On Schedulability Analysis of EDF Scheduling by Considering Suspension as Blocking

Mario Günzel and Jian-Jia Chen

Department of Informatics, TU Dortmund University, Germany

Abstract. During the execution of a job, it may suspend itself, i.e., its computation ceases to process until certain activities are complete to be resumed. This paper provides a counterexample of the schedulability analysis by Devi in Euromicro Conference on Real-Time Systems (ECRTS) in 2003, which is the only existing suspension-aware analysis specialized for uniprocessor systems when preemptive earliest-deadline-first (EDF) is applied for scheduling dynamic self-suspending tasks.

1 Introduction

Self-suspension behavior has been demonstrated to appear in complex cyber-physical real-time systems, e.g., multiprocessor locking protocols, computation offloading, and multicore resource sharing, as demonstrated in [3, Section 2]. Although the impact of self-suspension behavior has been investigated since 1990, the literature of this research topic has been flawed as reported in the review by Chen et al. [3].

Although the review by Chen et al. [3] provides a comprehensive survey of the literature, two unresolved issues are listed in the concluding remark. One of them is regarding the “correctness of Theorem 8 in [4, Section 4.5] ... supported with a rigorous proof, since self-suspension behavior has induced several non-trivial phenomena”. This paper provides a counterexample of Theorem 8 in [4, Section 4.5] and disproves the schedulability test.

We consider a set of implicit-deadline periodic tasks, in which each task \( \tau_i \) has its period \( T_i \), worst-case self-suspension time \( S_i \), and worst-case execution time \( C_i \). The relative deadline \( D_i \) is set to \( T_i \). There are two main models of self-suspending tasks: the dynamic self-suspension and segmented (or multi-segment) self-suspension models. Devi’s analysis in [4] considers the dynamic self-suspension model. That is, a task instance (job) released by a task \( \tau_i \) can suspend arbitrarily as long as the total amount of suspension time of the job is not more than \( S_i \).

The analysis by Devi in Theorem 8 in [4, Section 4.5] extended the analysis proposed by Jane W.S. Liu in her book [7, Page 164-165] for uniprocessor preemptive fixed-priority scheduling to uniprocessor preemptive EDF scheduling. Under preemptive EDF scheduling, the job that has the earliest absolute deadline has the highest priority. Despite the non-optimality of EDF for scheduling self-suspending task systems as shown in [8][1], EDF remains one of the most adopted scheduling strategies.

* This work has been supported by Deutsche Forschungsgemeinschaft (DFG), as part of SusAware (Project no. 398602212) and the collaborative research center SP876, subproject A1.
Devi’s analysis quantifies the additional interference due to self-suspensions from the higher-priority jobs by setting up the blocking time induced by self-suspensions. The correctness of the analysis by Liu in [7, Page 164-165] has been proved by Chen et al. [2] in 2016 for fixed-priority scheduling. The authors in [2] noted that “Even though the authors in this paper are able to provide a proof to support the correctness, the authors are not able to provide any rationale behind this method which treats suspension time as blocking time.”

Devi’s analysis for implicit-deadline task systems is rephrased as follows:

**Theorem 1 (Devi [4]).** Let $T = \{\tau_1, \tau_2, \ldots, \tau_n\}$ be a system of $n$ implicit-deadline periodic tasks, arranged in order of non-decreasing periods. The task set $T$ is schedulable using preemptive EDF if

$$\forall k : 1 \leq k \leq n \colon \frac{B_k + B'_k}{T_k} + \sum_{i=1}^{k} \frac{C_i}{T_i} \leq 1,$$

where

$$B_k = \sum_{i=1}^{k} \min\{S_i, C_i\}$$

$$B'_k = \max_{1 \leq i \leq k} \left(\max\{0, S_i - C_i\}\right).$$

Note that the notation follows the survey paper by Chen et al. [3] instead of the original paper by Devi [4]. Moreover, Devi considered arbitrary-deadline task systems with asynchronous arrival times. Our counterexample is valid by considering two implicit-deadline periodic tasks released at the same time.

**2 Counterexample for Devi’s Analysis**

The following task set $T$ with only two tasks provides a counterexample for Devi’s analysis:

- $\tau_1 : (T_1 = D_1 = 6, C_1 = 5, S_1 = 1)$ and
- $\tau_2 : (T_2 = D_2 = 8, C_2 = \epsilon, S_2 = 0)$, for any $0 < \epsilon \leq 1/3$.

The test of Theorem 1 is as follows:

- When $k = 1$, we have $B_1 = 1$ and $B'_1 = 0$. Therefore, when $k = 1$, $\frac{B_k + B'_k}{T_k} + \sum_{i=1}^{k} \frac{C_i}{T_i} = 1$.

- When $k = 2$, we have $B_2 = 1$ and $B'_2 = 0$. Therefore, when $k = 2$, $\frac{B_k + B'_k}{T_k} + \sum_{i=1}^{k} \frac{C_i}{T_i} = \frac{1}{5} + \frac{\epsilon}{5} + \frac{5}{6} = \frac{2 + 3\epsilon}{24} \leq 1$, since $\epsilon \leq 1/3$.

Therefore, Devi’s schedulability test concludes that the task set is feasibly scheduled by preemptive EDF. But, a concrete schedule as demonstrated in Figure [1] shows that one of the jobs of task $\tau_1$ misses its deadline even when both tasks release their first jobs at the same time.

The example in Figure [1] shows that a job of task $\tau_1$ may be blocked by a job of task $\tau_2$, which results in a deadline miss of the job of task $\tau_1$. However, in Devi’s schedulability analysis, such blocking is never considered since $B_1$ and $B'_1$ do not have any term related to $\tau_2$.
3 Conclusion and Discussions

The counterexample in Section 2 only requires task \( \tau_1 \) to suspend once. It shows that applying Devi’s analysis in [4] is unsafe even for the segmented self-suspension model under EDF scheduling. We note that the above counterexample is only for Theorem 8 in [4]. We do not examine any other schedulability tests in [4].

Although there have been many different analyses for preemptive fixed-priority scheduling, the only results for preemptive EDF are the analyses by Liu and Anderson [6], and Dong and Liu [5], which are originally formulated for multiprocessor systems, the suspension-aware analysis by Devi, and the trivial suspension-oblivious analysis, which considers suspension time of the self-suspending tasks as if they are usual execution time. (Detailed discussions can be found in [3, Section 4].) The invalidation of Devi’s analysis implies, that for preemptive EDF scheduling, there is no suspension-aware schedulability test specialized for uniprocessor systems.

References

1. J.-J. Chen. Computational complexity and speedup factors analyses for self-suspending tasks. In Real-Time Systems Symposium (RTSS), pages 327–338, 2016.
2. J.-J. Chen, G. Nelissen, and W.-H. Huang. A unifying response time analysis framework for dynamic self-suspending tasks. In Euromicro Conference on Real-Time Systems (ECRTS), pages 61–71, 2016.
3. J.-J. Chen, G. Nelissen, W.-H. Huang, M. Yang, B. Brandenburg, K. Bletsas, C. Liu, P. Richard, F. Ridouard, N. Audsley, R. Rajkumar, D. de Niz, and G. von der Brüggen. Many suspensions, many problems: a review of self-suspending tasks in real-time systems. Real-Time Systems, 55(1):144–207, 2019.
4. U. C. Devi. An improved schedulability test for uniprocessor periodic task systems. In 15th Euromicro Conference on Real-Time Systems (ECRTS), pages 23–32, 2003.
5. Z. Dong and C. Liu. Closing the loop for the selective conversion approach: A utilization-based test for hard real-time suspending task systems. In Real-Time Systems Symposium (RTSS), pages 339–350, 2016.
6. C. Liu and J. H. Anderson. Suspension-aware analysis for hard real-time multiprocessor scheduling. In 25th Euromicro Conference on Real-Time Systems, ECRTS, pages 271–281, 2013.
7. J. W. S. Liu. Real-Time Systems. Prentice Hall PTR, 1st edition, 2000.
8. F. Ridouard, P. Richard, and F. Cottet. Negative results for scheduling independent hard real-time tasks with self-suspensions. In Real-Time Systems Symposium (RTSS), pages 47–56, 2004.