Programming paradigms for GPU devices

1-3 April 2019
Cineca – Roma, Via dei Tizii, 6B
Teachers: L. Ferraro, S. Orlandini

AGENDA

1 April

09.15 Registration

9:30 – 13:00
- GPGPU Computing
- OpenACC intro
- hands-on

13:00 – 14:00 - Lunch time

14:00 – 17:30
- CUDA programming model
- CUDA event, error …
- hands-on

2 April

9:30 – 13:00
- CUDA memory management (Global memory, Shared memory)
- hands-on

13:00 – 14:00 - Lunch time

14:00 – 17:30
- CUDA streams
- CUDA multi-gpu
- hands-on

3 April

9:30 – 13:00
- CUDA Profiling and Optimizations
- overview of GPU enabled libraries
- hands-on

13:00 – 14:00 - Lunch time

14:00 – 17:30
- From CUDA to OpenCL
- hands-on

Programming paradigms for GPU devices

Content:

This course gives an overview of the most relevant GPGPU computing techniques to accelerate computationally demanding tasks on HPC heterogeneous architectures based on GPUs.
The course will start with an architectural overview of modern GPU based heterogeneous architectures, focusing on its computing power versus data movement needs. The course will cover both a high level (pragma-based) programming approach with OpenACC for a fast porting startup, and lower level approaches based on nVIDIA CUDA and OpenCL programming languages for finer grained computational intensive tasks. A particular attention will be given on performance tuning and techniques to overcome common data movement bottlenecks and patterns.

Learning outcomes:

The students will learn the strengths and weaknesses of GPUs as accelerators, the problem of data movement between host and device memories and how to overcome bottlenecks through concurrent computing/data-movement operations. The students will learn the basics of GPGPU programming using both higher and lower level programming approaches.

Topics:

Overview of architectural trends of GPUs in HPC. GPGPU parallel programming in heterogeneous architectures. Basis of OpenACC, CUDA and OpenCL programming.

Target audience:

Researchers and programmers interested in porting scientific applications or use efficient post-process and data-analysis techniques in modern heterogeneous HPC architectures.

Prerequisites:

A basic knowledge of C or Fortran is mandatory. Programming and Linux or Unix. A basic knowledge of any parallel programming technique/paradigm is recommended.