Programming paradigms for GPU devices

CINECA HPC Training Courses
Teachers: L. Ferraro, S. Orlandini

AGENDA

Day 1 – Introduction to GPGPU with OpenACC

9:30 – 13:00
- GPGPU Computing
- OpenACC intro (part 1)

13:00 – 14:30 - Lunch time

14:30 – 17:30
- OpenACC intro (part 2)
- hands-on

Day 2 – Introduction to CUDA basics

9:30 – 13:00
- CUDA programming model
- Porting a CPU code to CUDA

13:00 – 14:30 - Lunch time

14:30 – 17:30
- CUDA event, error …
- Matrix Product using Global Memory
- hands-on

Day 3 – Advanced CUDA topics

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

13:00 – 14:30 - Lunch time

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

Programming paradigms
for GPU devices

Description:

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 programming language 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.

Skills:

By the end of the course, students will be able to:

- understand the strengths and weaknesses of GPUs as accelerators
- program GPU accelerated applications using both higher and lower level programming approaches
- overcome problems and bottlenecks regarding data movement between host and device memories
- make best use of independent execution queues for concurrent computing/data-movement operations

Target audience:

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

Pre-requisites:

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.