DS}} < p_{\mathrm{DS}} + 2e_{\mathrm{DS}} \leq p_{k+1} \leq \cdots \leq p_n \leq 2p_{\mathrm{DS}} + e_{\mathrm{DS}}$ for some k, $1 \leq k \leq n-1$.

Note that our example satisfies the general assumption expressed by (3), i.e. $p_{\mathrm{DS}} = 60 \leq 100 = p_1 \leq p_2 = 150$, and both necessary conditions, i.e. $\frac{p_1}{p_{DS}} = \frac{100}{60} < 2$ and $\frac{p_n}{p_{DS}} = \frac{150}{60} = 2\frac{1}{2} < 2\frac{50}{60} = 2 + U_{DS}$. Moreover, because $p_2 \in [p_{\mathrm{DS}} + e_{\mathrm{DS}}, kp_{\mathrm{DS}} + 2e_{\mathrm{DS}})$, i.e. $150 \in [110, 160)$ for $k = 1$, we can increase p_2 to $p'_2 = kp_{\mathrm{DS}} + 2e_{\mathrm{DS}} = 160$. With this new value for p'_2, we can use the analysis for case 3. Because case 3 assumes $p_{\mathrm{DS}} \leq p_1$, the utilization bound for the ordinary periodic tasks are 0 (zero) for a utilization $U_{\mathrm{DS}} \geq \frac{1}{2}$ of task τ_{DS}. Hence, our example task set has a utilization larger than the least upper bound on schedulability.

5. Discussion

In this section, we briefly discuss (*i*) two errors identified in [3, 11] (*ii*) the similarity between the bounds in [7] for hierarchical FPPS and the bound in [3], and (*iii*) a novel utilization bound as presented in [10].

5.1. Derivations of bounds are error-prone

The derivation of the least upper bounds for schedulability under the rate monotonic algorithm for task sets with a so-called Deferrable Server (DS) task τ_{DS} is rather complex and therefore error-prone. We will illustrate this by two examples.

As a first example, the original bound given in Theorem 3 in [3] contains a typo in the denominator, i.e. a '2' is lacking in front of U_{DS} in (4), where we used our notation in Theorem 2.

Theorem 2 (Th. 3 in [3]) *For a set of $n+1$ fixed priority ordered tasks τ_{DS}, τ_1, τ_2, …, τ_n with a critical zone length greater than $T_{\mathrm{DS}} + C_{\mathrm{DS}}$, where τ_{DS} is the Deferrable Server, the least upper utilization bound as a function of U_{DS} is*

$$U = U_{\mathrm{DS}} + n\left[\left(\frac{U_{\mathrm{DS}}+2}{U_{\mathrm{DS}}+1}\right)^{1/n} - 1\right] \qquad (4)$$

which converges to

$$U = U_{\mathrm{DS}} + \ln\left(\frac{U_{\mathrm{DS}}+2}{2U_{\mathrm{DS}}+1}\right)^{1/n} \quad \textit{as } n+1 \rightarrow \infty \qquad (5)$$

This function has a minimum of 0.6518 when $U_{\mathrm{DS}} = 0.186$.

This typo originated during the derivation of equation (3) in that paper. The original bound (4) may therefore be optimistic[1]. Unfortunately, the same typo reappears in Theorem 7.2 in [5]. The derivation of a similar least upper bound in [2] resulted in an equation without that typo. Note that (4) specializes to the *LL*-bound for $U_{\mathrm{DS}} = 0$.

Another example is a least upper bound for the periodic tasks for case 3 (mentioned in Section 4.2) as described by equation (56) in [11]. Using our notation, that bound is given by

$$\mathrm{DS}_{\mathrm{per},\infty}(U_{\mathrm{DS}}) = \ln\left(\frac{1+U_{\mathrm{DS}}}{1+2U_{\mathrm{DS}}}(2-U_{\mathrm{DS}})\right) \text{ for } 0 \leq U_{\mathrm{DS}} \leq \tfrac{1}{2} \qquad (6)$$

Notably, $\mathrm{DS}_{\mathrm{per},\infty}(\frac{1}{2}) = \ln(\frac{9}{8}) > 0$. This is wrong, because for $U_{\mathrm{DS}} = \frac{1}{2}$ and $p_1 = p_{\mathrm{DS}}$ the largest value of e_1 is given by (see equation (7) in [11])

$$\begin{aligned} e_1 &= p_1 - 2e_{\mathrm{DS}} \\ &= \{p_1 = p_{\mathrm{DS}}\}\ p_{\mathrm{DS}} - 2e_{\mathrm{DS}} \\ &= \{\tfrac{e_{\mathrm{DS}}}{p_{\mathrm{DS}}} = \tfrac{1}{2}\}\ 2e_{\mathrm{DS}} - 2e_{\mathrm{DS}} \\ &= 0, \end{aligned}$$

and therefore $\mathrm{DS}_{\mathrm{per},\infty}(\frac{1}{2}) = 0$. The bound for case 3 is therefore optimistic.

[1] Note that (5) is correct.

5.2. Similarity between bounds in [3, 7]

We now show that the bound in [7] is similar to the bound in [3]. The following least upper bound is given in [7].

Theorem 3 (Th. 7 in [7]) *For a hierarchical reservation system, the least upper bound of the processor utilization factor for n child reserves under a parent reserve is*

$$U = n\left[\left(\frac{3-U^p}{3-2U^p}\right)^{1/n} - 1\right]. \quad (7)$$

Similar to our approach described in Section 4, we can model the *unavailability* of the parent reserve as a deferrable server task τ_{DS}. By substituting $U^p = 1 - U_{\text{DS}}$ in (7), we get

$$U = n\left[\left(\frac{U_{\text{DS}}+2}{2U_{\text{DS}}+1}\right)^{1/n} - 1\right]. \quad (8)$$

Hence, the least upper utilization bound derived in [7] for the child reserves is similar to the (corrected) utilization bound for the n tasks described in [3].

5.3. Utilization bound in [10]

In [10], Shin and Lee present a utilization bound $\text{UB}_{\Gamma,\text{RM}}(n, P_{min})$ that differs from their bound given in [8].

Theorem 4 (Th. 5.2 in [10]) *For scheduling unit $S(W,R,A)$, where $W = \{T(p_1,e_1),\ldots,T(p_n,e_n)\}$, $R = \Gamma(\Pi,\Theta)$, $A = \text{RM}$, and $p_i \geq 2\Pi - \Theta$, $1 \leq i \leq n$, its utilization bound $\text{UB}_{\Gamma,\text{RM}}(n,P_{min})$ is*

$$\text{UB}_{\Gamma,\text{RM}}(n,P_{min}) = U_\Gamma \cdot n\left[\left(\frac{2\cdot k + 2(1-U_\Gamma)}{k+2(1-U_\Gamma)}\right)^{1/n} - 1\right], \quad (9)$$

where $k = K_{\text{RM}}(P_{min}, R)$

For Theorem 4, $K_{\text{RM}}(P_{min}, \Gamma(\Pi,\Theta))$ is defined as

$$K_{\text{RM}}(P_{min},\Gamma(\Pi,\Theta)) = \max_{k\in\mathbb{Z}}\{(k+1)\Pi - \Theta < P_{min}\} \quad (10)$$

and $P_{min} = \min_{1\leq i\leq n} p_i$. Similar to the bounds in [3, 7], this bound specializes to the *LL*-bound for $U_\Gamma = 1$. Surprisingly, (9) does not specialize to a bound similar to (7) for $k = 1$. The reason why and its consequence are a topic of future work.

6. Conclusion

In this paper, we revisited utilization bounds for a periodic resource under the rate monotonic scheduling algorithm. We showed by means of an example that the existing utilization bound, as presented in [8, 9], is optimistic. We subsequently showed that by viewing the *unavailability* of the periodic resource as a deferrable server at highest priority, existing utilization bounds for systems with a deferrable server [3, 11] can be reused. Unfortunately, these earlier results also contain errors, as illustrated by two examples. Resolving the error in [11] and understanding why and the consequence of the fact that the bound in [10] does not specialize to the bounds in [3, 7] are topics of future work.

Acknowledgement

We thank Insik Shin for his comment on a previous version of this paper and for pointing us at [10].

References

[1] R. Bril. Towards pragmatic solutions for two-level hierarchical scheduling - Part I: A basic approach for independent applications. Technical Report CSR 07-19, Department of Mathematics and Computer Science, Technische Universiteit Eindhoven (TU/e), The Netherlands, August 2007. http://www.win.tue.nl/ rbril/publications/CSR-07-19.pdf.

[2] G. Buttazzo. *Hard real-time computing systems - predictable scheduling algorithms and applications (2^{nd} edition).* Springer, 2005.

[3] J. Lehoczky, L. Sha, and J. Strosnider. Enhanced aperiodic responsiveness in hard real-time environments. In *Proc. 8^{th} IEEE Real-Time Systems Symposium (RTSS)*, pages 261–270, December 1987.

[4] C. Liu and J. Layland. Scheduling algorithms for multiprogramming in a real-time environment. *Journal of the ACM*, 20(1):46–61, January 1973.

[5] J. Liu. *Real-Time Systems.* Prentice Hall, 2000.

[6] R. Rajkumar, K. Juvva, A. Molano, and S. Oikawa. Resource kernels: A resource-centric approach to real-time and multimedia systems. In *Proc. SPIE, Vol. 3310, Conference on Multimedia Computing and Networking (CMCN)*, pages 150–164, January 1998.

[7] S. Saewong, R. Rajkumar, J. Lehoczky, and M. Klein. Analysis of hierarchical fixed-priority scheduling. In *Proc. 14^{th} Euromicro Conference on Real-Time Systems (ECRTS)*, pages 152–160, June 2002.

[8] I. Shin and I. Lee. Periodic resource model for compositional real-time guarantees. In *Proc. 24^{th} IEEE Real-Time Systems Symposium (RTSS)*, pages 2–13, December 2003.

[9] I. Shin and I. Lee. Periodic resource model for compositional real-time guarantees. Technical Report MS-CIS-03-28, University of Pennsylvania, Department of Computer and Information Science, 2003.

[10] I. Shin and I. Lee. Compositional real-time scheduling framework with periodic model. *ACM Transactions on Embedded Computing Systems*, 7(3), April 2008.

[11] J. Strosnider, J. Lehoczky, and L. Sha. The deferrable server algorithm for enhanced aperiodic responsiveness in hard real-time environments. *IEEE Transactions on Computers*, 44(1):73–91, January 1995.