CORRECT AND EFFICIENT ACCELERATOR PROGRAMMING

CARP

Mission Statement

Massively parallel *accelerators*, primarily GPUs, are now widespread, offering tremendous compute power at a low cost. Tasks such as media processing, simulation, medical imaging and emotion tracking can be accelerated to beat CPU performance, sometimes by orders of magnitude, and performance is gained in both energy efficiency and execution speed, allowing intensive media processing software to run in low-power consumer devices. However, low-level programming of heterogeneous accelerators is notoriously difficult.

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 improving reliability.

Contract number

287767

Project coordinator

Imperial College London

Contact person

Alastair F. Donaldson
Department of Computing
Imperial College London
London SW7 2AZ, UK
Tel: +44 20 7594 8298
Fax: +44 20 7594 8932
alastair.donaldson@imperial.ac.uk

Project website

www.carpproject.eu

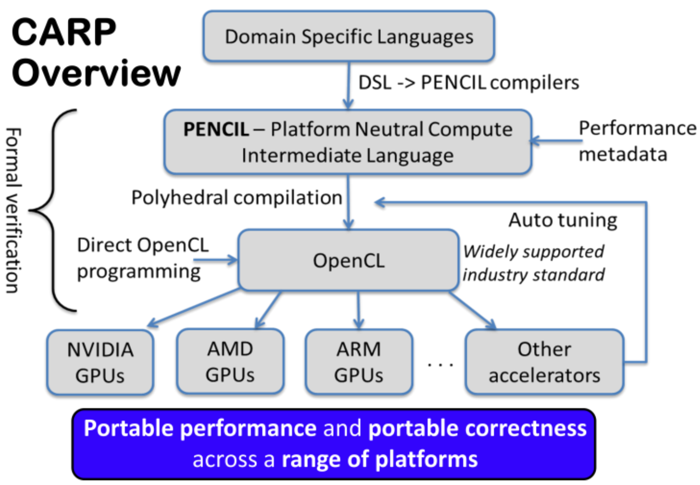
Community contribution to the project: 2.8 M Euro

Start date: 1 December 2011

Duration: 36 months

Technical Approach

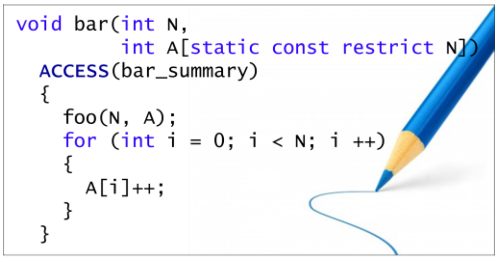
As shown in the figure, CARP is centred around the design of a novel language, *PENCIL* (Platform Neutral Compute Intermediate Language), for **productive accelerator programming**. PENCIL is supported by advanced compilation tools based on the **polyhedral model**, which generate low-level code targeting the widely adopted **OpenCL** industry standard. 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**.



To aid design of *portably correct* accelerator software, the CARP approach involves formal analysis at both the PENCIL and OpenCL levels using deductive verification methods, including **automated Hoare logic** and **separation logic**. 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**.

Selected results so far (see `www.carpproject.eu` for the latest news and publications)

- **PENCIL language definition:** The first draft of the PENCIL language is complete, allowing construction of compilation tools. For early details of PENCIL see our paper at **WOLFHPC'12**.
- **DSL → PENCIL → OpenCL compilation flow:** We have a prototype tool chain for compiling programs in VOBLA, a linear algebra DSL, to PENCIL and then to OpenCL for acceleration on GPU architectures.
- **GPUVerify tool:** GPUVerify is a highly automatic tool for source-level analysis of GPU kernels written in OpenCL and CUDA. See our **OOPSLA'12** paper for details, and check out GPUVerify on YouTube.
- **Techniques for software cost analysis**: We have designed methods for estimating the worst-case execution time of GPU kernels, and for formally establishing quantitative properties of probabilistic programs. See papers at **ECRTS** and **QEST**.




```
GPUVerify kernel analyser finished
with 0 verified, 1 error

kernel.cl: error: possible read-
write race on ((char*)A)[512]:

kernel.cl:4:7:  write by thread
(128, 0, 0) group (0, 0, 0)
  A[tid] = A[tid + 1];

kernel.cl:4:16: read by thread
(127, 0, 0) group (0, 0, 0)
  A[tid] = A[tid + 1];
```

Demonstration and Use

The CARP technology will be demonstrated via the implementation of real-time emotion 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. **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 emotion tracking and emotion measurement for commercial research, and break into new markets via real-time applications. **Rightware** will also be able to enhance their OpenCL benchmark suites, and **Monoidics** will market next-generation program analysis tools for accelerators.

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

These benefits will be facilitated by collaboration with four of Europe's leading academic research centres: **Imperial College London**, **École Normale Supérieure**, **RWTH Aachen University** and **University of Twente**, 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 platforms
- Lower energy consumption by accelerated software, leading to greener systems, improved mobile battery life and wider availability on economical platforms