

# The Next Convergence: High-performance and Mission-critical Markets

Sylvain Girbal<sup>†</sup>, Miquel Moretó<sup>\*,‡,⊙</sup>, Arnaud Grasset<sup>†</sup>, Jaume Abella<sup>‡</sup>,  
Eduardo Quiñones<sup>‡</sup>, Francisco J. Cazorla<sup>‡,ϕ</sup>, Sami Yehia<sup>ψ,1</sup>

<sup>†</sup>Thales Research & Technology (TRT), France

<sup>\*</sup>Universitat Politècnica de Catalunya (UPC), Barcelona, Spain

<sup>‡</sup>Barcelona Supercomputing Center (BSC), Spain

<sup>⊙</sup>International Computer Science Institute (ICSI), Berkeley, USA

<sup>ϕ</sup>Spanish National Research Council (IIIA-CSIC), Spain

<sup>ψ</sup>Intel Corporation, USA

**Abstract.** The well-known convergence of the high-performance computing and the mobile markets has been a dominating factor in the computing market during the last two decades. In this paper we witness a new type of convergence between the mission-critical market (such as avionic or automotive) and the mainstream consumer electronics market. Such convergence is fuelled by the common needs of both markets for more reliability, support for mission-critical functionalities and the challenge of harnessing the unsustainable increases in safety margins to guarantee either correctness or timing. In this position paper, we present a description of this new convergence, as well as the main challenges and opportunities that it brings to computing industry.

## 1 Introduction

A dominating factor in the computing market during the last two decades has been the convergence between the high-performance (HP) computing market pursuing for power efficiency and the embedded/mobile market pursuing for more performance and functionalities. This evolution has led to: 1) several open standards to control the balance between low power and high performance in current processors, 2) several APIs to control the low-power hardware features from the software layers (e.g. Operating System, OS) and 3) core designs that can be easily retargeted to provide different performance-power design points depending on the target market.

In this position paper we present our witnessed new type of convergence between the mission critical market (such as the avionic, automotive, healthcare and robotic) and the mainstream consumer electronics market. Such convergence is fuelled by the increasing requirements of the mission critical market for performance and functionalities and the growing needs of the consumer electronics

---

<sup>1</sup> This work was done while Sami Yehia was Research Engineer at Thales Research and Technology.

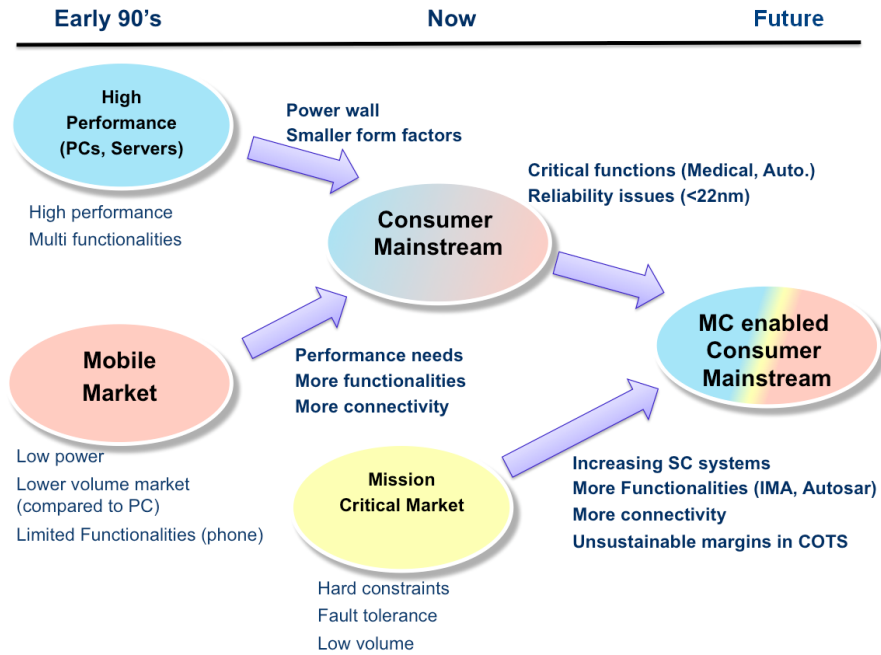


Fig. 1. Past and next convergence

market (such as mobile phones) to embed more critical functionalities and interact with other critical systems (such as car and health monitoring). At the same time, we are confronted with the challenge of harnessing the unsustainable increases in safety margins to guarantee either correctness (DVFS, frequency and/or voltage margins) or timing (WCET margins). We also show that this new convergence brings opportunities and challenges in the way hardware and software have to be designed to deal with mission-critical requirements.

In Section 2 we present the convergence between low power and high-performance markets, which we use as a reference to explain our devised new convergence. In Section 3 we show the main motivation behind the new convergence. Section 4 shows challenges and opportunities brought by this new convergence. We conclude in Section 5.

## 2 Background: low-power and high-performance markets convergence

In the 1990s, the mobile market was clearly a niche market guided only by low power constraints with very low performance and functionality requirements. Three factors motivated the convergence between this market and the mainstream market, see Figure 1. First, the increase in performance and function-

ality requirements of this market made low-power processors to include high-performance features. Second, in the 90's and 00's processors increased their performance at the expense of a rapid increase in power dissipation. For instance, Intel processors increased power from 15W (Pentium) to 115W (Pentium4) in less than 10 years. Limitations in heat dissipation as well as the increase in the energy cost led to a U-turn in high-performance processor design. Low-power features were increasingly incorporated to provide the highest performance under a given power envelope, e.g. Intel released Pentium M right after Pentium 4. Pentium M dissipated up to 27W. And third, processors need several years and astronomical costs to be designed, verified and validated. Hence, reusing designs across domains drastically decreases costs.

After more than 10 years of convergence, several standards have been defined to balance power and performance in all HP processors. One of the most common is the Advanced Configuration and Power Interface ACPI [2]. ACPI defines several low-power states based on stopping the activity, deactivating the clock and even cutting off voltage, as well as means to allow the OS to control the current state, so that different tradeoffs between performance and power can be used dynamically. This enables the same processor design being used in mobile and HP markets despite their different constraints.

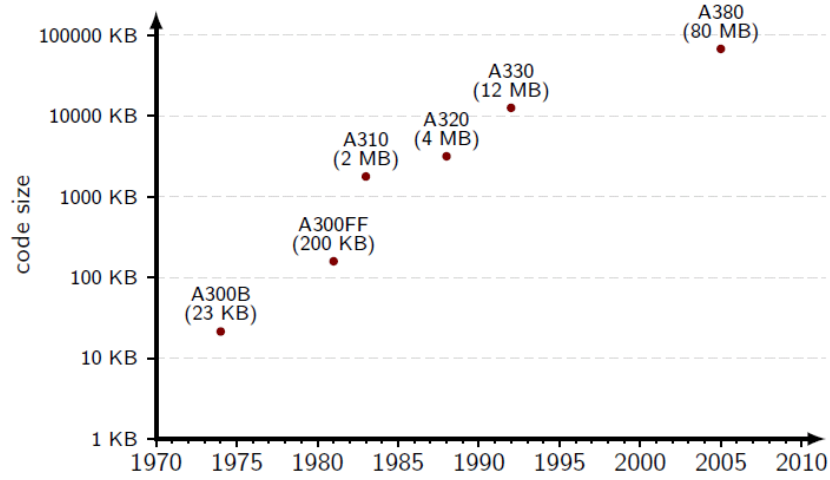
Recently, market convergence has extended also towards the lowest performance/power range of the mobile segment. Processors initially designed for handheld devices such as smartphones are widely used in HP segments such as tablet and netbook segments. For instance, some servers will be released soon based on, for instance, the ARM11 and Intel Atom processors [13]. given that they can provide higher performance/density (e.g., within the volume and power envelope of a particular server). Both ARM and Intel are interested in this path [11].

### 3 The next convergence

#### 3.1 Mission-critical Application Market

The mission-critical system designers have traditionally relied on low performance Components Off-The Shelf (COTS) to address the diverse, but limited in number, service requirements of mission-critical applications. In recent years this design approach has been found to be limited and cost-ineffective due to the growing need for more functionalities and performance, see Figure 2.

These game changing developments in the mission-critical market are underlined by the upcoming - unprecedented in diversity and level - performance requirements in the avionic, automotive and medical domain. For instance, in the avionic domain future generation of navigation systems (four-dimension trajectory and N-Fly zones management systems) [19] in aircrafts, 5th generation cockpit (with the possible introduction of single-crew capable airliners [8]), and collision avoidance are all functionalities with increasing performance requirements. In the automotive domain, future driver assistance systems for vehicles



**Fig. 2.** Growing of embedded software size in avionics

require a level of performance not far from that of a supercomputer (aka HPC or High-Performance Computing) for processing data from cameras, radar, LIDAR and other sensors to detect and decide about warnings, and to control autonomous braking or steering in critical situations. Both domains are over-arched by stringent requirements of lower fuel consumption, CO<sub>2</sub> emissions, with high-speed real-time processing and fault-tolerance. In the medical market, the microcontrollers used in the implantable devices, pacemakers, infusion pumps and digital hearing aids have also significant fault-tolerance, low-power and high-performance requirements.

To address these performance demands, several embedded COTS multicore processors exist in the market. Examples of such architectures are currently envisioned in new avionic and automotive systems. However, because of the increasing complexity of these architectures, analysis of the worst-case scenarios for mission-critical applications is extremely difficult, especially with the emergence of multicores in the embedded market. One typical example of this is the calculation of worst-case execution time (WCET) in hard real-time systems [20] where there is an increasing gap between (1) the WCET and the average best effort performance and (2) the WCET estimates and the actual WCET. This causes system designers to overprovision the system, reducing its power efficiency, while increasing system cost.

In addition, the relatively low volume of the mission critical market makes the development of specific processors meeting the specific requirements of mission critical applications extremely costly because of the high Non-Recurring Engineering (NRE) costs. Most of the solutions in the literature rely on overprovisioning the architecture: COTS processors, on which the OS has no control on the shared resources are used. At software level all the HW resources that

cannot be controlled, such as the cache, are disabled. Aggressive architectural techniques such as branch prediction and data path forwarding are also disabled to ease computing WCET estimates [20].

### 3.2 Mainstream Consumer Electronics Perspective

The microprocessor systems in mainstream or consumer electronics market are also facing challenges. At the hardware level, semiconductor technology poses serious challenges in the design of future microprocessor systems. First the shrinking technology nodes (16 nm in 2013 and 11nm in 2016 according to the ITRS 2010 roadmap [1], 14nm in 2013 according to Intel [4]) will make the processing elements subject to variability constraints, wear-out and soft errors such that the reliability of the basic computing elements will no longer be guaranteed. Thus, reliability will be a first-order design constraint similar to power and performance. Faults at the transistor and circuit levels will be more common than ever such that microprocessor and system designers will have to provide additional fault-tolerance features at several design layers.

At the application level, we foresee an increasing demand in autonomy, decision-making and artificial intelligence. This demand will very likely be over-arched by stringent requirements in term of safety, quality of service (QoS) and hard real-time constraints. Examples of such applications are future domestic robots, healthcare applications [5], intelligent cars, augmented reality, human++ applications [10]. Also the increasing demands in connectivity will naturally lead to a situation where mobile devices directly connect with more mission-critical systems such as cars, medical devices or aircrafts. Hence, providing mobiles with mission-critical capabilities will enable a new type of applications for end users [9].

### 3.3 Mission-critical/mainstream convergence

We observe a second challenging convergence trend, akin to the convergence between the computing PC market and mobile market during the last two decades (See Figure 1), between high-performance (general purpose) and mission-critical markets. On the one hand, the performance requirements of current and future mission-critical applications, as well as their connectivity with the less critical mobile electronic devices, make the use of low performance or over provisioned architectures not a viable solution anymore. On the other hand, applications such as healthcare monitoring will start to hit the mainstream market and the increasing safety margins for reliability issues starts to make the available transistors and available performance more and more difficult to exploit. One key common aspect in mission-critical and mainstream systems is that they both rely on over provisioning some of their resources in order to make some guarantees.

### 3.4 First steps into this convergence

Historically, time predictability was only an issue in the real-time market, where some simultaneous multi-threaded (SMT) processor designs for real-time pro-

cessing provide hardware support for time composability<sup>1</sup> [21]. This support consists in several software-controllable hardware mechanisms to assign more or less resources to the different running threads.

Time predictability is becoming increasingly important outside the real-time domain too. Our view is that most future applications will require some form of hard real-time behavior for at least part of their operation. For domestic robots, cars, planes, telesurgery, and Human++ implants, it is clearly necessary to impose limitations on the delay between sensing and giving the appropriate response. For parallel applications, it is important that all processes running in parallel have balanced execution time in order to maximally exploit the parallel resources of the platform, and limit the synchronization overhead. Especially on heterogeneous multi-cores, being able to accurately estimate execution times is crucial for performance optimization.

Overall, we see a convergence from the high performance market towards features to control resource allocation: several different high-performance processors such as the IBM POWER7, incorporate software-controllable hardware thread priorities [14]. Analogously, we see a convergence of real-time processors incorporating high-performance features such as multicore processing [18].

## 4 Challenges

Next, we attempt to pave the way toward achieving convergence between the mission-critical and the consumer electronic markets so as to reduce the high NRE cost of the mission critical market and provide efficient mission-critical capabilities to the general purpose market.

To achieve such a convergence we need to enrich existing mainstream processing architectures with capabilities that provide non-functional guarantees to address the mission-critical needs, without overprovisioning the processing architecture or diminishing its efficiency. In other words, we want to provide the user some Quality of Service (QoS) according to her or his functional requirements.

### 4.1 User requirements

Because QoS objectives are often application specific, the envisioned QoS approach provides an efficient and general interface that can satisfy QoS objectives over a range of applications. It is required to define the necessary API and language support to express the requirements. A common factor for all type of applications is that the application requirements provided to the QoS system must be architecture independent. This removes the need of the application programmer to provide a different set of metrics for every target architecture, guaranteeing performance and requirements portability. In particular our envisioned QoS system combines the requirements coming from mission-critical-like

<sup>1</sup> An application is time-composable on top of a given hardware/software platform if and only if its timing behaviour remains predictable despite of other applications being run before or concurrently.

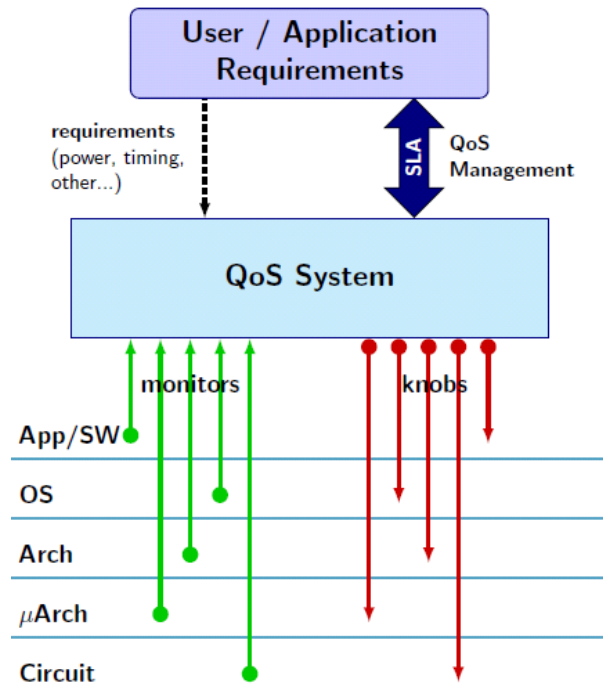


Fig. 3. Integral QoS Approach

applications and general-purpose applications, possibly running them concurrently (mixed criticality systems). In the case of mission-critical applications, we need much stronger guarantees and a pure best effort approach is not sufficient. Several approaches have previously advocated for the introduction of non-functional properties in programming languages, especially domain specific ones [12][17]. For example, timing characteristics should be paramount in the development of the application as well as the architecture.

Historically, QoS standards have been widely used in networked systems in order to ensure high-quality performance of data-flow, especially for real-time multimedia applications, without reactively expanding or over-provisioning the networks. The advent of multicore, reliability issues and the demands of mission-critical applications make the adoption of QoS and resource reservation mechanisms in multicore processor systems the only way to efficiently use these systems and meet these mission-critical demands.

## 4.2 QoS Service Level Agreements

Establishing a Service Level Agreement (SLA) between the application and the OS is the key to offering a guaranteed QoS to the user, especially for mission

critical systems. For example, a user may want to guarantee the completion of a task within a specified deadline. In such a situation, the application will attempt to establish a SLA with the OS before starting the task. The OS will accordingly allocate and block the necessary resources needed to guarantee the completion of the task within the deadline. In the case where the OS is unable to allocate the necessary resources to offer such guarantee, the application will be notified before starting the task and can trigger a backup mechanism. The advantage of such a scheme is that the designer can analyze its application based on a reasonable assumption of available resources, environmental conditions and performance degradation (due to wear-out effects) and not on the assumption of the worst-case scenario where no resources are available for the task or the assumption of an overprovisioned system that will guarantee the availability of resources at the expense of wasting most of them.

### 4.3 Challenges and Opportunities at SW level

At software level, a promising research area are QoS API, Middleware and architecture designs, that allow applications or users to be provided a guaranteed level of service by efficiently using the available resources, according to the abovementioned Service Level Agreement (SLA), and without overprovisioning the architecture to achieve the required level of service. This represents a major departure and different philosophy from previous design approaches that in general address QoS with narrow scope that is neither cost-effective nor scalable. We advocate for holistic approaches across computing layers in which QoS becomes part of the design approach at each computing layer. Similarly to Instruction Set Architectures (ISA) that provide the abstract interface between low-level programming and the hardware implementation, interfaces at each layer must be able to express QoS requirements. QoS compliant designs will need to provide means for bidirectional communication for monitors to observe and knobs to control the system, see Figure 3. Some primitive notions of QoS principles are present in existing systems (such as load balancing, dynamic voltage scaling, etc.) [3] but are not pervasive enough and meant to provide QoS. In other words our aim is to define the necessary services required for meeting non-functional requirements (or properties), the necessary micro-architectural, architectural, middleware and system support to efficiently implement these services, how these services are expressed in the application and how they are deployed over all layers down to the hardware.

Recently, probabilistic timing analysis (PTA) [6] has been delivered as a new approach to consider non-functional properties of the system such as the timing with high accuracy and low cost. In particular, its measurement-based variant (MBPTA) [7] has been shown to provide tight WCET estimates at low cost, thus defeating the limitations of classical approaches. We envision PTA as an appealing path to follow to simplify the analysis of converged systems so that high performance and non-functional QoS guarantees can be achieved at low cost.



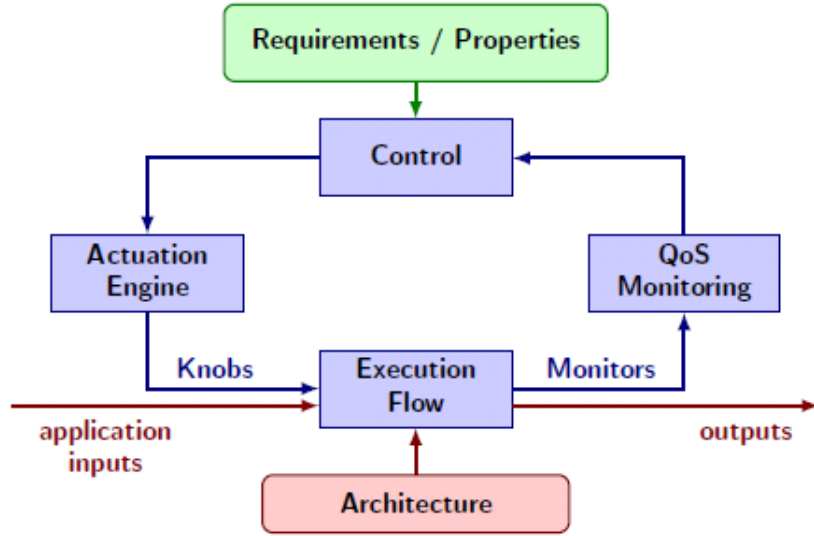


Fig. 4. Detailed view of the Integral QoS Approach

#### 4.4 Challenges and Opportunities at HW level

Monitors are used to collect the different parameters and state of the system periodically and verify that the tasks progress according to the requirements, see Figure 4. The concept of monitors generalizes the standard concept of Performance Monitoring Counters (PMCs) in several aspects. Monitors provide much richer information than PMCs (e.g. information about the interaction between tasks), cover several metrics (not just performance) and provide feedback that allows deducing part of the future behaviour of the running applications.

Depending on the behaviour of the system, the knobs regulate the different tasks and system behaviour to ensure a guaranteed QoS. Monitors and knobs can exist at different layers of the system. For example HW counters at the architecture levels can be used to monitor cache misses, job queues at the OS level can be used to monitor the system load, and temperature sensors at the circuit level can be used to monitor the temperature. Similarly knobs can consist of migrating tasks (OS level), shutting down some cores for power saving (architecture level) or performing some DVFS actions at the circuit level.

In line with PTA, we envision PTA-friendly hardware [15] and software [16] designs that remove abrupt behavior variations due to small changes in the environment by construction, so that the platform becomes more predictable, thus facilitating the work of software layers on top in charge of guaranteeing QoS to the user-level applications.

Further, the fact that non-functional properties become probabilistically predictable with PTA-friendly platforms facilitates certification against safety standards because analogous phenomena is already certified for hardware faults.

## 5 Conclusions

We envision a new type of convergence between the mission-critical market (such as avionic or automotive) and the mainstream consumer electronics market: both markets have common needs for more support for mission-critical functionalities together with an increase in safety margins to guarantee either correctness or timing. To reach these goals future computer designs should implement an integral QoS approach in which QoS becomes part of the design approach at each computing layer. The hardware has to provide features that enable software analyzing and controlling processor internal resource allocation through intelligent knobs and sensors. These hardware features could be enabled from the processor, depending on the mission-critical/performance needs of the target market, so that the same processor design can be used for different mission-critical requirements. The software stack should be able to provide interfaces that allow user QoS requirements to reach the appropriate layer as well as the proper knobs to provide non-functional guarantees to address the mission-critical needs.

## Acknowledgements

M. Moretó, J. Abella, F. J. Cazorla and E. Quiñones have been partially supported by the Spanish Ministry of Science and Innovation under grant TIN2012-34557 and the HiPEAC Network of Excellence. E. Quiñones has also been supported by the Spanish Ministry of Science and Innovation under the grant Juan de la Cierva JCI-2009-05455. M. Moretó is supported by a Fulbright/MEC Fellowship. The authors thank all the anonymous reviewers for their constructive comments and suggestions.

## References

1. International technology roadmap for semiconductors, 2010 update overview. <http://www.itrs.net/Links/2010ITRS/Home2010.htm>, 2010.
2. White Paper November 2008. Intel® turbo boost technology in intel® core™ microarchitecture (nehalem)based processors. 2008.
3. A. Bartolini, M. Cacciari, A. Tilli, L. Benini, and M. Gries. A virtual platform environment for exploring power, thermal and reliability management control strategies in high-performance multicores. In *GLSVLSI*, 2010.
4. M. Bohr. Silicon technology leadership for the mobility era, September 2012. Intel Developer Forum.
5. Stephen V. Cantrill. Computers in patient care: the promise and the challenge. *Commun. ACM*, 53(9):42–47, 2010.
6. F.J. Cazorla, E. Quiñones, 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. PROARTIS: Probabilistically Analysable Real-Time Systems. Technical Report 7869(<http://hal.inria.fr/hal-00663329>), INRIA, to appear in ACM TECS.

7. L. Cucu, L. Santinelli, M. Houston, C. Lo, T. Vardanega, L. Kosmidis, J. Abella, E. Mezzeti, E. Quiñones, and F.J. Cazorla. Measurement-based probabilistic timing analysis for multi-path programs. In *ECRTS*, 2012.
8. Andrew Doyle. Thales outlines thinking on single-crew cockpits. *Flight-globale*, July 2010.
9. Holzbock et al. Evolution of aeronautical communications for personal and multimedia services. *IEEE Communications*, 2003.
10. Pendnders et al. Human++: Emerging technology for body area networks. *VLSI-SoC: Research Trends in VLSI and Systems on Chip*, pages 377–397, 2008.
11. EuroCloud. Energy-conscious 3d server-on-chip for green cloud. <http://www.eurocloudserver.com>.
12. Thomas A. Henzinger, Benjamin Horowitz, and Christoph M. Kirsch. Giotto: A time-triggered language for embedded programming. *Springer-Verlag*, 2001.
13. Intel Newsroom. Chip shot: Intel Atom processors codenamed "Centerton" to power first HP's extreme low-energy production servers [http://newsroom.intel.com/community/intel\\_newsroom/blog/2012/06/19/intel-atom-processors-codenamed-centerton-to-power-first-hps-extreme-low-energy-production-servers](http://newsroom.intel.com/community/intel_newsroom/blog/2012/06/19/intel-atom-processors-codenamed-centerton-to-power-first-hps-extreme-low-energy-production-servers), 2012.
14. Ron Kalla, Balaram Sinharoy, William J. Starke, and Michael Floyd. POWER7: IBM's next-generation server processor. In *IEEE Micro*, volume 30, 2010.
15. L. Kosmidis, J. Abella, E. Quiñones, and F.J. Cazorla. A cache design for probabilistic real-time systems. Technical Report UPC-DAC-RR-CAP-2012-18 (<https://www.ac.upc.edu/app/research-reports/html/2012/24/CacheDesignForPRTS.pdf>), To appear in DATE, 2013.
16. L. Kosmidis, C. Curtsinger, E. Quiñones, J. Abella, E. Berger, and F.J. Cazorla. Probabilistic timing analysis on conventional cache designs. Technical Report UPC-DAC-RR-CAP-2012-20 (<https://www.ac.upc.edu/app/research-reports/html/2012/27/main.pdf>), To appear in DATE, 2013.
17. Edward A. Lee. Computing needs time. *Commun. ACM*, 52(5), 2009.
18. MERASA. Multi-core execution of hard real-time applications supporting analysability. [www.merasa.org](http://www.merasa.org).
19. European Organisation For The Safety of Air Navigation. Study report on avionics systems for the time frame 2007, 2011 and 2020. *Eurocontrol*, 2004.
20. R. Wilhelm, J. Engblom, A. Ermedahl, N. Holsti, S. Thesing, D. Whalley, G. Bernat, C. Ferdinand, R. Heckmann, T. Mitra, F. Mueller, I. Puaut, P. Puschner, J. Staschulat, and P. Stenström. The worst-case execution-time problem—overview of methods and survey of tools. *ACM Transactions on Embedded Computing Systems*, 7(3), April 2008.
21. N. Yamasaki, I. Magaki, and T. Itou. Prioritized SMT architecture with IPC control method for real-time processing. In *RTAS*, 2007.