CISTER
Research Centre in
Real-Time & Embedded
Computing Systems

Journal Paper

Correspondence article: a correction of the reduction-based schedulability analysis for APA scheduling

Felipe Cerqueira
Björn B. Brandenburg
Geoffrey Nelissen*

*CISTER Research Centre

CISTER-TR-180809

2018/08/06

Correspondence article: a correction of the reduction-based schedulability analysis for APA scheduling

Felipe Cerqueira, Björn B. Brandenburg, Geoffrey Nelissen*

*CISTER Research Centre

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: grrpn@isep.ipp.pt

http://www.cister.isep.ipp.pt

Abstract

In this correspondence letter, we document and correct a flaw in the reduction-based analysis for real-time scheduling with arbitrary processor affinities (APA). To provide further confidence, the corrected claims have been formalized and machine-checked using the Coq proof assistant.

Real-Time Systems
https://doi.org/10.1007/s11241-018-9315-x

Correspondence article: a correction of the reduction-based schedulability analysis for APA scheduling

Arpan Gujarati[1] · Felipe Cerqueira[1] · Björn B. Brandenburg[1] · Geoffrey Nelissen[2]

1 Introduction

Multiprocessor real-time operating systems such as VxWorks, LynxOS, QNX, and real-time variants of Linux allow each task to have an *arbitrary processor affinity* (APA), which is the set of processors on which the task is allowed to execute. Gujarati et al. (2013, 2015a) developed two schedulability analyses for hard real-time tasks subject to such APA constraints: a reduction-based approach, first presented at ECRTS 2013 (Gujarati et al. 2013), and an approach based on linear programming (LP), which was introduced in this journal (Gujarati et al. 2015a) alongside a discussion of the earlier reduction-based technique. Unfortunately, the reduction-based technique was later found to be partially flawed (Gujarati et al. 2015b).

This letter serves to document the flaw (along with illustrative counterexamples) and to correct the reduction-based approach presented by Gujarati et al. (2015a). To avoid further errors, the corrected claims have been formally verified with PROSA (Cerqueira et al. 2016a, b), a recently developed framework for mechanized schedulability anal-

Partially funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—391919384.

✉ Björn B. Brandenburg
bbb@mpi-sws.org

Arpan Gujarati
arpanbg@mpi-sws.org

Felipe Cerqueira
felipec@mpi-sws.org

Geoffrey Nelissen
grrpn@isep.ipp.pt

1 Max Planck Institute for Software Systems (MPI-SWS), Paul-Ehrlich-Straße G26, 67663 Kaiserslautern, Germany

2 CISTER Research Centre, ISEP, Polytechnic Institute of Porto (IPP), Rua Alfredo Allen, 535, 4200-135 Porto, Portugal

Published online: 06 August 2018

ysis based on the Coq proof assistant. A more detailed version of this erratum is available as an appendix of the revised ECRTS 2013 paper (Gujarati et al. 2015b).

We note that the limitations identified in this letter pertain only to claims that were carried forward from the earlier conference paper (Gujarati et al. 2015a, Sect. 4); they do not affect any of the later contributions (Gujarati et al. 2015a, Sects. 3, 5, and 6). In particular, the following observations and contributions remain unaffected as they do not depend on the flawed analysis: (i) with regard to feasibility, global job-level dynamic priority (JLDP) scheduling is equivalent to APA JLDP scheduling (Gujarati et al. 2015a, Sect. 3.2); (ii) the LP-based response-time analysis (RTA) for APA scheduling (Gujarati et al. 2015a, Sect. 5); and (iii) the empirical evaluation (Gujarati et al. 2015a, Sect. 6). Notably, even the empirical evaluation of the reduction-based analysis remains valid because the specific schedulability tests used in the experiments happen to not be affected by this erratum.

Before stating the refuted claims along with corresponding counterexamples, we briefly clarify our notation. A task set $\tau = \{T_1, \ldots, T_n\}$ denotes a set of n real-time tasks to be scheduled on a set $\pi = \{\Pi_1, \ldots, \Pi_m\}$ of m identical processors. Each task $T_i = (e_i, d_i, p_i)$ is characterized by a *worst-case execution time* e_i, a *relative deadline* d_i, and a *minimum inter-arrival time* p_i. Each task T_i also has an associated processor affinity $\alpha_i \subseteq \pi$. For any set of processors $\rho \subseteq \pi$, $tasks(\rho) = \{T_k \in \tau \mid \alpha_k \cap \rho \neq \emptyset\}$ defines the set of tasks that can be scheduled on at least one processor in ρ.

2 Refuted claims and counterexamples

In the reduction-based approach (Gujarati et al. 2013, 2015a), the APA scheduling problem is reduced to "global-like" subproblems to which existing global schedulability tests are applied. Specifically, the task set is deemed schedulable under APA scheduling if, for each task, a subproblem can be found that passes a global schedulability test.

It was claimed (Gujarati et al. 2015a, Lemma 5) that *any* global schedulability analysis can be reused for analyzing the "global-like" subproblems. Unfortunately, this is an incorrect overgeneralization: the reduction-based approach is compatible with some, but not *all* schedulability tests for global scheduling. In particular, it is not compatible with *exact* schedulability tests.

Refuted Claim 1 (Lemma 5 of Gujarati et al. (2015a)) *"If a task $T_i \in tasks(\alpha_i)$ is schedulable when the reduced task set $tasks(\alpha_i)$ is globally scheduled on the reduced processor platform α_i using a JLFP policy A, then T_i is also schedulable under APA scheduling of τ on the processor platform π using the same JLFP policy A."*

Counterexample 1 Consider a sporadic task set τ^{spo} consisting of tasks $T_1 = (1, 2, 2)$, $T_2 = (1, 3, 3)$, $T_3 = (5, 1000, 1000)$, and $T_4 = (1, 5, 5)$, to be scheduled on a platform $\pi = \{\Pi_1, \Pi_2\}$. Assume that tasks are ordered by decreasing priorities, *i.e.*, T_1 has the highest priority, T_2 has the second highest priority, and so on. We show that T_4 is schedulable under global fixed-priority (FP) scheduling, but that it is not schedulable under APA scheduling when certain processor affinities are enforced, thus contradicting Refuted Claim 1.

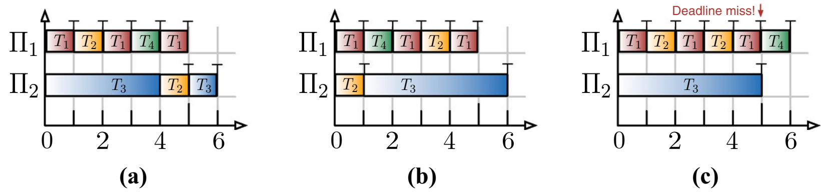


Fig. 1 FP schedules of τ^{spo} under global and APA scheduling. Job completions are denoted by vertical arrows with flat endpoints. Insets **a** and **b** assume global scheduling and correspond, respectively, to Cases 1 and 2 in Counterexample 1. Inset **c** corresponds to the schedule of τ^{spo} under APA scheduling

Let J denote an arbitrary job of T_4, and consider any time window of length 4 starting with the arrival of job J. We consider two cases for the arrival times of tasks T_1 and T_2 to show that T_4 is schedulable under global FP scheduling for any job arrival sequence.

Case 1 *T_1 and T_2 do not release a job simultaneously in the time window*. Even if T_1 and T_2 release jobs as often as possible (limited by their periods), there can be no more than three jobs of T_1 and T_2 in the time window of length 4 (as per assumption their job arrivals never coincide). This limits the total workload of T_1 and T_2 executed in the time window to at most 3 time units. Moreover, because tasks are sequential, the workload of T_3 executed in the time window is at most 4 time units. Thus, the cumulative workload of T_1, T_2 and T_3 executed in this time window of length 4 is at most 7 time units, and there hence exists at least 1 time unit at which one of the two processors is available to T_4 to meet its deadline (see Fig. 1a).

Case 2 *there exists a time t in the time window at which T_1 and T_2 release jobs simultaneously*. Let J_1 and J_2 denote the jobs of T_1 and T_2 that are released in parallel at some time t in this window. Assuming that jobs are released with maximum rate, at most four jobs of T_1 and T_2 are released in the time window of length 4. Thus, the total workload of T_1 and T_2 executed in the window is at most 4 time units. Moreover, since T_3 cannot be scheduled while both higher-priority jobs J_1 and J_2 are executing at time t, the workload that task T_3 executes in the time window is bounded by 3 time units. Thus, the cumulative workload that T_1, T_2, and T_3 execute in the time window is bounded by 7 time units, and there hence exists at least 1 time unit at which either Π_1 or Π_2 is available to T_4 to meet its deadline (see Fig. 1b).

In both cases, it follows that the response time of J is at most 4 time units, which implies that T_4 meets all deadlines under global scheduling. However, this is not the case when certain processor affinities are enforced: if $\alpha_1 = \alpha_2 = \{\Pi_1\}$, $\alpha_3 = \{\Pi_2\}$, and $\alpha_4 = \{\Pi_1, \Pi_2\}$, then T_4 misses a deadline, as shown in Fig. 1c. The fact that T_4 is schedulable under global scheduling (Figs. 1a, 2b) does not imply that it is schedulable under APA scheduling (Fig. 1c), which contradicts Refuted Claim 1. □

While revising the ECRTS 2013 paper (Gujarati et al. 2013) to correct Refuted Claim 1, we further became aware that Lemma 7 of Gujarati et al. (2015a), phrased as if it applies generally to *any* JLFP policy, does not apply to certain policies such as *earliest-deadline first* (EDF). We present a counterexample.

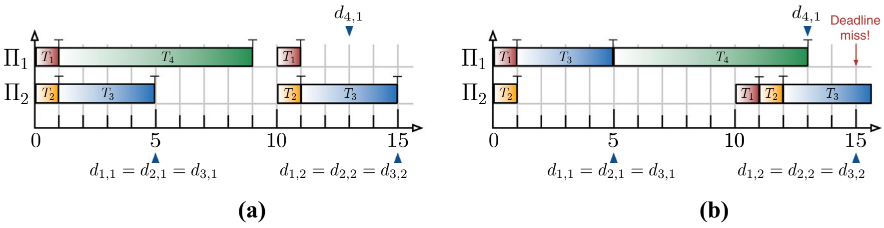


Fig. 2 EDF schedules of task set τ under APA scheduling assuming task T_3 has affinity $\alpha'_3 = \{\Pi_2\}$ (inset **a**), or affinity $\alpha_3 = \{\Pi_1, \Pi_2\}$ (inset **b**). The absolute deadline of the j^{th} job of T_i is denoted as $d_{i,j}$. In inset (a), T_3 meets its deadline at time 15. In inset (b), the first job of T_3 delays a job of T_4, which in turn interferes with the later-released second job of T_3, causing it to miss its deadline at time 15

Refuted Claim 2 (Lemma 7 of Gujarati et al. (2015a)) *"If a task $T_i \in \tau$ is schedulable under APA scheduling with the affinity $\alpha'_i \subset \alpha_i$ and task set τ, then T_i is also schedulable under APA scheduling with affinity α_i and task set τ."*

Counterexample 2 Consider a task set τ consisting of tasks $T_1 = (1, 5, 10)$, $T_2 = (1, 5, 10)$, $T_3 = (4, 5, 10)$, and $T_4 = (8, 13, 13)$, to be scheduled on a platform $\pi = \{\Pi_1, \Pi_2\}$ under APA scheduling using EDF (ties are broken in favor of lower-indexed tasks). The processor affinities are given by $\alpha_1 = \{\Pi_1, \Pi_2\}$, $\alpha_2 = \{\Pi_1, \Pi_2\}$, $\alpha'_3 = \{\Pi_2\}$ and $\alpha_4 = \{\Pi_1\}$, respectively. Let T_3 be the task to be analyzed. Figure 2a shows the schedule of τ assuming a synchronous, periodic arrival sequence: the first jobs of T_1, T_2, and T_3 each have an absolute deadline at time 5, the first job of T_4 has an absolute deadline at time 13, and the second jobs of T_1, T_2, and T_3 each have an absolute deadline at time 15.

Note that T_3 never misses a deadline, as illustrated in Fig. 2a and proven for the general case hereafter. If T_1 and T_2 release jobs simultaneously, as in Fig. 2a, they execute in parallel. If one releases a job later than the other (not shown in the figure), at least one of them either has an absolute deadline past T_3's current job's deadline, or will have completed before T_3's next job is released. In either case, the total delay experienced by any of T_3's jobs due to T_1 and T_2 never exceeds one time unit, and hence there is always enough time for T_3 to complete.

Next, consider the same task set τ except that T_3 has a larger affinity $\alpha_3 = \{\Pi_1, \Pi_2\}$ that includes α'_3. If T_3 starts execution on Π_1 (as allowed under APA scheduling), T_4 is delayed because of its restrictive affinity. As illustrated in Fig. 2b, this causes the first job of T_4 to interfere with the second job of T_3, which causes the latter to miss its deadline at time 15. Thus, even though T_3 is schedulable assuming a reduced affinity $\alpha'_3 = \{\Pi_2\}$, it is not schedulable under EDF with a larger affinity $\alpha_3 = \{\Pi_1, \Pi_2\}$, which contradicts Refuted Claim 2. □

3 APA-compatibility of response-time analysis

Despite these counterexamples, the reduction-based analysis still holds for certain global schedulability tests, which we term *APA-compatible* tests.

In the following, we state the reduction-based schedulability analysis for FP APA scheduling using one such APA-compatible test, namely Bertogna and Cirinei's RTA (Bertogna and Cirinei 2007).

Consider an arbitrary interval $[t_0, t_0 + \Delta)$ of length Δ and a constrained-deadline task T_k with a maximum response-time bound $R_k \leq d_k$. Bertogna and Cirinei (2007) established $W'_k(\Delta) = n'_k(\Delta) \cdot e_k + \min(e_k, \ \ \Delta + R_k - e_k - n'_k(\Delta) \cdot p_k)$ as an upper bound on the maximum workload of T_k during $[t_0, t_0 + \Delta)$, where $n'_k(\Delta) = \lfloor (\Delta + R_k - e_k)/p_k \rfloor$ bounds the maximum number of jobs of T_k that can be scheduled in this interval. Based on this workload bound, $I_i^{k'}(t) = \min(W'_k(t), \ t - e_i + 1)$ denotes an upper bound on the interference of a higher-priority task T_k on task T_i in an interval of length t. If T_k's priority is lower than T_i or if $i = k$, then $I_i^{k'}(t) = 0$.

Theorem 1 *Let τ denote a task set to be scheduled under FP APA scheduling on a processor platform π. For any $T_i \in \tau$, if an upper bound on the maximum response-time $R_k \leq d_k$ is known for each higher-priority task $T_k \in \tau$ (under FP APA scheduling on platform π), and if there exists a non-empty set of processors $\alpha'_i \subseteq \alpha_i$ such that $R_i = e_i + (1/|\alpha'_i|) \cdot \sum_{T_k \in tasks(\alpha'_i)} I_i^{k'}(R_i)$ has a least positive solution $R_i \leq d_i$, then R_i upper-bounds T_i's response-time in any schedule of τ under FP APA scheduling on the processor platform π.*

Proof Proven by Gujarati et al. (2015b, Lemma 9), and verified with a machine-checked proof in Prosa (Cerqueira et al. 2016b, Result A7). □

Bertogna and Cirinei's RTA for EDF scheduling is also APA-compatible. The reduction-based analysis for EDF is precisely stated and proved in Gujarati et al. (2015b, Lemma 10), and mechanized and verified in Prosa (Cerqueira et al. 2016b, Result A8).

4 Discussion

Despite the flaws identified in Refuted Claims 1 and 2, the other analyses and experimental results in both the ECRTS 2013 (Gujarati et al. 2013) and the RTS 2015 (Gujarati et al. 2015a) papers remain valid. As shown in the revised version of the ECRTS 2013 paper (Gujarati et al. 2015b), all four response-time analyses by Bertogna and Cirinei (2007), *i.e.*, RTA for FP scheduling *without* slack updates, RTA for FP scheduling *with* slack updates, RTA for EDF scheduling *without* slack updates, and RTA for EDF scheduling *with* slack updates, are APA-compatible, meaning that they can be applied in the reduction-based approach.

In addition, the issue communicated in this letter does not reduce the practical significance of APA scheduling, since it is widely supported in real-time operating systems such as VxWorks (Wind River Systems 2014), LynxOS (Lynx 2018), QNX (QNX 2010), and real-time variants of Linux. In particular, the EDF APA scheduler SCHED_DEADLINE (Lelli et al. 2016) was merged into mainline Linux in March 2014, complementing the existing FP APA schedulers SCHED_FIFO and SCHED_RR.

Finally, we reiterate that APA scheduling is an interesting research direction that has yielded several publications by various authors in recent years (Baruah and Brandenburg 2013; Cerqueira et al. 2014; Bonifaci et al. 2016). Although we have proven, and verified with a machine-checked proof (Cerqueira et al. 2016b), that Bertogna and Cirinei's RTA is APA-compatible (Gujarati et al. 2015b), identifying and characterizing the class of "all APA-compatible global schedulability tests" remains an open problem that we leave to future work.

References

Baruah S, Brandenburg BB (2013) Multiprocessor feasibility analysis of recurrent task systems with specified processor affinities. In: 34th IEEE international real-time systems symposium (RTSS'13)

Bertogna M, Cirinei M (2007) Response-time analysis for globally scheduled symmetric multiprocessor platforms. In: 28th IEEE international real-time systems symposium (RTSS'07)

Bonifaci V, Brandenburg BB, DAngelo G, Marchetti-Spaccamela A (2016) Multiprocessor real-time scheduling with hierarchical processor affinities. In: 28th Euromicro conference on real-time systems (ECRTS'16)

Cerqueira F, Gujarati A, Brandenburg BB (2014) Linux's processor affinity API, refined: shifting real-time tasks towards higher schedulability. In: 35th IEEE international real-time systems symposium (RTSS'14)

Cerqueira F, Stutz F, Brandenburg BB (2016a) Prosa: a case for readable mechanized schedulability analysis. In: 28th Euromicro conference on real-time systems (ECRTS'16)

Cerqueira F, Stutz F, Brandenburg BB (2016b) Prosa: mechanized proofs of reduction-based APA schedulability analysis. http://prosa.mpi-sws.org/releases/v0.2/apa/

Gujarati A, Cerqueira F, Brandenburg BB (2013) Schedulability analysis of the Linux push and pull scheduler with arbitrary processor affinities. In: 25th Euromicro conference on real-time systems (ECRTS'13)

Gujarati A, Cerqueira F, Brandenburg BB (2015a) Multiprocessor real-time scheduling with arbitrary processor affinities: from practice to theory. Real Time Syst 51(4):440–483

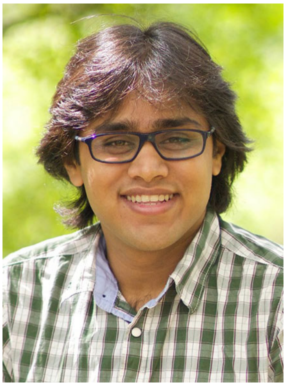
Gujarati A, Cerqueira F, Brandenburg BB (2015b) Revised version: schedulability analysis of the Linux push and pull scheduler with arbitrary processor affinities, revision 1. https://www.mpi-sws.org/~bbb/papers/

Lelli J, Scordino C, Abeni L, Faggioli D (2016) Deadline scheduling in the Linux kernel. Soft Pract Exp 46(6):821–839

Lynx Software Technologies (2018) LynxOS RTOS. http://www.lynx.com/products/real-time-operating-systems/lynxos-rtos/

QNX Software Systems (2010) Processor affinity or bound multiprocessing? Easing the migration to embedded multicore processing. http://www.qnx.com/content/dam/qnx/whitepapers/2010/qnx_bmp_vs_processor_affinity.pdf

Wind River Systems, Inc (2014) Wind River VxWorks Platforms 6.9. https://www.windriver.com/products/product-notes/PN_VE_6_9_Platform_0311.pdf



Arpan Gujarati is a doctoral student at the Max Planck Institute for Software Systems (MPI-SWS) in Kaiserslautern, Germany. Prior to joining MPI-SWS, he completed his undergraduate studies in Computer Science at Birla Institute of Technology and Science in Pilani, India (BITS Pilani), and also worked as a Software Engineer in the Cloud Networking Group at Citrix R&D, India for a year. His research interests span a broad range of topics varying from real-time systems and hard real-time scheduling to distributed systems and fault-tolerant computing.

Felipe Cerqueira is a doctoral student at the Max Planck Institute for Software Systems (MPI-SWS) in Kaiserslautern, Germany. Before moving to Germany, he completed his undergraduate studies in Computer Science at the Federal University of Bahia (UFBA), Brazil. He is interested in both theoretical and practical aspects of real-time scheduling and operating systems. More recently, he has been working on the Prosa project, using a proof assistant to formalize real-time scheduling theory and develop machine-checked schedulability analyses.

Björn B. Brandenburg is a tenured faculty member at the Max Planck Institute for Software Systems (MPI-SWS) in Kaiserslautern, Germany, and head of the Real-Time Systems Group. Prior to joining MPI-SWS, he obtained his PhD under the supervision of Jim Anderson in 2011 and his MSc in 2008, both at the University of North Carolina at Chapel Hill. His work has been recognized with three dissertation awards, several best-paper awards (including at RTSS, RTAS, ECRTS, and EMSOFT), and the inaugural SIGBED Early Career Award in 2018. His current research interests are centered on the design, implementation, and analysis of predictable multiprocessor real-time operating systems and the establishment of a foundation for trustworthy, machine-checked proofs of temporal correctness.

Geoffrey Nelissen is a research scientist at CISTER (Research Centre in Real-Time and Embedded Computing Systems), a research centre co-hosted by the Faculty of Engineering of the University of Porto (FEUP) and the School of Engineering (ISEP) of the Polytechnic Institute of Porto. Prior to joining CISTER, he studied in Brussels at the Université Libre de Bruxelles (ULB), where he earned his PhD in January 2013 and his master degree in electrical engineering in 2008. His research activities are mostly related to the modelling and analysis of real-time and safety critical embedded systems. His research interests span all theoretical and practical aspects of real-time embedded systems design with a particular emphasis on the analysis and configuration of real-time parallel applications on multicore and distributed platforms.