Contribution ID: **31**   Type: **not specified**

Portability, efficiency and maintainability: the case of OpenACC

Friday, 13 February 2015 11:20 (20 minutes)

An increasing number of HPC systems rely on heterogeneous node architectures combining traditional multicore CPUs with power efficient accelerators.

Writing efficient applications for these systems could be cumbersome today, since porting may require code rewriting using new programming languages, such CUDA or OpenCL, threatening maintainability, stability and correctness.

Several innovative programming environments try to tackle this problem; among them OpenACC offers an high-level approach based on directives: porting applications to heterogeneous architectures "simply" requires to annotate existing –C, C++ or Fortran –codes with specific "pragma" clauses to identify regions to offload and run on accelerators.

This approach guarantee high portability of codes since support for different accelerators relies on compilers, however one has to carefully assess the relative costs of portability versus computing efficiency. In this presentation we address precisely this issue, using as a test-bench a Lattice Boltzmann code.

We describe our experience in implementing and optimizing a multi-GPU Lattice Boltzmann code using OpenACC and OpenMPI, focusing also on overlapping communications and computation to make the code scaling on a large number of accelerators.

Presenter: Dr CALORE, Enrico (INFN, Sezione di Ferrara)