HPC Comparison of Hypre vs Pstream as external linear algebra library for OpenFOAM

Ivan Spisso[1], Giorgio Amati[2]

[1] *SuperComputing Applications and Innovation Department, Cineca, Via Magnanelli 6/3, 40033, Casalecchio di Reno, Bologna, Italy, i.spisso@cineca.it*

[2] *SuperComputing Applications and Innovation Department, Cineca, Via dei Tizii 6, Roma, 00185 Italy, g.@cineca.it*

Introduction

OpenFOAM acts as a major player in the Open Source CFD arena, due to its flexibility, but on the other side due to its complexity it is hard to define correctly performance figure and scaling. From previous work, it is known that the scalability of the linear solvers restrict the parallel usage up to the order of few thousand of cores [1]. Meantime, the technological trends of exascale HPC is moving towards the use of order of millions of cores [2] as reported in the last Top 500 List [3]. The actual trend of the HPC cluster is to use nodes with high-density of core (more than 50 cores per per CPU) in conjunction with an accelerator (GPUs or Xeon PHI), with more cache level and with the order of O(100,000) of interconnected nodes and O(10,000,000) of available cores [4].

Actual bottleneck and state-of-the-art

Up to date, the well known bottlenecks for a full enabling of OpenFOAM for massively parallel clusters are

✓ the limit in the parallelism paradigm (Pstream Library) [5]

✓ the I/O data storage system [6]

✓ sub-optimal sparse matrices storage format (LDU) format that does not enable any cache-blocking mechanism (SIMD, vectorization) [7]

The standard versions of OpenFOAM, regardless of the different distributions, scales reasonably well up to the order of few thousands of cores. Figure 1, which shows benchmark results of OpenFOAM on standard x-86 HPC architecture (Broadwell, Marconi A1 [8]), gives evidence that using typical industrial size (~27 M of cells) is possible to reach good valid speed-up up to 4000 cores. Figure 2 shows the speed-up of the lid-driven cavity flow with 64 M of cells compared with three different architectures. BDW and SKL architectures shows excellent speed-up up to 50 nodes, whereas KNL has a poor speed-up but can reach an higher level of cores (256 nodes = 17.408 cores).

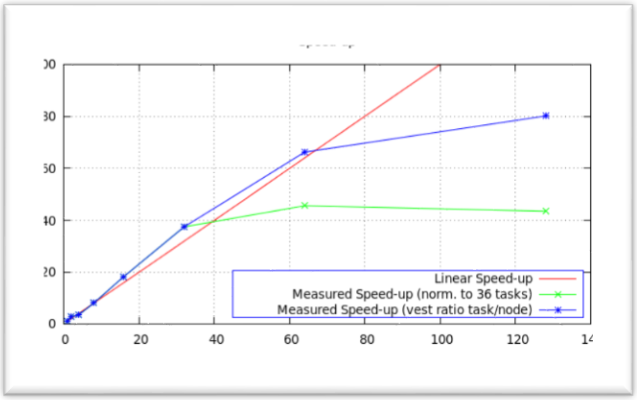

Figure 1: Speed-up of 3D lid driven cavity 27 M of cells, using 36 task per node or looking for the best ratio task-node [4].

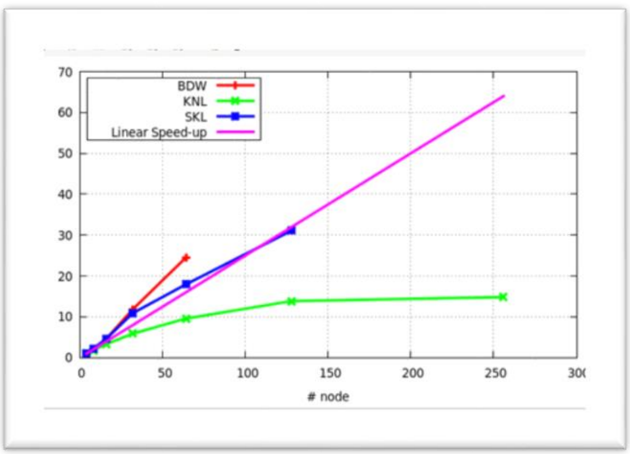

Figure 2: Speed-up of 3D lid driven cavity 64 M of cells, using different HPC architectures.

The linear algebra core libraries, implemented in the Pstream Library, act as a bottleneck for the scalability. In most cases, the memory bandwidth is a limiting factor for the solver. Additionally, global reductions are frequently required in the solvers. The LDU sparse matrix storage format used internally does not enable any cache-blocking mechanism.

Recently, it has been shown that a modified version of OpenFOAM, hardware-bundled, is able to tackle problems of the orders of 1~100 billion of cells, with a parallelism of 10~100 thousands of cores, by modifying the implementation of OpenFOAM's Pstream class and adding a full-hybrid (MPI+OpenMP) version [7]. This version is a custom version, not public available.

Proposed Work(plan)

The present work aims to compare HYPRE as linear algebra package against the standard Pstream library used in OpenFOM.
Livermore's HYPRE library of linear solvers makes possible larger, more detailed simulations by solving problems faster than traditional methods at large scales. It offers a comprehensive suite of scalable solvers for large-scale scientific simulation, featuring parallel multigrid methods for both structured and unstructured grid problems. The HYPRE library is highly portable and supports a number of languages [8].

This is the whole proposed work plan:

- Select one/two simple benchmarks (lid-driven cavity flow or similar with different size, initial conditions, etc. etc.) to be used as bench test for linear solver algebra (GAMG vs PCG)
- Run with OpenFOAM and HYPRE on different architectures and parallelism
  - Profiling of OpenFOAM and HYPRE on HPC cluster: the main blocks to be considered are matrix assembly, the linear solver vs. everything else.
  - Establish of-the-shelf profiling tools able to give useful performance index on different and heterogeneous HPC's ecosystem: Intel tools (and HPCtoolkit)
  - Spotting performance bottlenecks on selected benchmarks
- Interface *ldutocsr* format (*ldutohypre*): prototype stage to add an interface to transcribe the LDU format into another sparse matrix that is supported by an external solver package such as hypre or petsc, as CSR format. This temporarily interface could be used to benchmark improvements possible by use of one of these external solver packages.
- Quantify the benefit of using external libraries as linear solver algebra vs. OpenFOAM standard. When the interface *ldutohypre/ldutocsr* is used, most likely the memory allocation has to be duplicated. For the solution of the PDE, the format transfer from/to LDU to/from CSR format is necessary. A profiling will be very useful to quantify thr time spent in the format transfer.
- Rewrite of OpenFOAM matrix storage classes: This stage would involve a rewrite of the OpenFOAM matrix storage classes (*crsMatrix* instead of *lduMatrix*) to accommodate a lightweight interface to HYPRE and/or petsc.

When the last stage has been taken, OpenFOAM will be able to automatically benefit from solver improvements coming from these projects.
The aforementioned work-plan is an extended work, which is envisaged to be pushed forward with support from academia and OpenFOAM developers. The preliminary/on-going part of the work will be presented at the conference.

References

1. M. Culpo, Current Bottlenecks in the Scalability of OpenFOAM on Massively Parallel Clusters, PRACE White Paper, August 2012.
2. HPC Enabling of OpenFOAMfor CFD applications, 6-8 April 2016, Casalecchio di Reno, Bologna, Italy
3. https://www.top500.org/lists/2018/06/
4. I. Spisso, G. Amati, Towards exafoam: Deployment of unstructured mesh implict CFD code suitable for exascale, Tech. Rep., 12th OpenFOAM Workshop, Exter, 23-27 Juky 2017.
5. I. Spisso, G. Amati, V. Ruggiero, C. Fiorina: Porting, optimization and bottleneck of OpenFOAM in KNL environment, Intel eXtreme Performance Users Group (IXPUG) Europe Meeting, CINECA, Casalecchio di Reno, 5-7 March 2018
6. K. Meredith, M. Olsen, N. Podhorszki: Integrating ADIOS into OpenFOAM for Disk I/O, 4th Annual OpenFOAM User Conference 2016, Cologne, Germany.
7. C. Fiorina, G. Amati, V. Ruggiero, I. Spisso: Evaluation of the Intel Xeon Phi performances for high fidelity nuclear applications, SHAPE white paper, unpublished.
8. Marconi USER GUIDE
9. Pham Van Phuc, Shimizu Corporation, Fujitsu Limited, Riken: Large Scale Transient CFD Simulations for Building using OpenFOAM on a World’s Top-class Supercomputer, 4th Annual OpenFOAM User Conference 2016, Cologne, Germany.
10. Hypre: Scalable Linear Solver and Multigrid Methods