



CISTER

Research Center in
Real-Time & Embedded
Computing Systems

Conference Paper

Real-time Parallel Applications on Many-core Architectures

Luís Miguel Pinho

Vincent Nélis

Patrick Meumeu Yomsi

CISTER-TR-151206

Real-time Parallel Applications on Many-core Architectures

Luís Miguel Pinho, Vincent Nélis, Patrick Meumeu Yomsi

CISTER Research Center

Polytechnic Institute of Porto (ISEP-IPP)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail:

<http://www.cister.isep.ipp.pt>

Abstract

There is an increasing eagerness to deploy and execute parallel applications on manycore infrastructures, preserving the time-predictability of the execution as required by real-time practices to upper-bound the response time of the embedded application. In this context, this communication describes the application of the currently-available real-time analysis techniques and tools on such platforms and with highly parallel activities, and presents the approach which is being developed within the P-SOCRATES project.

Real-time Parallel Applications on Many-core Architectures

Luís Miguel Pinho, Vincent Nélis, Patrick Meumeu Yomsi
CISTER/INESC-TEC, Polytechnic Institute of Porto, Portugal
email: <lmp, nelis, pamyos>@isep.ipp.pt

***Abstract.** There is an increasing eagerness to deploy and execute parallel applications on many-core infrastructures, preserving the time-predictability of the execution as required by real-time practices to upper-bound the response time of the embedded application. In this context, this communication describes the application of the currently-available real-time analysis techniques and tools on such platforms and with highly parallel activities, and presents the approach which is being developed within the P-SOCRATES project.*

Traditionally, High Performance Computing (HPC) has been the realm and primary focus of specialized industries and specific groups within academia as it demands analytics and simulation applications that require large amounts of data to be processed. Similarly, researchers and industry in the embedded computing (EC) domain have focused mainly on specific systems with specialized and fixed set of functionalities for which timing requirements prevailed over performance requirements. Today, both the HPC and EC domains are broadening their initial focus to other application areas due to the ever-increasing availability of more powerful processing platforms, but therefore they need affordable and scalable software solutions [1], [2].

The need for energy-efficiency (in the HPC domain) and flexibility (in the embedded computing domain), that come along with Moore's law greedy demand for performance and the advancements in the semiconductor technology, have progressively paved the way for the introduction of many-core systems — i.e., multi-core chips containing a high number of cores (tens to hundreds) — in both domains. Today, many-core computing fabrics are being integrated together with general purpose multi-core processors to provide a heterogeneous architectural harness that eases the integration of previously hard-wired accelerators into more flexible software solutions. The HPC computing domain has seen the emergence of accelerated heterogeneous architectures, most notably multi-core processors integrated with General Purpose Graphic Processing Units (GPGPU) [3]. Examples of many-core architectures in the HPC domain include the Intel Xeon Phi [4]. Similarly, the real-time embedded domain has seen the emergence of the Kalray MPPA (Multi-Purpose Processor Array) [5], which includes four quad-core CPUs coupled with a many-core processor. One can also cite the Parallela from Epiphany and the Keystone II from Texas Instrument. In most cases, the many-core fabric acts as a processing accelerator [2].

The introduction of such platforms has set up the basic environment that allowed for the deployment of new types of applications sharing objectives and requirements from both the EC and HPC domains. For such applications, the correctness of the result depends on both performance and real-time requirements, and the failure to meet those is critical to the functioning of the system. Real-time Complex Event Processing (CEP) systems [6] are an example of such applications; they challenge the performance capabilities by crossing the boundaries between the two domains.

It is in that context that the project P-SOCRATES started in October 2013 [7]. P-SOCRATES stands for “Parallel Software Framework for Time-Critical Many-core Systems”. It is an European project which intends to allow current and future applications with high-performance and real-time requirements to fully exploit the huge performance opportunities brought by the most advanced many-core processors, whilst ensuring a predictable performance and maintaining (or even reducing) development costs of applications. The purpose of PSOCRATES is to develop an entirely new design framework, from the conceptual design of the system functionality to its physical implementation, to facilitate the deployment of standardized parallel architectures in all kinds of systems.

The main problem in applying classic real-time techniques to many-core systems is related to the difficulties in deriving reliable and tight upper bounds on the worst-case execution time (WCET) and response time (WCRT) of real-time tasks. Although different timing and schedulability analysis techniques are available in the real-time

literature to derive tight WCET and WCRT bounds in single-core systems, such techniques cannot be easily extended to many-core systems.

Most of the current state-of-the-art techniques for analysis and scheduling assume that the system activities (tasks) are functionally independent and most of their parameters are exactly known at design time. For example, most of the schedulability tests proposed so far assume that the worst-case execution time of an activity is known at design time and invariant. However, when running on a “real” hardware platform, tasks that are co-scheduled on different cores share hardware resources, such as caches, communication buses and main memory, and this introduces implicit functional dependencies among them, as concurrent accesses to the same resource are not allowed, affecting their timing behaviour. This effect magnifies when scaling to a many-core. As a consequence, current analysis and scheduling techniques cannot be applied as-is to many-cores, but they need to be augmented by to include all the sources of contention due to shared resources.

This communication will present how P-SOCRATES is tackling these challenges, both within computation model and software stack as well as the associated analysis to parallelize applications on a many-core architecture while providing guarantees on their response times. Preliminary results towards this direction have already been presented (e.g. [8 - 13]).

Acknowledgments

This work was partially supported by the European Union under the Seventh Framework Programme (FP7/2007-2013), grant agreement n 611016 (P-SOCRATES).

References

- [1] S. Girbal, M. Moreto, A. Grasset, J. Abella, E. Quiñones, F. Cazorla, and S. Yehia, “The next convergence: High-performance and missioncritical markets,” in 1st Workshop on High-performance and Real-time Embedded Systems (HiRES), 2013.
- [2] L. Pinho, E. Quiñones, M. Bertogna, A. Marongiu, J. Pereira-Carlos, C. Scordino, and M. Ramponi, “Time criticality challenge in the presence of parallelised execution,” in 2nd Workshop on High-performance and Real-time Embedded Systems (HiRES), 2014.
- [3] L. Seiler, D. Carmean, E. Sprangle, T. Forsyth, M. Abrash, P. Dubey, S. Junkins, A. Lake, J. Sugerman, R. Cavin, R. Espasa, E. Grochowski, T. Juan, and P. Hanrahan, “Larrabee: A many-core x86 architecture for visual computing,” in ACM SIGGRAPH 2008 papers, vol. 27(3), 2008, pp. 1–15.
- [4] Intel Xeon Phi Product Family, Intel Corporation, last access 8 April 2015, <http://www.intel.com/content/www/us/en/processors/xeon/xeon-phi-detail.html>.
- [5] Kalray, Kalray Corporation, last access 8 April 2015, <http://www.kalrayinc.com>.
- [6] D. Luckham, “The power of events: An introduction to complex event processing in distributed enterprise systems,” in Addison-Wesley Longman Publishing Co. Inc., 2001.
- [7] P-SOCRATES - Parallel Software Framework for Time-Critical Many-core Systems. <http://www.p-socrates.eu>
- [8] Maia, C, Bertogna, M, Nogueira, L, Pinho, L, "Response-Time Analysis of Synchronous Parallel Tasks in Multiprocessor Systems", 22nd International Conference on Real-Time Networks and Systems (RTNS 2014). 8 to 10, Oct, 2014. Versailles, France.
- [9] Nélis, V, Yomsi, P, Pinho, L, "Methodologies for the WCET Analysis of Parallel Applications on Many-core Architectures", Accepted in The Euromicro Conference on Digital System Design (DSD 2015). 26 to 28, Aug, 2015. Funchal, Portugal.
- [10] Fonseca, J, Nélis, V, Raravi, G, Pinho, L, "A Multi-DAG Model for Real-Time Parallel Applications with Conditional Execution", The 30th ACM/SIGAPP Symposium On Applied Computing (SAC 2015). 13 to 17, Apr, 2015, Embedded Systems. Salamanca, Spain.
- [11] A. Melani, M. Bertogna, V. Bonifaci, A. Marchetti-Spaccamela, G. Buttazzo, “Response-Time Analysis of Conditional DAG Tasks in Multiprocessor Systems“, 27th Euromicro Conference on Real-Time Systems, July 2015, Lund, Sweden
- [12] M. A. Serrano, A. Melani, R. Vargas, A. Marongiu, M. Bertogna, E. Quiñones, “Timing Characterization of OpenMP4 Tasking Model”, ACM/IEEE International Conference on Compilers, Architectures and Synthesis of Embedded Systems (CASES), Amsterdam (The Netherlands), October 4-9, 2015
- [13] L. Pinho, V. Nélis, P. Meumeu Yomsi, E. Quiñones, M. Bertogna, P. Burgio, A. Marongiu, C. Scordino, P. Gai, M. Ramponi, M. Mardiak, “P-SOCRATES: a Parallel Software Framework for Time-Critical Many-Core Systems”, Elsevier Microprocessors and Microsystems, to appear.