Probabilistic Real-Time Scheduling and its Possible Link to Mixed-Criticality Systems

Georg von der Brüggen*, Sergey Bozhko†, Mario Günzel*,
Kuan-Hsun Chen§, Jian-Jia Chen*, and Björn B. Brandenburg†

*TU Dortmund University, Germany
†Max Planck Institute for Software Systems (MPI-SWS), Germany
§University of Twente, The Netherlands

***Abstract*—Proving hard real-time guarantees based on a classical analysis may significantly underutilize the processor in the average case. Therefore, instead of considering a very rare worst-case scenario, a probabilistic scheduling analysis determines the probability of a deadline miss. Such an analysis assumes that task execution times are given by a set of modes representing the range of possible execution scenarios. Considering tasks with multiple modes and different levels of assurances, in this case expressed as different probabilities to miss deadlines, for different tasks provides a natural link to mixed-criticality systems.**

This work summarizes recent results in probabilistic real-time scheduling and some potential problems that should be considered when linking these results to mixed-criticality systems. In addition, possible connections between mixed-criticality systems and probabilistic analysis are detailed. The goal of this work is to start a discussion to determine whether such probabilistic results may be interesting for mixed-criticality research.

I. Introduction

A classical, deterministic scheduling analysis for hard real-time systems examines the question whether, given a set of (recurrent) tasks, all task instances meet their deadline in all circumstances. These analyses assume that jobs are always executed according to their WCET. Proving timing guarantees under these pessimistic assumptions may significantly underutilize the processor in the average case.

Considering this dilemma, the mixed-criticality approach proposed by Vestal [13] in 2007 has started an active research field within the real-time systems community (the latest version of the survey by Burns and Davis [3] lists 660 related papers). In a mixed-criticality system, tasks have multiple execution modes with different related execution-time budgets. For instance, a dual-criticality system can be in high-criticality or low-criticality mode and is comprised of two kinds of tasks, high-criticality and low-criticality tasks. At system start, the system is in low-criticality mode and timing guarantees are provided for all tasks in the system. If one of the high-criticality tasks overshoots its execution time budget, the system switches to the high-criticality mode, where the execution time budget of high-criticality tasks is increased while no guarantees are provided for low-criticality tasks.

This mixed-criticality model, in its most basic form, has received considerable criticism [10], [9], [14] since low-criticality tasks are abandoned once the system switched to high-criticality mode and no return to low-criticality mode was considered. This resulted in research into more realistic mixed-criticality models and graceful degradation of the service of low-criticality tasks. Please see Section 6 in the survey by Burns and Davis [3] for details.

As a result, mixed-criticality research more frequently considered systems where tasks may switch their execution behavior frequently instead of assuming a single mode switch. In such a scenario, it seems natural to consider situations where only a small subset of tasks exhibits larger execution times for a limited time interval. Hence, performing a system mode switch may be both costly and unnecessary. Furthermore, in 2020, an empirical survey by Akesson et al. [1] revealed that 62% of the responding real-time practioners work on systems that include soft or firm real-time tasks and for 45% of the systems even the most critical functions can endure occasional deadline misses. Hence, an alternative approach to provide guarantees in mixed-criticality systems may be to determine how large the probability for a deadline miss actually is if intervals in which larger execution times occur are short or larger execution times are rare and to adjust runtime measures accordingly. For example, a system mode switch might only be performed if the probability that a deadline miss occurs in a critical function exceeds a certain threshold.

The risk of deadline misses can be quantified in a probabilistic schedulability analysis, usually considering either the deadline miss rate (that is, the percentage of deadline misses in the long run) or the worst-case deadline failure probability (WCDFP) (i.e., an upper bound on the probability of the first deadline miss in a busy window). A survey on probabilistic schedulability in real-time systems community has been provided by Davis and Cucu-Grosjean in 2019 [8]. We provide a summary on recent work in the area, point out open research questions, and possible links to mixed-criticality systems[1]. Our goal is to determine whether such probabilistic results are potentially interesting for the mixed-criticality research community.

[1]This submission is an extension of the one presented at the 15th Workshop on Models and Algorithms for Planning and Scheduling (MAPSP) 2022, which only focused on probabilistic scheduling but did not consider mixed-criticality systems. The version submitted to MAPSP can be found at https://mapsp2022.polito.it/Proceedings.pdf, page 143-145.

II. Probabilistic Analysis: Basics and Problems

We assume that a task's execution time is described as a set of possible modes, each defined by a pair of (i) its maximum execution time in that mode and (ii) the related probability, e.g., $\begin{smallmatrix} C_i \\ \mathbb{P}_i \end{smallmatrix} = \left(\begin{smallmatrix} 3 & 5 \\ 0.9 & 0.1 \end{smallmatrix}\right)$ means that τ_i has an execution time of at most 3 with probability 0.9 and an execution time 5 with probability 0.1.

Assuming a given release pattern, the probability that jobs miss their deadline under a given scheduling algorithm can be calculated via job-level convolution. Figure 1 shows an example of job-level convolution under static-priority scheduling.

The example considers 3 jobs, 2 of the higher-priority task τ_1 and 1 of task τ_2. The goal is to determine the probability that the job of τ_2 misses its deadline. We start in an initial state where the execution time is 0 with probability 1, that is, no job has yet been executed. Jobs are convolved one by one with the current states by summing up the ETs while multiplying the probabilities. This iteratively calculates the probability that the job of τ_2 meets its deadline at $t = 8$ or at $t = 14$, since all possible job-cost combinations are considered.

However, one of the main problems in probabilistic analysis can also be observed in Figure 1, namely, that the number of states can be exponential in the number of jobs for a job-level convolution. Therefore, it can only directly be applied if, on the one hand, the number of jobs that must be considered is small and, on the other hand, the number of release patterns that must be considered is small as well. Otherwise, the computational complexity is too high to be feasible in practice. As a result, two important research questions are:

1) How can the number of release patterns that have to be examined be reduced?
2) How can the deadline miss probability for one of these scenarios be determined efficiently?

Especially in the context of mixed-criticality systems, these calculations must also be applicable when the execution times of jobs are not independent due to a mode switch.

In the following, we give a brief overview on the progress on these fundamental research questions.

III. Efficient Approximation of Miss Probabilities

One approach to speed up the calculation using job-level convolution is reducing the number of states by re-sampling [12]. Specifically, states are combined to reduce the number of states as soon as the number of states exceeds a configurable threshold. However, re-sampling also reduces the precision of the calculation in a way that, in a non-trivial manner, depends on the concrete re-sampling scheme. Markovic et al. [11] introduced optimal re-sampling schemes that minimize the precision loss. However, bounding the loss remains an open problem. Markovic et al. [11] also detailed how cyclic convolution can be used instead of direct convolution to improve the calculation efficiency.

Instead of considering all possible job-cost combinations at the same time, the Monte-Carlo Response Time Analysis by Bozhko et al. [2] analyzes job traces individually. In each iteration, one specific trace (for instance, the one indicated with brown arrows in Figure 1) is sampled. Specifically, in each iteration, jobs are considered one by one, each time drawing one of the possible execution times according to the related probabilities. To estimate the deadline failure probability for the job under analysis, the number of observed deadline misses is counted and divided by the number of iterations, and then combined with an estimate of the confidence interval at a configurable level of assurance. The Monte-Carlo Response Time Analysis is scalable to scenarios with a very large number of jobs and is easily parallelizable. It allows to provide estimates with a known precision interval, but may require an infeasible number of samples when this interval must be too small.

Another approach is to not consider the jobs in order of arrival but to instead evaluate all relevant intervals individually. For instance, for the example in Figure 1, first the deadline failure probability for the interval [0,8] and then for the interval [0,14] would be calculated. The main idea of this approach is to make up for always starting from scratch by speeding up the calculation for each interval. The task-level convolution by von der Brüggen et al. [15] utilizes the fact that, when a specific interval is considered, the workload contributed by a specific task only depends on the number of jobs in a specific mode, but not on their specific order.

Analytic bounds estimate the probability for each interval individually as well. They, however, do not consider individual job modes to determine the workload the jobs contributes. Instead, the probability that the workload in a given interval is larger than the interval length is estimated directly, using analytic bounds. The most prominent approach is the line of work from Chen et al. [6], [4] utilizing Chernoff Bounds. While results exploiting Hoeffding's or Bernstein inequalities [15] are preferable regarding runtime, Chernoff Bounds usually provide a better tradeoff between runtime and precision. However, Chernoff Bounds do not provide any precision guarantees.

IV. Determining a Worst-Case Release Patterns

Similar to the idea of the classical critical instant, one approach to reduce the analysis complexity is to determine a certain scenario that always provides the worst case or an upper bound on the deadline miss probability.

When considering the worst-case deadline failure probability under static-priority scheduling, Maxim and Cucu-Grosjean [12] proposed such a scenario in 2013, and Chen and Chen [4] provided an alternate proof in 2017. Unfortunately, the depicted scenario, which is identical to the classical critical instant, has been contradicted with a counterexample by Chen et al. [5] in 2022. Chen et al. [5] also provided two over-approximations for the worst-case release pattern, which can be utilized to over-approximate the worst-case deadline failure probability. The question whether there is one specific release pattern that always results in a worst-case workload for any interval under static-priority scheduling remains open.

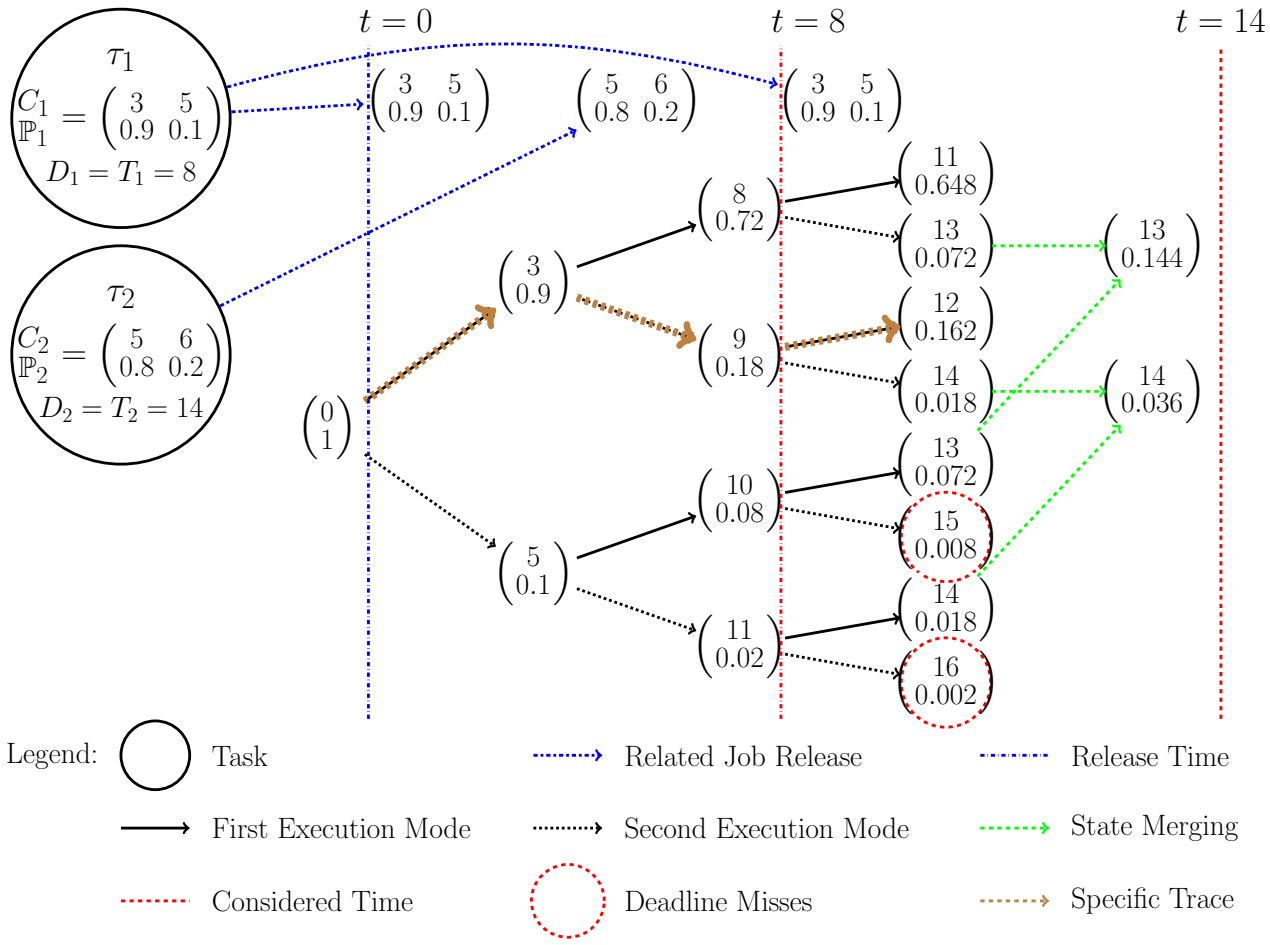

Fig. 1. A convolution example for two tasks under rate-monotonic scheduling.

Fortunately, the results discussed in the previous section are still applicable, as they provide efficient calculation methods for a given release pattern, but do not utilize any specific property of the critical instant.

For earliest-deadline first scheduling, von der Brüggen et al. [16] showed a worst-case scenario that upper-bounds the worst-case deadline failure probability in 2021.

No approach that can analytically bound the deadline miss rate is known under either static-priority scheduling or earliest-deadline first scheduling, as the result by Chen et al. [7] for static-priority scheduling is not applicable anymore due to the recently provided counterexample [5].

V. Job Dependencies

The previously discussed worst-case scenarios and calculation methods assume that the probabilities for job execution times are probabilistically independent. Therefore, applying them to mixed-criticality systems, where all, or at least some, tasks jointly switch into high-criticality mode is not straight forward. Nevertheless, von der Brüggen et al. [16] provided an over-approximation that, under earliest deadline first, enables dependencies in an restricted scenario. Specifically, they assumed that the dependencies can be modelled as acyclic task chains and that job modes depend on the modes of predecessors in those chains. This scenario is similar to mixed-criticality systems where some tasks triggers high-criticality behavior in all tasks in the system. It thus may be applicable to mixed-criticality systems. Alternatively, this approach may allow to analyze scenarios where a subset of the tasks in the system switch to high-criticality mode.

VI. Possible Links

In addition to the links already mentioned so far, there are a number of possible connections between mixed-criticality systems and probabilistic analysis of real-time systems. The notion of different levels of assurance for high-criticality and low-criticality tasks naturally can be interpreted as different thresholds for acceptable residual risk of deadline misses. Especially since high-criticality (or degraded-mode) behavior is expected to be rare at runtime, if lower-criticality tasks are seen as having a higher tolerance for occasional deadline misses, a probabilistic view would allow considerable resources to be reclaimed.

For example, it might be interesting to explore whether it is possible to extend the Monte Carlo approach by Bozhko et al. [2] to a probabilistic mixed-criticality setup. If it is

possible to identify a small number of relevant job-arrival sequences that must be considered, it may be possible to sample bounds on the deadline failure probability with a configurable, criticality-specific level of confidence. In particular, it may be possible to take mode changes into account simply by sampling (also) schedules in which mode changes occur, so that the final probability bounds reflect not only assurances for high-criticality tasks, but also how low-criticality tasks fare in the event of a mode change. In other words, a probabilistic approach could provide low-criticality tasks with much stronger guarantees than merely "best effort" in the event that a higher-criticality mode is entered.

Probabilistic analyses could also exploit another opportunity related to mode changes. A classic dual-criticality analysis must address (at least) three scenarios: the system in stable low-criticality mode, the system in stable high-criticality mode, and crucially, the system as it transitions from low- to high-criticality. The latter case, the time of mode transition is the most challenging aspect from a scheduling point of view, because increased high-criticality demand coincides with the lingering effects of pre-mode-change low-criticality interference, and hence typically represents the "assurance bottleneck." A probabilistic analysis could exploit that it is unlikely that low-criticality tasks exhibit maximum resource demand (and generate maximum interference) at precisely the moment when a high-criticality task triggers a mode change — at least if tasks of different criticalities are independent. A more refined analysis down the line could then further extend such an analysis to take into account possible dependencies between high- and low-criticality tasks.

In a different direction, it would also be interesting to inject the central notion of mixed-criticality systems into probabilistic modeling. Specifically, the idea that high- and low-criticality task parameters express different levels of assurance can also be seen as different levels of confidence in the correctness of specified mode probabilities. For example, when characterizing the chance that a job enters an "exceptional mode" associated with an increased execution cost (rather than its cheaper "normal mode"), it is reasonable to expect that a more risk-averse estimate would be obtained for a high-criticality task than for a low-criticality task. Consequently, it could be interesting to explore a different kind of what-if analysis: what happens to high-criticality tasks if the probability distribution assumed for low-criticality tasks turns out to be optimistic? This is akin to the classic mixed-criticality question — what happens to high-criticality tasks if low-assurance WCET estimates are optimistic — but comes with a twist that makes it considerably more difficult: whereas it is obvious when a low-assurance WCET estimate is exceeded, it is generally much harder to pinpoint when a low-assurance execution-time distribution is refuted by observations in practice. Thus, this line of exploration faces not only hard stochastic analysis problems, but also open question concerning the design of runtime mechanisms that would be appropriate for probabilistic mixed-criticality systems.

VII. Conclusion

Probabilistic timing analysis may be an interesting approach when considering mixed-criticality systems, as it may provide argumentation to postpone or omit a mode switch if the probability that a high-criticality task may miss a deadline is sufficiently small. Furthermore, even if mode switches become unavoidable, a probabilistic analysis may be able to recover much pessimism at a specified degree of residual risk.

However, the field of probabilistic scheduling itself still holds multiple open research questions, especially for establishing worst-case arrival patterns, when bounding the deadline-miss rate, and when considering probabilistically dependent jobs.

Therefore, extensions to mixed-criticality are not straight forward and will require further advances in the field of probabilistic scheduling. It thus seems interesting to start a discussion on how such extensions could benefit mixed-criticality research.

References

[1] B. Akesson, M. Nasri, G. Nelissen, S. Altmeyer, and R. I. Davis. An empirical survey-based study into industry practice in real-time systems. In *41st IEEE Real-Time Systems Symposium, RTSS*, 2020.

[2] S. Bozhko, G. von der Brüggen, and B. B. Brandenburg. Monte carlo response-time analysis. In *42nd IEEE Real-Time Systems Symposium, RTSS*, 2021.

[3] A. Burns and R. Davis. Mixed criticality systems-a review. Technical report, University of York, 2022. 13th edition.

[4] K.-H. Chen and J.-J. Chen. Probabilistic schedulability tests for uniprocessor fixed-priority scheduling under soft errors. In *Symposium on Industrial Embedded Systems*, 2017.

[5] K.-H. Chen, M. Günzel, G. von der Brüggen, and J.-J. Chen. Critical instant for probabilistic timing guarantees: Refuted and revisited. In *Real-Time Systems Symposium*, 2022.

[6] K.-H. Chen, N. Ueter, G. von der Brüggen, and J.-J. Chen. Efficient computation of deadline-miss probability and potential pitfalls. In *Design, Automation & Test in Europe*, 2019.

[7] K.-H. Chen, G. von der Brüggen, and J.-J. Chen. Analysis of deadline miss rates for uniprocessor fixed-priority scheduling. In *24th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, RTCSA*, 2018.

[8] R. I. Davis and L. Cucu-Grosjean. A survey of probabilistic schedulability analysis techniques for real-time systems. *Leibniz Trans. Embed. Syst.*, 6(1):04:1–04:53, 2019.

[9] R. Ernst and M. D. Natale. Mixed criticality systems - A history of misconceptions? *IEEE Design & Test*, 33(5):65–74, 2016.

[10] A. Esper, G. Nelissen, V. Nélis, and E. Tovar. How realistic is the mixed-criticality real-time system model? In *RTNS*, 2015.

[11] F. Markovic, A. V. Papadopoulos, and T. Nolte. On the convolution efficiency for probabilistic analysis of real-time systems. In *Euromicro Conference on Real-Time Systems, ECRTS*, 2021.

[12] D. Maxim and L. Cucu-Grosjean. Response time analysis for fixed-priority tasks with multiple probabilistic parameters. In *Real-Time Systems Symposium*, 2013.

[13] S. Vestal. Preemptive scheduling of multi-criticality systems with varying degrees of execution time assurance. In *RTSS*, 2007.

[14] G. von der Brüggen, K.-H. Chen, W.-H. Huang, and J.-J. Chen. Systems with dynamic real-time guarantees in uncertain and faulty execution environments. In *37th IEEE Real-Time Systems Symposium, RTSS*, 2016.

[15] G. von der Brüggen, N. Piatkowski, K.-H. Chen, J.-J. Chen, and K. Morik. Efficiently approximating the probability of deadline misses in real-time systems. In *Euromicro Conference on Real-Time Systems*, 2018.

[16] G. von der Brüggen, N. Piatkowski, K.-H. Chen, J.-J. Chen, K. Morik, and B. B. Brandenburg. Efficiently approximating the worst-case deadline failure probability under EDF. In *42nd IEEE Real-Time Systems Symposium, RTSS*, 2021.