PROXIMA: A Probabilistic Approach to the Timing Behaviour of Mixed-Criticality Systems

Robert I Davis

Department of Computer Science, University of York, Deramore Lane, York, YO10 5GH, UK.
Email: rob.davis@york.ac.uk

Tullio Vardanega

Department of Mathematics, University of Padova, Trieste 63, 35121 Padova, Italy.
Email: tullio.vardanega@math.unipd.it

Jan Andersson

Aeroflex Gaisler AB, Kungsgatan, 12, 411 19 Göteborg, Sweden. Email: jan@gaisler.com

Francis Vatrinet

Sysgo SAS, Route de Sartrouville 54, 78230 Le Pecq, France. Email: fva@sysgo.com

Mark Pearce, Ian Broster

Rapita Systems Ltd., Atlas House, Osbaldwick Link Road, York YO10 3JB, UK.
Email: {mpearce, ianb}@rapitasystems.com

Mikel Azkarate-Askasua

Ikerlan S.Coop. Paseo J. M. Arizmendiarrieta 2, 20500 Mondragon, Spain.
Email: MAzkarateAskasua@ikerlan.es

Franck Wartel

Airbus Operations SAS Route de Bayonne 316, 31060 Toulouse, France. Email: franck.wartel@airbus.com

Liliana Cucu-Grosjean

INRIA Paris-Rocquencourt, Domaine de Voluceau, BP 105 78153, Le Chesnay France.
Email: liliana.cucu@inria.fr

Mathieu Patte

Astrium SAS, 31 rue des Cosmonautes – ZI du Palays 31402 Toulouse Cedex 4.
Email: mathieu.patte@astrium.eads.ne

Glenn Farrall

Infineon Technologies UK Ltd, Infineon House, Great Western Court, Hunts Ground Road, Bristol, BS34 8HP,UK. Email: glenn.farrall@infineon.com

Francisco J Cazorla

Barcelona Supercomputing Center and IIIA-CSIC, c/Jordi Girona, 29, Edificio Nexus II. 08034 Barcelona, Spain.
Email: francisco.cazorla@bsc.es

Abstract

This position paper outlines the innovative probabilistic approach being taken by the EU Integrated Project PROXIMA to the analysis of the timing behaviour of mixed criticality real-time systems. PROXIMA supports multi-core and mixed criticality systems timing analysis by use probabilistic techniques and hardware/software architectures that reduce dependencies which affect timing. The approach is being applied to DO-178B/C and ISO26262.

Keywords: mixed-criticality systems; probabilistic real-time systems; WCET, software performance.

1 Introduction

EU industries developing Critical Real-Time Embedded Systems (CRTES), such as Aerospace, Space, Automotive, and Railways, face relentless demands for increased processor performance to support advanced new functionality. This demand is due to the ever-rising proportion of system value that is now delivered in software. For these industries, economic success depends on the ability to design, implement, qualify and certify advanced real-time embedded systems within bounded effort and costs as well as pre-deployment assurance. Timing correctness as a means to guaranteed performance is one of the key dimensions of interest to qualification and certification for mission-, business- or safety-critical systems alike. Strong by-design evidence is therefore needed to build solid arguments of correctness that can satisfy certification bodies.

Over the next decade, CRTES industries in Europe will face a once-in-a-life-time disruptive challenge brought about by the transition to multicore processors and the architectural revolution that the advent of the manycore era brings. This step change in both processing capability and architecture (towards complex networked systems on a single chip), provides the opportunity to integrate multiple applications of mixed-criticality levels onto the same hardware platform. This has the advantages of reducing system size, weight and power consumption (SWaP), through a reduction in the number of devices, subsystems, and their cabling and connectors. Such integration has benefits in terms of reduced procurement costs, assembly costs, and improved reliability. However, the challenge also brings a severe threat relating to a key problem of CRTES. Unlike with conventional computing systems, developers of CRTES must provably demonstrate the correctness of the system in terms of both functional and timing/temporal behaviour. Current generation CRTES, based on relatively simple single-core processors, are already extremely difficult to analyse in terms of their temporal behaviour, resulting in incorrect operation that risks costing EU industry in high post-deployment costs (including "no-fault-found" and product recalls). The advent of multicore and manycore platforms exacerbates this problem, rendering timing analysis techniques unable to scale and ineffectual, with potentially dire consequences for the quality and reliability of future products. An innovative new approach is needed.

The PROXIMA approach is to adopt probabilistic analysis techniques to develop an efficient (tractable) and effective (tight) analysis of the temporal behaviour of complex mixed-criticality applications on novel and COTS (commercial-off-the-shelf) multicore and manycore platforms. Solid research results from the FP7 STREP PROARTIS (www.proartis-project.eu) project [1] support this approach. The concept is based on using probabilistic analysis techniques [1, 2, 11, 12, 13] to derive tight bounds on the software timing behaviour of applications, reflecting requirements on failure rates commensurate with their criticality. PROXIMA defines architectural paradigms, usually based on the idea of randomizing the timing behaviour of hardware components, e.g. random replacement caches. These paradigms break the causal dependence in the timing behaviour of execution components at hardware and software level that can give rise to pathological cases, and reduces that risk of timing faults to quantifiably small levels. PROXIMA also supports COTS hardware components via the use of higher level (e.g., software-based) randomization paradigms [13] that compensate for any probabilistic-analysis unfriendly features in them.

2 PROXIMA concepts

PROXIMA aims to enable the CRTES industry to successfully exploit the transition to multicore and manycore processor technology with a development approach that draws the most benefit and incurs the least disruption from it. Benefit will come from the ability to deploy more value-added, competitive-edge, heterogeneous and mixed-criticality functionality in more heavily integrated hardware platforms.

Containment of disruption will come from the ability to develop, analyse, build, and qualify CRTES incrementally. To meet that aim PROXIMA pursues an avenue of innovation relating to *composability in the time domain*, scalable across single-core, multicore and manycore processor architectures, without resorting to static partitioning and its intrinsic need for overprovisioning. Hence PROXIMA will solve a key challenge with mixed-criticality applications: the determination of trustworthy and tight bounds on the timing behaviour of applications. Thus low-criticality applications can be assured to not adversely affect higher-criticality ones while allowing for maximally efficient sharing of hardware and software resources among them, without the resource wastage inherent in fully deterministic approaches that use partitioning at every level.

The challenge is addressed by the use of probabilistic techniques, doing away with much of the need (and cost) of the detailed design knowledge required to causally model the timing behaviour of all system resources of interest. When the resource latency can be accurately captured with a probabilistic law and resource composition is designed to avoid causal dependence, the intrinsic complexity of novel multicore and manycore processor architectures naturally becomes treatable by probabilistic timing analysis.

2.1 High-performance mixed-criticality systems

PROXIMA is developing and exploiting innovative *probabilistic* analysis techniques and associated technology, to replace *deterministic approaches* originally designed for single-core processor systems that are rendered unsuitable or ineffectual with the advent of multicore and manycore architectures. This disruptive change makes current industrial practice inadequate for the development of the next-generation high-performance CRTES. Selective transformations are necessary for the development techniques and implementation technologies, which however can only be sustained if they minimise the cost of adoption. PROXIMA fosters that path of transformation.

The precursor PROARTIS project [1, 2, 11, 12, 13] has broken new ground in the domain of probabilistic timing analysis and paved the way to its application on single-core processors. In particular PROARTIS has shown that a wide range of probabilistic analysis techniques exist (including the theory of copulas, extreme value statistics, etc. [3]), that can be applied to the timing analysis of real-time systems so long as certain assumptions apply, notably *statistical independence* (e.g. times are not dependent on the execution or history) or some *definitional dependence* (times are solely defined by the software/hardware, e.g. constant time). It is important to note that these assumptions do not apply in most hardware/software architectures because the response time of resources (such as caches, pipelines) in modern processors is a (complex) function of the past history of use. Ironically, the fact that the behaviour of those resources is fully deterministic is of no benefit for the purposes of timing analysis. This is because the state space behind it is too vast to be precisely computed for single-core processors and is expected to be intractable for multicore and manycore systems.

The breakthrough strategy envisaged by PROXIMA is to introduce *architectural design principles* that result in temporal behaviour for which the hypothesis of either statistical independence or definitional dependence can be made to hold and therefore enables a meaningful application of probabilistic analysis. This fundamental property is achieved by moving away from deterministic behaviour to time randomised behaviour for jittery execution resources (e.g., cache, network-on-chip, memory allocation etc.) at both the hardware and software level without causing disturbance to the local and global functional behaviour affected by those resources.

2.1 CRTES criticality levels, probabilities, and failure rates

The use of probabilistic bounds in systems that require high assurance may seem counter-intuitive; however, the reality is that probabilistic modelling is a close match to the intrinsic nature of those systems. The mechanical parts of those systems (for example in aircraft) are designed with a failure rate in mind. This is so because effects such as radiation, mechanical stress and extreme temperatures induce a low, but non-zero and cumulative probability of failure for those parts and thus for the computing hardware itself. As a consequence, the system as a whole acquires a distinct probability of failure in a given time interval. This failure rate is measured in terms of the number of failures per hour (or billion hours).

By analogy, deviations in timing behaviour such as, for example the exceedance of given bounds in some execution time duration, may be considered as another type of failure that the system may experience. This reasoning should not be misrepresented as a shift in intent from designing software that meets its functional requirements to designing software that may fail in some well-defined way. Instead, it addresses the risk of execution time variability that originates from *outside* of the software itself, and stems from processor-level hardware resources whose innate jittery timing behaviour cannot be restrained by design other than at the cost of extreme overprovisioning.

The objective of probabilistic timing analysis is to provide WCET (worst case execution time) estimations and end-to-end worst-case response times (WCRT) that can be determined to be "safe enough" with respect to application time constraints, so that they keep the overall failure rate of the application below the specific threshold of acceptability (e.g. 10^{-9} per hour) for that application. Probabilistic and statistical approaches are a natural fit to mixed-criticality systems where applications at different criticality levels have different, domain-specific requirements in terms of acceptable timing failure rates, for example failure rates of 10^{-7} per hour for low criticality and 10^{-9} per hour for high criticality applications.

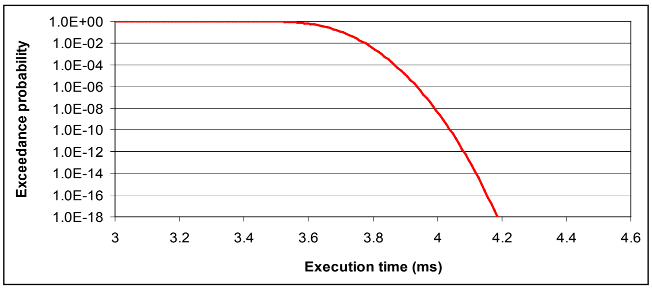

Figure 1 Probability of timing failure per hour

Probabilistic timing analysis provides a continuum of WCET bounds with associated probabilities of exceedance. By way of example, an application may have a probability of less than 10^{-9}, 10^{-13} and 10^{-18} of exceeding an execution time of 4.0ms, 4.1ms and 4.2ms, respectively, each time it executes– see Figure 1. Assuming, as a simple exemplar, that the execution time budget of the application is 4.1ms (for which its WCET has an exceedance probability of 10^{-13} each time it executes), and that it executes at 50Hz (i.e a 20ms period, or 180,000 times per hour) then its expected timing failure rate, due to budget overruns, is less than 10^{-7} per hour, which may be acceptable for a low criticality application.

From this line of reasoning a close relation can be drawn with respect to criticality levels as defined, for example in avionics and automotive standards (DO-178B, ISO-26262) where a failure is defined as a deviation from a specified behaviour, the possible consequences of which determine its severity classification.

- In DO-178B, the Design Assurance level (DAL) is determined from the safety assessment process and hazard analysis by examining the effects of a failure condition in the system. The failure conditions are categorised by their effects on the aircraft, crew, and passengers, with comprehensive analysis methods used to establish the software level A-E: A Catastrophic, B Hazardous, C Major, D Minor and E no failure. Here, catastrophic failure must have a likelihood of occurring that is Extremely Improbably ($<10^{-9}$) – as defined by FAA Advisory Circular AC-25-1309, whereas level B corresponds to Extremely Remote ($<10^{-7}$).
- In ISO-26262, each Automotive Safety Integrity Level (ASIL) is associated with an observable incident rate. Hence applications of ASIL D must have an observable incidence rate lower than 1 every 10^{9} hours, i.e. 10^{-9} per hour. For ASIL C, B, and A the observable incidence rate must be lower than 10^{-8} per hour, 10^{-8} per hour and 10^{-7} per hour respectively.

The probabilistic approach of PROXIMA is a perfect match with those approaches. For applications with high criticality a probabilistic WCET (pWCET) estimate with a low probability of failure is chosen. (In the case of automotive this probability should be smaller than the incident rate in time defined by ISO-26262 multiplied by the number of times an application is executed per hour). In the case of DAL we take the pWCET estimate of each application based on its DAL. If a given application is DAL A, we ensure that the probability of that application having a failure in time is Extremely Improbably. Overall, the objective of probabilistic timing analysis is to provide WCET and end-to-end worst-case response time (WCRT) estimates which are 'safe enough' for the application, the meaning of which is determined by its criticality level.

As part of the PROXIMA project, a detailed analysis is planned examining the possible integration of project outcomes within certification standards and validation processes, with a certification authority acting as an external safety assessment reviewer.

3 Mixed criticality

Mixed-criticality CRTES bring a strong requirement to isolate the behaviour of applications in both the functional and time domains, otherwise the argument for integration is undermined because low criticality applications could impact those of high criticality to an unbounded extent, requiring all to be developed to the same rigorous, expensive and time consuming standard (appropriate for high criticality). To deal with this issue, and not increase verification and validation costs, industries from different domains have developed standardised software frameworks that provide elements of time isolation among software components on single-core processors (e.g. IMA in the avionics domain, and to some extent, AUTOSAR in the automotive domain). Both approaches support a hierarchical development process: the high level integration of the system should be straightforward from the composition of the timing behaviour of the software components. To do so, the system must support the *time composability* property: the worst-case timing behaviour of a component must not change (or change predictably) when other components are integrated into the system. In multicore and manycore processors, this time composability property is not usually obtained because of the *dependences* on the execution time introduced by simultaneous access to shared resources. The execution time may vary greatly depending on the software components being run, i.e. depending on the system integration. Researchers have proposed to upper bound the maximum delay a software component can suffer due to interference when accessing shared resources such as buses [4, 5] or memory controllers [6]. For those resources where considering the maximum delay would remove the benefit of using them, e.g. cache, partitioning solutions have been considered [4].

Much of the recent research into mixed-criticality systems [10] owes its origins to the work of Vestal's [7] which introduced varying degrees of WCET assurance, with larger WCET estimates obtained at higher levels of assurance (criticality level). This research shows that with a mixed-criticality system, simple reservation based policies such as time partitioning (discussed above), or allocation to processing cores based on criticality level can be inefficient; requiring significantly more processing resources than other appropriate scheduling approaches [8].

The alternative of using fixed priority scheduling (as used in automotive i.e. AUTOSAR) and assigning priorities based on criticality also results in severe resource under-utilisation [7, 9]. There is scope therefore for more sophisticated resource sharing policies and analyses to address the overprovision.

4 Time isolation and composition

With the advent of multicore and manycore processors, most complex CRTES are evolving into mixed-criticality systems. A key research question in mixed-criticality CRTES on these platforms is how to reconcile the conflicting requirements of partitioning for assurance and sharing for efficient resource usage [10].

PROXIMA addresses this question with respect to the twin requirements of time isolation and time composition. Asymmetric time isolation ensures that low criticality applications cannot adversely affect the timing behaviour of high criticality applications and hence do not need to be developed or verified to same rigorous standards. Time composability ensures that the guaranteed timing behaviour of an application is not affected by the actual timing behaviour of other applications when the system is integrated. Together, time isolation and composability alleviate the effort and cost of system integration which is a major contributor to overall development costs, by permitting differential verification of software components added to a verified system. To date, timing isolation is normally accomplished via strict partitioning at all levels in the HW/SW stack; however, this comes at a high cost in terms of sizing for the worst case at every level, which while tolerable for single-core will prove unworkable with the transition to multicore and manycore.

The technology developed within the PROXIMA project attacks the root of the time composability problem by reducing, or even completely eliminating, the execution time dependencies resulting from sharing processor resources. As a result, the cost of acquiring the required knowledge to model the timing behaviour of the system can be reduced. In this way, software execution times are less dependent on previous and simultaneous execution of other software components and the system integration can be easily achieved.

The use of probabilistic approaches will recover the time composability property, avoiding the need to consider the maximum delay when accessing shared resources, or using time partitions. In the ideal case, if all the dependence on execution history is eliminated, each individual resource will be time-composable, allowing software components to be replaced without requiring that the timing behaviour of other components is re-analysed.

PROXIMA technology also attacks the problem of overprovision intrinsic in simple partitioning and resource sharing approaches by providing hardware and software mechanisms and policies for resource sharing (between applications at the same and different criticality levels) that promote strong asymmetric isolation. This will minimise overprovision on two counts: firstly by enabling a structured abandonment of low criticality applications commensurate with their assurances and the rare need for high criticality applications to exceed a low assurance WCET budget defined for them. Secondly, by permitting effective resource reclamation when high criticality applications do not make use of their entire resource or WCET budget, permitting where feasible limited overrun capability for low criticality applications, improving their actual failure rates and hence perceived system quality.

5 Conclusions

In this short positional paper, we have outlined the innovative approach being taken by the PROXIMA project towards the analysis of future mixed-criticality real-time systems executing on multi- and many-core hardware platforms. PROXIMA has identified timing correctness as one of key dimensions of interest to qualification and certification of these mission-, business-, or safety-critical systems. The underlying concepts of PROXIMA involve the replacement of existing deterministic analysis techniques that are already reaching their limits on relatively simple single-core processors with more capable probabilistic analysis techniques. These techniques are supported by both hardware and software randomization that reduces the probability of pathological cases occurring to quantifiably low levels, that are significantly below the acceptable failure rates determined for the system.

Acknowledgments

This work has received funding from the European Community's Seventh Framework Programme [FP7/2007-2013] under the PROXIMA Project (www.proxima-project.eu), grant agreement number 611085.

References

[1] F. Cazorla, E. Quinones, T. Vardanega, L. Cucu, B. Triquet, G. Bernat, E. Berger, J. Abella, F. Wartel, M. Houston, L. Santinelli, L. Kosmidis, C. Lo, and D. Maxim (2013). *PROARTIS: Probabilistically analysable real-time systems*. ACM Transactions on Embedded Computing Systems.

[2] L. Cucu-Grosjean, L. Santinelli, M. Houston, C. Lo, T. Vardanega, L. Kosmidis, J. Abella, E. Mezzeti, E. Quinones, F. J. Cazorla (2012). *Measurement-based probabilistic timing analysis for multi-path programs*. In Proceedings of the Euromicro Conference on Real-Time Systems (ECRTS).

[3] W. Feller (1996), *An introduction to Probability Theory and Its Applications*. Wiley.

[4] M. Paolieri, E. Quinones, F.J. Cazorla, G. Bernat, M. Valero (2009). *Hardware Support for WCET Analysis of Multicore Systems*. In proceedings International Symposium on Computer Architecture.

[5] J. Rosen, A. Andrei, P. Eles, Z. Peng (2007). *Bus access optimization for predictable implementation of real-time applications on multiprocessor systems-on-chip*. In proceedings Real-Time Systems Symposium, pp 49-60.

[6] B. Akesson, K. Goossens, M. Ringhofer (2007). *Predator: A predictable SDRAM memory controller*, CODESISSS.

[7] S. Vestal (2007). *Preemptive scheduling of multi-criticality systems with varying degrees of execution time assurance.* In proceedings of the Real-Time Systems Symposium.

[8] S.K. Baruah, V. Bonifaci, G. D'Angelo, H. Li, A. Marchetti-Spaccamela, N. Megow, and L. Stougie (2010). *Scheduling real-time mixed-criticality jobs*. In Proceedings of the 35th International Symposium on the Mathematical Foundations of Computer Science, volume 6281 of Lecture Notes in Computer Science, pp 90–101.

[9] S.K. Baruah, A. Burns, R.I. Davis (2011). *Response Time Analysis for Mixed Criticality Systems*. In proceedings of the Real-Time Systems Symposium (RTSS), pp 34-43.

[10] A. Burns and R. I. Davis (2013), *Mixed Criticality Systems – A Review*. Available from http://www-users.cs.york.ac.uk/~burns/

[11] R.I. Davis, L. Santinelli, S. Altmeyer, C. Maiza, L. Cucu-Grosjean (2013). *Analysis of Probabilistic Cache Related Pre-emption Delays*. In proceedings of the Euromicro Conference on Real-Time Systems (ECRTS), pp 168-179.

[12] L. Kosmidis, E. Quiñones, J. Abella, T. Vardanega, F.J. Cazorla (2013). *Achieving Timing Composability with Measurement-Based Probabilistic Timing Analysis*. In proceedings International Symposium on Object / component / service-oriented Real-time distributed computing.

[13] L. Kosmidis, C. Curtsinger, E. Quinones, J. Abella, E. Berger, F.J. Cazorla (2013). *Probabilistic timing analysis on conventional cache designs*. In proceedings Design, Automation & Test in Europe pp.603-606.