HPC^2 – a fully-portable CFD solution for heterogeneous computing

X. Alvarez[a], A. Gorobets[a,b], F. X. Trias[a], R. Borrell[c], G. Oyarzun[a]

[a] *Heat and Mass Transfer Technological Center of UPC, Technical University of Catalonia, C/ Colom 11, 08222, Terrassa, Spain*
[b] *Keldysh Institute of Applied Mathematics RAS, 4A, Miusskaya Sq., Moscow, 125047,Russia*
[c] *Barcelona Supercomputing Center (BSC)*

Abstract

The variety of HPC architectures competing in the exascale race makes the portability of codes of major importance. In this work, the HPC^2 (HPC Highly Portable Code) code is presented as a fully portable CFD code for modeling of incompressible turbulent flows, based on an algebraic operational approach. As a result, at the time-integration phase the code relies on a reduced set of only three algebraic operations: the sparse matrix-vector product (SpMV), the linear combination of two vectors (axpy) and the dot product. The discrete operators are built using a fully conservative discretization for unstructured meshes. This algebraic approach combined with a hybrid parallel MPI+OpenMP+OpenCL implementation naturally provides portability across a wide range of computing architectures. Finally, the performance has been studied on more than a hundred of GPUs, mobile ARM-based fused CPU-GPU devices and on multicore CPUs.

Keywords: Parallel CFD, MPI, OpenMP, OpenCL, heterogeneous computing, hybrid architecture

---

The HPC^2 project aims at developing a CFD code suitable for modern supercomputer modeling of incompressible turbulent flows. Massively-parallel computing devices of various architectures are being adopted by the newest supercomputers in order to overcome the actual power constraint in the context of the exascale challenge. With this trend the portability and efficiency of software is of crucial importance. For this purpose, the CFD algorithm is based on an algebraic operational approach using the symmetry-preserving discretization of Navier-Stokes equations on a collocated unstructured grid [1]. As a result, at the time-integration phase the code relies on a reduced set of only three algebraic operations which facilitates porting and optimizing the code for each platform. Furthermore, this software basis is not limited to CFD and can be used in any other operator-based problem.

Special attention has been paid to the code structure which is based on a multi-layer concept and composed of 4 layers in order to ensure modularity and facilitate developers specialization, see Figure 1). Each layer is independent of the upper layers and is able to use modules of the lower layers as black boxes. At the lowest layer algebraic objects (vectors and matrices) and operations

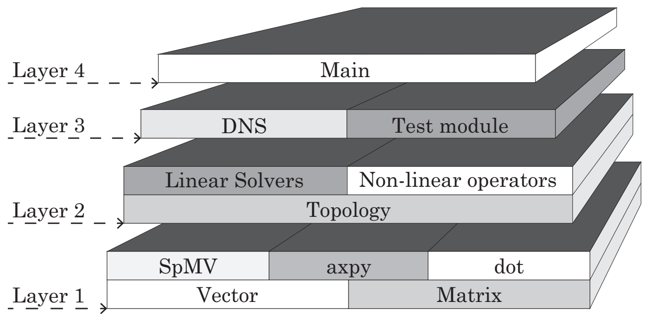

Figure 1: Scheme of the multi-layer code structure.

(SpMV, axpy, dot) are implemented for all required architectures in a portable way using MPI, OpenMP and OpenCL standards. At the layers above, all the mathematical algorithms (e.g. conjugate gradient) and CFD algorithms (e.g. DNS) are easily implemented using the available portable algebraic objects. Furthermore, the complex object-oriented implementation of the preprocessing stage which includes processing of mesh data, construction of discrete operators etc. is detached from the time-integration core. In this way various external simulation codes can be coupled with HPC^2 via a simple file format where the discrete operators are given in the algebraic form.

Acknowledgements: The work has been financially supported by RSF, project 15-11-30039 (OpenCL implementations). The following HPC systems we used for tests: MareNostrum and Mont-Blanc (BSC), K-100 (KIAM), MVS-10P (JSCC RAS), Lononosov-2 (MSU). The authors thankfully acknowledge these institutions.

References

[1] F. X. Trias, O. Lehmkuhl, A. Oliva, C.D. Pérez-Segarra, R.W.C.P. Verstappen, Symmetry-preserving discretization of Navier-Stokes equations on collocated unstructured meshes, Journal of Computational Physics 258 (2014) 246–267.