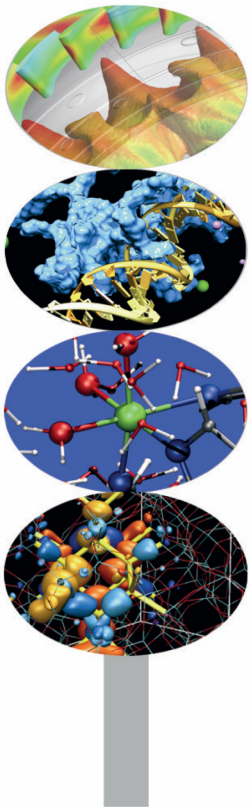
CINECA SCAI
SuperComputing Applications and Innovation

Cineca TRAINING High Performance Computing 2016

HPC enabling of OpenFOAM® for CFD applications



Towards the exascle: OpenFOAM® perspective

Ivan Spisso, Giorgio Amati

06-08 April 2016, Casalecchio di Reno, BOLOGNA.
SuperComputing Applications and Innovation (SCAI) Department, CINECA

PRACE CINECA

Overview of OpenFOAM

1. Parallel aspects and performance
2. Suggestions / Best Practices
3. Further Work

Running in parallel 1/2

- OpenFOAM is first and foremost a *C++ library* used to solve in discretized form systems of Partial Differential Equations (PDE).
- The "Engine" of OpenFOAM is the Numerical Method. To solve equations for a continuum, OpenFOAM uses a numerical approach with the following features: segregated, iterative solution, finite volume method, co-located variables, equation coupling.
- The method of parallel computing used by `OpenFOAM` is based on the standard Message Passing Interface (MPI) using the strategy of domain decomposition.

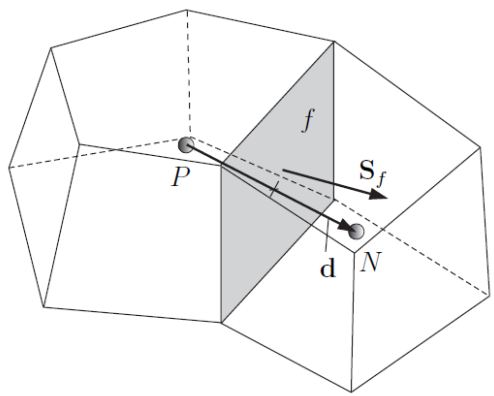


Figure: Finite Volume Discretization

Running in parallel 2/2

- The geometry and the associated fields are broken into pieces and allocated to separate processors for solution.
- A convenient interface, `Pstream`, is used to plug any Message Passing Interface (MPI) library into OpenFOAM. It is a light wrapper around the selected MPI Interface

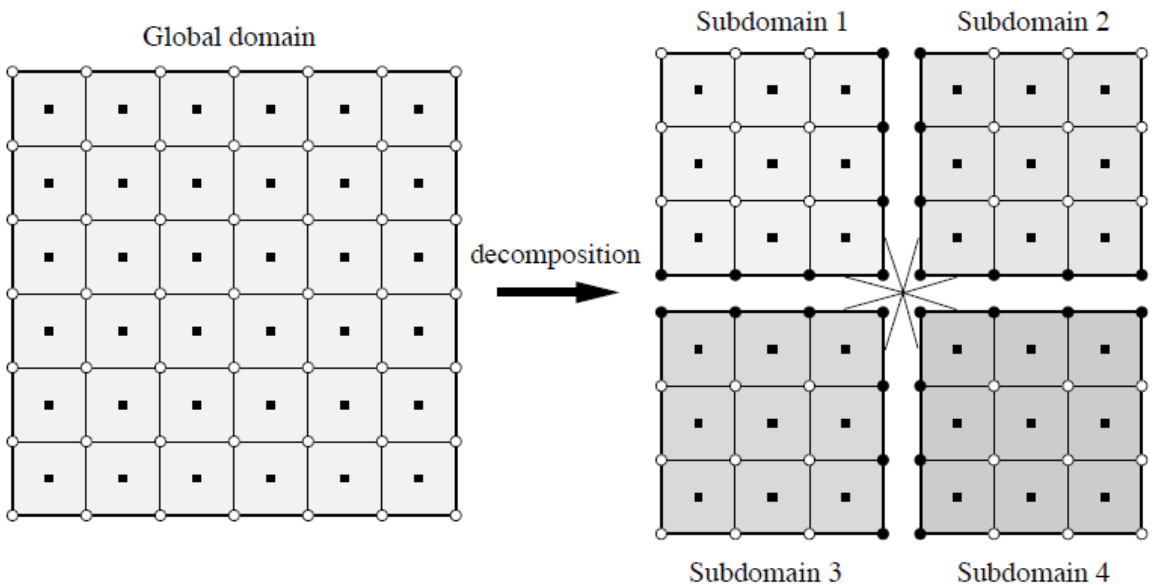


Figure: Zero Layer Domain Decomposition

Performance and bottlenecks

An analysis has been done in the framework of PRACE 1IP to study the current bottlenecks in the scalability of OpenFOAM on Massively parallel clusters.

- OpenFOAM scales reasonably well up to thousands of cores, upper limit $\sim$ 1,000 cores.
- An in-depth profiling identified the calls to the `MPI_AllReduce` function in the linear algebra as core libraries as the main communication bottleneck
- A sub-optimal performance on-core is due the sparse matrices storage format that does not employ any cache blocking.

M. Culpo, Current Bottlenecks in the Scalability of OpenFOAM on Massively Parallel Clusters,

PRACE White Paper, available on-line at www.prace-ri.eu

PRACE White Paper

http://www.prace-ri.eu/application-scalability/

P. Dagna, J.Hertzer: Evaluation of Multi-threaded OpenFOAM Hybridization for Massively Parallel Architectures,

PRACE White Paper, available on-line at http://www.prace-ri.eu/IMG/pdf/wp98.pdf

M. Moylesa, P. Nash, I. Girotto: Performance Analysis of Fluid-Structure Interactions using OpenFOAM

PRACE White Paper, available on-line at http://www.prace-ri.eu

M. Moylesa, P. Nash, I. Girotto: Performance Analysis of Fluid-Structure Interactions using OpenFOAM

PRACE White Paper, available on-line at http://www.prace-ri.eu/IMG/pdf/wp98.pdf

T. Ponweiser, P. Stadelmeyer, and T. Karásek,
Fluid-Structure Simulations with OpenFOAM for Aircraft Design

PRACE white paper, http://www.prace-ri.eu/IMG/pdf/wp172.pdf.

A. Duran, M. S. Celabi, S. Piskin and M. Tuncel: Scalability of OpenFOAM for Bio-medical FLow Simulations,

PRACE White Paper, available on-line at

http://www.prace-ri.eu/IMG/pdf/WP162.pdf

Actual bottlenecks 1/2

Missing for a full enabling on Tier-0 Architecture:
Improve the parallelism paradigm, to be able to scale from the actual $\sim$ 1,000 cores to at least one order of magnitude ($\sim$ 10,000 or 100,000 procs).

Scalability of the linear solvers

- The linear algebra core libraries are the main communication bottlenecks for the scalability
- Whole bunch of MPI_Allreduce stems from an algorithmic constraint and is unavoidable, increasing with the number of cores, . . . unless an algorithmic rewrite is proposed.

Generally speaking, the fundamental difficulty is the inability to keep all the processors busy when operating on very coarse grids.

Actual bottlenecks 2/2

Improve the I/O, which is a bottleneck for big simulation.
For example LES/DNS with hundreds of cores that requires very often saving on disk.

- State of the art: A few million cells is now considered relatively small test case. Cases of this size will not scale usefully beyond 1K cores and there is not much to be done to improve this.
- Where we are looking at is radical scalability $\Longrightarrow$ The real issues are in the scaling of cases of 100's of millions of cell on 10K+ cores.

Overview of OpenFOAM

Best Practices ···

Tune your application on HPC enviroment

- strong scaling $\Longrightarrow$ how the solution time varies with the number of processors for a fixed total problem size
- The performance results vary depending on different parameters including the nature of the tests, the solver chosen, the number of cells per processors, the class of cluster used, choice of MPI distributions, etc
- Choose the linear system solvers: use the geometrical multi-grid solver (GAMG) for very large problems [1]. The GAMG solver can often be the optimal choice, particularly for solving the pressure equation
- Compile OpenFOAM in SP (Single Precision), if possible for your application.

[1] W. Briggs, V. Henson, and S. McCormick, *A Multigrid Tutorial: Second Edition* Society for Industrial and Applied Mathematics, 2000.

· · · Best Practices

Tune your application on HPC enviroment

- Rule of the thumb: tests cases up to tenths of millions of cells scales well with orders of thousands of cores
- try to reduce the communication time when running your test-cases
- increase (as much as possible) the number of cells per processors to find out the optimal number for your application in the selected cluster
- check the memory usage. Now you have a lot of memory per node, try to use it
- Extrapolate the optimal number of cells per cores for your test-case on the selected hardware

Overview of OpenFOAM

(Suggested) Further work · · ·

Strategies for improvement the scalability of the code

- Modified implementation of OpenFOAM's Pstream class: Ponweiser et al. [2] has changed the implementation of OpenFOAM's inter-process communication class from point-to-point MPI communication routines to collective MPI communication routines. The scaling characteristic could be enhanced considerably. (good scalability has been measured up to 4096 cores).

[2] T. Ponweiser, P. Stadelmeyer, and T. Karásek,
Fluid-Structure Simulations with OpenFOAM for Aircraft Design
PRACE white paper, www.prace-ri.eu.

· · · (Suggested) Further work

Strategies for improvement the scalability of the code

- algorithmic rewrite to reduce the communication: the GAMG solver need to be changed to agglomerate processors as well as cells as the mesh is coarsened to reduce the local transfer costs by matching the communications structure of the machine.
- use the accelerators today available on the HPC cluster (GPU enabling) or Intel-Phi (hybridization with OpenMP)
- co-design, that is design closely integrated to the systemware development: PGAS (Partitioned Global Address Space), UPC (Unified Parallel C), Chapel, X10 and Fortress language developments

All these strategies requires a significant effort and very good OpenFOAM (C++) programmer