Microprocessors and Microsystems 38 (2014) 960–975

Contents lists available at ScienceDirect

Microprocessors and Microsystems

journal homepage: www.elsevier.com/locate/micpro

Improving the design flow for parallel and heterogeneous architectures running real-time applications: The PHARAON FP7 project

Héctor Posadas [a,*], Alejandro Nicolás [a], Pablo Peñil [a], Eugenio Villar [a], Florian Broekaert [b], Michel Bourdelles [b], Albert Cohen [c], Mihai T. Lazarescu [d], Luciano Lavagno [d], Andrei Terechko [e], Miguel Glassee [f], Manuel Prieto [g]

[a] University of Cantabria, Santander, Spain
[b] Thales Communications & Security, Gennevilliers, France
[c] INRIA and École Normale Supérieure, Paris, France
[d] Politecnico di Torino, Torino, Italy
[e] Vector Fabrics, Eindhoven, The Netherlands
[f] IMEC, Leuven, Belgium
[g] Tedesys, Santander, Spain

ARTICLE INFO

Article history:
Received 28 October 2013
Revised 7 May 2014
Accepted 8 May 2014
Available online 2 June 2014

Keywords:
European projects
Embedded system
Multiprocessor systems
Heterogeneous systems
Power management

ABSTRACT

In this article, we present the work-in-progress of the EU FP7 PHARAON project, started in September 2011. The first objective of the project is the development of new techniques and tools capable to guide and assist the designer in the development process, from UML specifications to implementation and debug on multicore platform. This tool chain will offer the possibility to propose and implement several parallelization strategies and drive the designer into implementation steps. The second objective of the project is to develop monitoring and control techniques in the middleware of the system capable to automatically adapt platform services to applications requirements and therefore reduce power consumption in a transparent manner for applications.

1. Introduction

Recent market data show that a critical increase in the number of multicore architectures used in projects is currently taking place [1]. During the last decade, those architectures have expanded from only targeting some very specific domains with very high processing needs (e.g. engine control), to become the actual implementation paradigm for mainstream embedded systems. This kind of architectures is getting increasing acceptance into the computing industry, and has become very common in the notebook and tablet markets, among others. This enabled the latest and greatest embedded systems to integrate a growing range of complex functionalities. A smart phone, for example, is capable to communicate through 3G and WIFI connections while running other applications on Android or Windows Phone. It integrates phone services with high performance graphics and sophisticated software applications such as real-time video and audio.

Designers are facing challenging problems as hardware architectures are evolving faster than multicore software development techniques. These techniques are not yet capable to provide efficient methodologies to exploit the full potential of multicore architectures satisfying all the requirements of embedded systems, including performance and power consumption. Accurately predicting the performance of an application implemented on such architectures has become very difficult, because of numerous factors such as cache coherency. Moreover, commonly taught programming models, that are generally based on sequential languages, are no longer sufficient, since early consideration of parallelism in applications has become critical. The lack of efficient software design techniques increases both software development costs and implementation risk in terms of costs and delays. Parallelism, heterogeneity, complex memory structures, efficient power monitors and controllers, are among the list of new functionalities provided by recent multicore systems that require to be adequately tackled by new design

* Corresponding author.
E-mail addresses: posadash@teisa.unican.es (H. Posadas), nicolasa@teisa.unican.es (A. Nicolás), pablop@teisa.unican.es (P. Peñil), villar@teisa.unican.es (E. Villar), florian.broekaert@thalesgroup.com (F. Broekaert), michel.bourdelles@thalesgroup.com (M. Bourdelles), Albert.Cohen@inria.fr (A. Cohen), mihai.lazarescu@polito.it (M. T. Lazarescu), luciano.lavagno@polito.it (L. Lavagno), andrei@vectorfabrics.com (A. Terechko), glasseem@imec.be (M. Glassee), mprieto@tedesys.com (M. Prieto).

http://dx.doi.org/10.1016/j.micpro.2014.05.003
0141-9331/

tools, as proposed by the PHARAON (Parallel and Heterogeneous Architecture for Real-time ApplicatiONs) project.

1.1. The PHARAON project

The PHARAON project is a European collaborative initiative between universities, research labs and companies that is aimed at proposing solutions to overcome these limitations. It is sponsored by the European Commission that covers part of the costs and assists partners in the project management.

The objective of PHARAON is to achieve a breakthrough towards broader adoption of multicore architectures and to enable the development of complex systems with high processing needs and low-power requirements. For such purpose, the project focuses on solving two major problems appearing in these types of systems. First, the development of parallel software, capable of exploiting multiple processor cores, is much more complex and, therefore, more expensive than traditional sequential software, which increases the product cost. Second, the increased complexity of services provided by the systems requires more energy and, hence, is associated with a reduction of autonomy.

To overcome these problems, the PHARAON project targets the development of two different sets of techniques and tools, aiming at best exploiting the low-power capabilities of modern multi-core processors, both at design time and at runtime (Fig. 1). These techniques and tools tackle both the programming and power management challenges mentioned previously.

The first set directly affects the design flow, from UML/MARTE specifications to implementation on multicore platforms. The objective is to assist the designer in finding the most adequate software architecture while taking into account hardware constraints at design time. To do so, tools developed in PHARAON can evaluate the parallel structure of an application and propose improvements, in terms of parallelization constructs. At the same time, the toolset will be able of automatically generating the multi-processor embedded code required to deploy the communicating SW components on the processing cores of the system, including DSPs and GP–GPUs.

The second set of techniques and tools affect the runtime behavior of the application. The objective is to adapt the performance of the platform (frequency and voltage, for example) in order to consume only the required energy. For this purpose, project partners are developing monitoring and control techniques that are integrated in the code generated at design time to map the SW to the processors of the platform. This middleware automatically adapts platform services to application requirements during execution, and therefore reduces power consumption. A reconfiguration system and a low power scheduler are integrated with other run-time components on top of the platform to do so.

As a result, the PHARAON project has the goal of reducing the software development cost by 25% and to increase the battery life of embedded systems by nearly 20%.

The project is coordinated by Thales Communications & Security which is a large French company. Tedesys (Spain) and Vector Fabrics (Netherlands) are two SMEs completing the industrial partners. Academic partners include Politecnico di Torino (Italy), Ecole Normale Supérieure (France) and University of Cantabria (Spain). Finally, the Interuniversitary Micro-Electronics Centrum research institute (Belgium) completes the consortium.

As a result, the next sections present the improvements achieved during the first two years of the project, in which design tools have been developed and their application to the project use cases have started. The second section summarizes the state of the art in the area. Then, the design flow proposed in the project is detailed in the third section. The fourth section presents the design-time tools developed during the project, including their results. The fifth section is devoted to the runtime management tools. Then the application of the tools to several industrial use cases is described. Finally the conclusions highlight the project perspectives.

2. Evolution beyond the state of the art

Code parallelization is one of the most widely studied topics in compilers for parallel machines since the 1970s. However, the level of parallelism that can be identified using automated techniques is very limited, since they require specific coding styles (e.g. perfectly nested loops, no conditionals and affine indexing) and hence have limited applicability.

Recent approaches like the Compaan project at the University of Leiden [2], or the Pico Express high-level synthesis software from

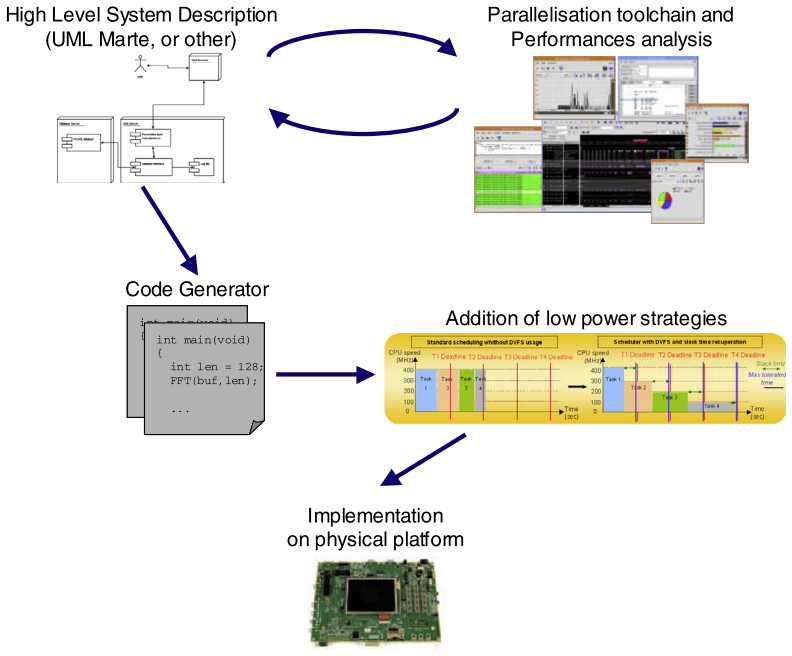

Fig. 1. PHARAON global approach and tools interactions.