Towards the safe deployment of runtime monitors in mode-change supported Cyber-Physical Systems

Giann Nandi

David Pereira

José Proença

Eduardo Tovar

CISTER-TR-210608

2021/06/28

Towards the safe deployment of runtime monitors in mode-change supported Cyber-Physical Systems

Giann Nandi, David Pereira, José Proença, Eduardo Tovar

CISTER Research Centre

Polytechnic Institute of Porto (ISEP P.Porto)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: giann@isep.ipp.pt, drp@isep.ipp.pt, pro@isep.ipp.pt, emt@isep.ipp.pt

https://www.cister-labs.pt

Abstract

Complex safety-critical Cyber-Physical Systems require modern approaches that can provide evidence of their correct functioning. Among the many state-of-the-art approaches is runtime verification, which constantly verifies if a system's behavior complies with its specification. However, the coupling of monitors causes an inevitable overhead that could compromise the system's safety. We present the concept of a framework capable of analyzing the schedulability of a set of mode-change supporting Cyber-Physical Systems in the presence of coupled runtime monitors.

Towards the safe deployment of runtime monitors in mode-change supported Cyber-Physical Systems

Giann Spilere Nandi, David Pereira, José Proença, Eduardo Tovar
CISTER Research Centre/ISEP Polytechnic Institute of Porto, Portugal
{giann, drp, pro, emt}@isep.ipp.pt

Abstract

Complex safety-critical Cyber-Physical Systems require modern approaches that can provide evidence of their correct functioning. Among the many state-of-the-art approaches is Runtime Verification, which constantly verifies if a system's behavior complies with its specification. However, the coupling of monitors causes an inevitable overhead that could compromise the system's safety. We present the concept of a framework capable of analyzing the schedulability of a set of mode-change supporting Cyber-Physical Systems in the presence of coupled runtime monitors.

Author Keywords. Runtime Verification, Cyber-Physical Systems, Real-Time Scheduling.

1. Introduction

Runtime Verification is a technique based on Formal Methods that dynamically checks if a set of system's runs satisfy correctness properties regarding formal specifications (Leucker and Schallhart, 2009). It consists of enhancing systems with computational entities, named monitors, that run alongside the system and issue verdicts about its behavior. By being executed together with a target system, monitors incur an inevitable overhead that, depending on its deployment, can negatively influence the system's safety and security properties. Such impact is especially troublesome in the context of safety-critical systems (e.g., Cyber-Physical Systems (CPS)), where the occurrence of faults can result in catastrophes.

2. The Challenge

Among the many safety properties related to CPS is the schedulability of real-time tasks. When we consider software-based monitors, their inevitable performance overhead must also be considered by the real-time system's scheduler. In this work, we are especially interested in safety-critical CPS that support the concept of mode-change, which consists of systems optimizing the utilization of their computational resources by a (possible) combination of *1)* adjusting the scheduling parameters of their current real-time tasks; *2)* adding new tasks that were not needed before; *3)* aborting tasks that are not needed anymore. Supporting changes of modes in safety-critical CPS is challenging as such systems must guarantee the correct scheduling of their task set in each of the modes individually (before and after a mode-change) and during the transition period between two modes.

3. Proposed Work

To aid the obtention of evidence of the correct scheduling of real-time tasks, even in the presence of coupled runtime monitors, we propose a framework that extends a set of works in the real-time literature that support mode-change schedulability analysis. These analyses contemplate specific system configuration settings, including the number of cores, scheduling algorithm, and allocation policy of tasks.

As input for the framework, developers and engineers would need to specify computing nodes with a set of execution modes, their respective task sets and monitors with their scheduling parameters, and the possible mode transitions supported in the system. As output, our framework would tell if a system is schedulable considering finite and possible infinite sequences of mode changes by combining state-of-the-art algorithms and our case-specific tailored analysis.

At the moment, we support two types of system configurations: single-core with a fixed-priority scheduling algorithm (rate monotonic) (Huang, and Chen, 2015), and multi-core, also with rate monotonic, but considering a global allocation policy of the tasks (Baek, Shin, Lee, 2020). The decision to only support such system configurations at this early stage goes in line with one of the latest surveys of industrial practices in the real-time domain (Akesson, 2020). To validate our results, we expect to use our framework to analyze a set of industry-focused use cases and hypothetical use cases that could represent problems that are not yet present in real-world applications.

References

Baek, H., K. G. Shin, and J. Lee. 2020. "Response-Time Analysis for Multi-Mode Tasks in Real-Time Multiprocessor Systems." IEEE Access. https://doi.org/10.1109/access.2020.2992868

Huang, W.-H., and J.-J. Chen. 2015. "Techniques for Schedulability Analysis in Mode Change Systems under Fixed-Priority Scheduling." In 2015 IEEE 21st International Conference on Embedded and Real-Time Computing Systems and Applications. IEEE. https://doi.org/10.1109/rtcsa.2015.36

Akesson, B., M. Nasri, G. Nelissen, S. Altmeyer, and R. I. Davis. 2020. "An Empirical Survey-Based Study into Industry Practice in Real-Time Systems." In 2020 IEEE Real-Time Systems Symposium (RTSS). IEEE. https://doi.org/10.1109/rtss49844.2020.00012

Leucker, M., and C. Schallhart. 2009. "A Brief Account of Runtime Verification." The Journal of Logic and Algebraic Programming. https://doi.org/10.1016/j.jlap.2008.08.004

Acknowledgments

This work was partially supported by National Funds through FCT/MCTES (Portuguese Foundation for Science and Technology), within the CISTER Research Unit (UIDP/UIDB/04234/2020); also by FCT within project ECSEL/0016/2019 and from the ECSEL Joint Undertaking (JU) under grant agreement No 876852. The JU receives support from the European Union's Horizon 2020 research and innovation program and Austria, Czech Republic, Germany, Ireland, Italy, Portugal, Spain, Sweden, Turkey. Disclaimer: This document reflects only the author's view and the Commission is not responsible for any use that may be made of the information it contains