

Real-time scheduling algorithms

With increasing functionality and automation in cyber-physical real-time systems, the computational demand is steadily increasing, while resources available for servicing this demand are limited due to size, weight and power restrictions. As a result a large-scale module integration trend has emerged, so that many different applications are being cohosted on the same processing unit. Some of these functionalities are critical for the correct behavior of the system as a whole, while some others only provide enhanced quality-of-service. A classic example of this scenario is the Integrated Modular Avionics (IMA) system in modern air-crafts, which co-hosts critical functions such as engine-controls and flight-management together with less critical ones such as diagnostics and in-flight entertainment. Such systems are called mixed-criticality real-time systems.

Scheduling of mixed-criticality real-time systems on processing units is recognized as one of the most important problems to be addressed for future CPS, and it is the topic of focus in this project. Designing scheduling algorithms that can efficiently, and if possible optimally, utilize processing capacity is all the more crucial in resource-constrained application domains such as those described above. Unfortunately however, the mixed-criticality real-time scheduling problem has already been identified as a computationally hard problem (NP-Hard), meaning that in all likelihood optimal algorithms are impossible. This hardness results from one of the key scheduling requirements for such systems stated as follows. As long as sufficient processing capacity is available, all system functions must be scheduled to meet their deadlines. But if processing capacity is reduced or the system demand is increased due to some run-time event, then only high-criticality functions must be scheduled to meet their deadlines.

This research involves the design of real-time scheduling algorithms for safety-critical systems to meet their stringent timing requirements. We will focus on scheduling algorithms for both single- as well as multi-core processors. Multi-core processors introduce additional scheduling complexity because of the need to manage urgency (deadline) with concurrency (parallelism). TORCS based simulation platform is being developed to verify these systems. The designed algorithms are also implemented in Real-Time Operating Systems (RTOS) such as LITMUS-RT, ERIKA and MICRIUM for evaluation. Finally, to validate the performance of our algorithms we also plan to develop a Linux-based plug-and-play testing framework.

Principal Investigator	Asst. Prof. Arvind Easwaran
School/Department	School of Computer Science and Engineering