Special issue on scheduling and timing analysis for advanced real-time systems

Robert I. Davis · Emmanuel Grolleau

Published online: 16 December 2014
© Springer Science+Business Media New York 2014

1 Forward

Real-time embedded systems such as those found in automotive, aerospace, and other domains are characterised not only by the need for functional correctness, but also the need for temporal or timing correctness. Typically they continually monitor and respond to stimuli from the environment and the physical systems that they control. In order for such systems to behave correctly, they must not only execute the correct computations, but also do so within predefined time constraints or deadlines on the elapsed time between a stimuli and the corresponding response.

Advances in software development methods, hardware technology, and the need to focus on energy efficiency pose significant challenges to the analysis of the overall timing behaviour of real-time systems. With hardware technology continuing to scale through the sub-micron to the deep sub-micron domain (e.g. 35–10 nm) factors such as the increased probability of circuit failure causing permanent faults, and increased power consumption and problems of heat dissipation come to the fore. Further, increasing hardware capability has enabled significant increases in the volume and complexity of the software deployed in real-time applications; a trend that is set to continue. As a way of managing this complexity, software is often now produced via Model Based Design methods complemented by automatic code generation. Such an approach poses a severe challenge to the effectiveness of current worst-case execution time analysis techniques, but also opens up opportunities for tracking software

R. I. Davis
University of York, York, UK

E. Grolleau (✉)
LIAS, ISAE-ENSMA, 1 avenue Clément Ader BP 40109,
86961 Chasseneuil, France
e-mail: grolleau@ensma.fr

semantics throughout the software life-cycle in order to improve the accuracy of both timing and schedulability analysis.

Timing analysis is an important part of critical systems safety analysis which typically involves computing a safe and tight upper-bound on the maximum execution time required by a program on a particular hardware platform, referred to as the Worst-Case Execution Time (WCET). *Safe* means that the WCET estimate should upper bound the actual execution times that may result when the system runs, including those where some kinds of hardware failures occur, thus allowing for graceful degradation. *Tight* means that the WCET estimate should not be overly pessimistic as this would require an over dimensioning of the system resources. These two fundamental problems are tackled in the two timing analysis papers in this special issue.

Schedulability analysis involves determining if the tasks that make up an application can be guaranteed to meet their deadlines when scheduled according to a particular policy under the control of a Real-Time Operating System. While meeting deadlines is a fundamental requirement in hard real-time systems, the scheduling policy also impacts other factors such as energy consumption. Here dynamic energy consumption may be reduced by using a scheduling policy that allows the processor to run at a lower frequency while still ensuring that all deadlines are met. By contrast, static energy consumption may be reduced by using a policy that produces long idle periods, thus allowing the processor to be put into a low power state. The problem of optimally scheduling a multiprocessor system for minimum energy consumption is addressed in the schedulability analysis paper in this special issue.

This special issue contains three papers at the forefront of real-time systems research. Each of these papers appeared in a preliminary form at the 21st International Conference on Real-Time Networks and Systems (RTNS) in 2013. These preliminary papers received *Outstanding Paper Awards* identifying them as research of the highest quality. The papers that appear in this special issue all contain significant and comprehensive additional contributions.

The first paper is “*Static Probabilistic Worst Case Execution Time Estimation for Architectures with Faulty Instruction Caches*”, by Damien Hardy and Isabelle Puaut. In its preliminary form, this paper received the *Best Paper Award* at RTNS 2013.

Hardware scaling into the deep sub-micron domain while improving performance, results in an increased probability of circuits failing permanently. This effect is particularly significant for the SRAM cells used for cache memory. Space redundancy techniques used to provide the appearance of fault-free chips at current technology levels will no longer be cost effective at smaller scales and other methods such as fine grained disabling and re-configuration are required instead. The reduction in performance over the lifetime of a chip due to the fine grained disabling of cache blocks has serious implications for the validity of WCET estimates. This paper introduces a static analysis method that determines a probabilistic WCET bound given an instruction cache configuration and the probability of an SRAM cell and hence a cache block failure. Since a static analysis is used to bound the longest execution path, and a deterministic cache replacement policy is assumed, the probabilistic nature of this bound derives solely from the probability of cell failure. The method avoids exhaustive evaluation of all combinations of faulty cache blocks and hence remains tractable, while upper bounding the results that would be obtained by an exhaustive evaluation.

Interestingly, the method enables an exploration of the architectural design trade-offs between cache block size, associativity, and cell size, which all impact the WCET bound obtained.

The second paper is, “*A Scheduling Algorithm to reduce the Static Energy Consumption of Multiprocessor Real-Time Systems*” by Vincent Legout, Mathieu Jan and Laurent Pautet. In its preliminary form, this paper received the *Best Student Paper Award* at RTNS 2013.

In the deep sub-micron domain, processor static power consumption due to leakage currents exceeds dynamic power consumption due to capacitive switching. This requires a different approach to managing energy consumption since shutting off circuits becomes more effective than running at lower frequencies. As a result, hardware designs include a number of low power states to manage energy consumption. Here, deeper low power states switch off more hardware components, and hence require a longer transition back to the active state, and have a higher energy penalty for doing so. This paper introduces an optimal multiprocessor scheduling algorithm which makes efficient use of low power states. The schedules produced by the offline algorithm optimise the length of idle periods so that the most effective low power states can be used. These idle periods are then lengthened at runtime when tasks do not utilise all of their execution time budgets. Interestingly, the approach extends to mixed criticality applications, permitting a trade-off between energy consumption and deadline miss rates for low criticality tasks.

The third paper is “*Timing Analysis Enhancement for Synchronous Program*” by Pascal Raymond, Claire Maiza, Catherine Parent-Vigouroux, Fabienne Carrier, and Mihail Asavae.

In the automotive and aerospace industries, Model Based Design methods are often used to automatically generate the code for complex software components. To verify the timing behaviour of such components, it is necessary to obtain tight bounds on their worst-case execution times. This is a challenging problem, since the use of intermediate, auto-generated code can mean that much of the original semantic information is lost. This paper shows how high-level information can be traced from a high-level design in a synchronous language such as Lustre or Scade, and a model checker used to prove the infeasibility of certain paths through the control flow graph. This information is then used to provide constraints used in the computation of the WCET estimate. Interestingly the methods remain applicable when compiler optimisations are applied to the intermediate C code. These optimisations reduce the WCET by a factor of five or more, while the use of high level information reduces the WCET estimates by 40/50% for the optimised/un-optimised code.

Together, these papers form an excellent cross-section of state-of-the-art research applicable to advanced hard real-time systems. They will undoubtedly provide a catalyst for further exciting research in this area in the future.