

2PARMA

PARallel PARadigms and Run-time MAnagement techniques for Many-core Architectures

KEYWORDS: Parallelisation and Programmability, Continuous Adaptation, Virtualization, Design Space Exploration, Run-time Resource Management, Multi-processor System-on-Chip Architectures



Project Coordinator

Name: **Prof. Cristina Silvano**
Institution: **Politecnico di Milano**
Email: silvano@elet.polimi.it

Project Technical Manager

Name: **Prof. William Fornaciari**
Institution: **Politecnico di Milano**
Email: fornacia@elet.polimi.it

Project website: <http://www.2parma.eu>

Partners:

Politecnico di Milano (I)
STMicroelectronics (I)
Fraunhofer – HHI (D)
IMEC (B),
ICCS (G)
RWTH Aachen University (D)
CoWare (B)

Duration: **36 months**

Start: **2010.01.01**

Total Cost: **€ 3 993 107**

EC Contribution: **€ 2 741 000**

Contract Number: **FP7-ICT-2009-4-248716**

demonstrate methodologies, techniques and tools by using innovative hardware platforms provided and developed by the partners, including the “Platform 2012”, an early implementation of Many-core Computing Fabric provided by STMicroelectronics. To ensure a wide range of application scenarios comprising the typical computation-intensive workload of a general-purpose computing system, a set of industrial high performance demanding applications will be used and customized by using the developed techniques. The selected applications are: Scalable Video Coding (HHI), Cognitive Radio (RWTH / IMEC), and Multi View Image Processing (IMEC).

In concrete terms, 2PARMA project has the following main goals:

Programmability of Many-core Computing

Fabrics: 2PARMA project tackles the issue of programmability of Multi-core Computing Fabrics at both the programming language and Operating System level. On one hand, it leverages the increasingly popular Component-Based Software Engineering and develops parallelism extraction techniques to identify opportunities for parallelisation at a high level in the design phase. Then, 2PARMA employs extensions of existing standards for parallel programming, such as OpenCL, to express data parallelism for Many-core Computing Fabrics. On the Operating System level, 2PARMA provides the means to define and deploy logic peripherals to the Many-core Computing Fabric, preserving isolation among them and efficient communication between host and Computing Fabric.

2PARMA will focus on parallelisation, virtualisation, design space exploration and run-time management of many-core architectures

Main Objectives

The 2PARMA project focuses on the definition of suitable parallel programming models, instruction set virtualisation, run-time resource management policies and mechanisms as well as design space exploration methodologies for Many-core Computing Fabrics. The 2PARMA project will

Virtualisation and Continuous Adaptation: portable bytecode representations of software are employed in 2PARMA project to provide not only portability but also the ability to adapt applications at runtime to the available resources. Instruction set virtualisation provides the means to tailor the application to the subset of the computing resources given by the coexistence of multiple applications on the computing platform, by employing dynamic compilation and optimisation techniques.

Runtime Management: 2PARMA project develops intelligent policies to manage the system resources taking into account the Quality of Service requirements imposed by the application, while optimising the resource usage for system-wide performance and energy consumption goals.

Design Space Exploration: continuous adaptation and run-time management require large amount of information on the system and the applications to take effective and timely decisions. 2PARMA goes *beyond traditional design space exploration (DSE)* by defining a methodology to provide synthetic information about the points of operation of each application with respect to the subsets of resources available to it. DSE methodologies developed in 2PARMA provide also architectural customisation to support parallel programming models, especially communication and memory mapping.

Project Outcomes

Integrated Compiler Toolchain and OS Layer to start from a componentised application source code (C-based) to be semi-automatically parallelised into an OpenCL program and then compiled to bytecode and further dynamically translated to machine code. Then, the machine code execution and deployment will be supported by an OS layer to provide isolated logical devices efficiently communicating (device-to-device and host-to-device).

Design toolset for supporting the HW/SW co-exploration: Starting from a configurable architecture and a parallelised application description (coming from the compilation toolchain) the toolset will explore the HW/SW design space by generating:

- A bottleneck analysis and optimisation of the target architecture with respect to the parallel application;
- Robust tuning of the design-time (static) configurable parameters with respect to run-time (dynamic) system behaviour;
- A set of operating points for the run-time configurable parameters to be used for dynamic resource management.

Run-time Manager: A set of techniques to manage at run-time the system resources. Based on the set of operating points given by the DSE tool at design time and the info collected at run-time on system workload and resource utilization, the run-time management techniques will optimise data allocation and data access scheduling, task mapping and scheduling and power consumption.

