CORRECT AND EFFICIENT ACCELERATOR PROGRAMMING

CARP

Mission Statement

Massively parallel *accelerator processors*, primarily graphics processing units (GPUs), have become widely available to end-users. Accelerators offer tremendous compute power at a low cost, and tasks such as media processing, simulation, medical imaging and eye-tracking can be accelerated to beat CPU performance by orders of magnitude. Performance is gained in energy efficiency and execution speed, allowing intensive media processing software to run in low-power consumer devices. Despite these advantages, accelerators present a serious challenge for software development, due to the growing plethora of accelerator devices on the market. Writing and maintaining software for a diverse range of platforms is not feasible without better programming languages and automated tool support.

The aim of CARP is to design techniques and tools for *correct and efficient accelerator programming*:

- Novel and attractive methods for constructing system-independent accelerator programs
- Advanced code generation techniques to produce highly optimised system-specific code from system-independent programs
- Scalable static techniques for analysing accelerator software *both qualitatively and quantitatively*

These methods will provide a unified development flow for accelerated software, reducing cost and time-to-market quotas, increasing energy efficiency (and thus battery life in mobile devices) and improving reliability.

Technical Approach

As shown in the figure, CARP will centre around the design of a novel language, *PIL* (Portable Intermediate Language), for **productive accelerator programming**. Advanced compilation tools will be investigated, extending and relaxing the **polyhedral model**, and generating low-level code from PIL programs, targeting the widely adopted industry standard **OpenCL**. Portable performance will be achieved through profile-based auto-tuning, enabling compiled code to run efficiently across a range of accelerator platforms. Compiler optimisations will be geared towards reducing execution time and increasing **energy efficiency**. The latter goal will be achieved through quantitative cost analysis, using techniques based on **constraint solving** and **stochastic model checking**.

To aid design of *portably correct* accelerator software, formal analysis at both at the PIL and OpenCL levels will be employed. This will leverage

Contract number

287767

Project coordinator

Imperial College London

Contact person

Alastair F. Donaldson

Department of Computing

Imperial College London

180 Queen's Gate

London SW7 2AZ

UK

Tel: +44 20 7594 8298

Fax: +44 20 7594 8932

alastair.donaldson@imperial.ac.uk

Project website

www.carpproject.eu

Community contribution to the project

2.8 M Euro

Project start date

1 December 2011

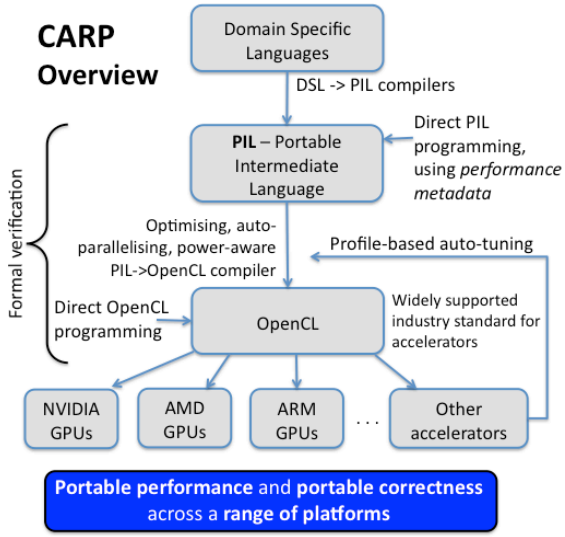
Duration

36 months

and extend existing successes with **symbolic model checking**, **separation logic** and **dynamic symbolic execution**. The aim will be to automatically discover bugs in accelerator software, establish correctness for critical portions of code, *e.g.* libraries, and aid **program understanding**.

Demonstration and Use

The CARP technology will be demonstrated via the implementation of real-time eye-tracking algorithms by **Realeyes**. Currently, Realeyes perform post-processing to extract and analyse gaze information from video streams. The CARP compilation techniques will enable this to be performed in **real-time**, across **multiple platforms** including **mobile devices**, in an **energy-efficient** manner. The verification technology will provide maximum confidence in the **reliability and probity** of this software. The CARP verification techniques will also be demonstrated by **Rightware** and **ARM**, through analysis of their OpenCL benchmark and demo suites.



An overview of the CARP approach

Monoidics and **ARM** will use the state-of-the-art program analysis techniques developed during CARP to enhance their software development tool-chains. In addition, selected software tools arising from CARP will be made publicly available, enabling general adoption.

Scientific, Economic and Societal Impact

ARM's involvement as world leaders in processor design, together with input from three high-technology European SMEs, will ensure rapid commercialisation of the project outputs. This will enhance Europe's competitiveness in the emerging global accelerator software market, ultimately leading to the creation of high-profile technology jobs within EU member states. It is expected that, as a result of the project, **Realeyes** will significantly increase their share of the estimated 800M€ market in eye-tracking and emotion measurement for commercial research, and break into further markets via novel real-time applications. **Monoidics** will significantly increase projects with third parties and sales of their *Infer* product, and **Rightware** will be able to boost annual growth.

Project partners	Country
Imperial College London	UK
École Normale Supérieure	FR
ARM	UK
Realeyes (**SME**)	EE
RWTH Aachen University	DE
Monoidics (**SME**)	UK
Twente University	NL
Rightware (**SME**)	FI

These benefits will in large part be facilitated by collaboration with four of Europe's leading academic research centres: **Imperial College London**, **École Normale Supérieure**, **RWTH Aachen University** and **Twente University**, whose research efforts will lead to exciting scientific discoveries and high-impact publications, and who will themselves benefit enormously from such an intensive collaboration with industry.

For society at large, CARP will result in increased reliability and capability of consumer software, especially running in mobile devices, as well improvements in critical application domains where accelerators are becoming prominent, such as real-time medical imaging.

Key Features

- Order-of-magnitude improvement in productivity of accelerator software development
- Performance of compiled code competitive with that of hand-optimised code, on multiple accelerator platforms
- Lower energy consumption by accelerated software, leading to greener systems, improved battery life in mobile devices, & wider availability on economical platforms