Enhancing Computational Aero-Acoustic Processes for Ground Vehicles using Scale Resolving Open Source CFD

Torbjörn Larsson, Johan Hammar (Creo Dynamics AB)
Jing Gong, Michaela Barth, Lilit Axner (KTH-PDC)

OpenFOAM Workshop 2018, Shanghaii, China – June 24-29, 2018

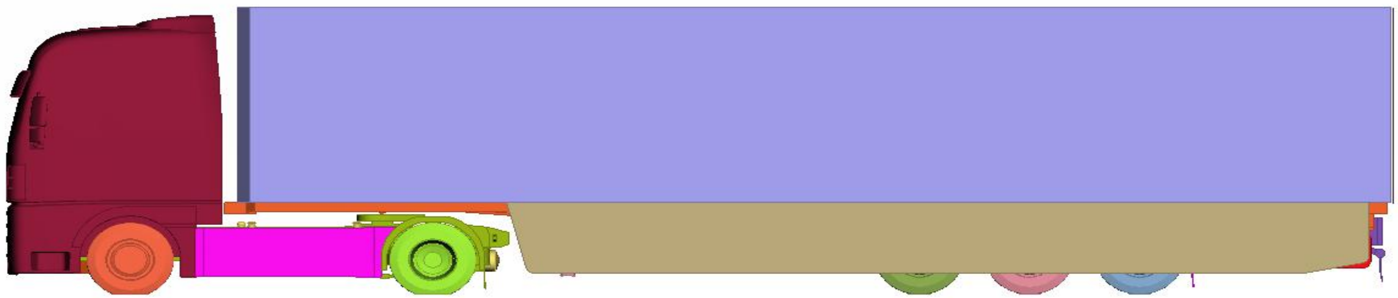
Outlines

- Introduction
- Benchmarking study
- Open source CFD workflow
- OpenFOAM utilities and solvers
- Results
- Conclusions and discussions

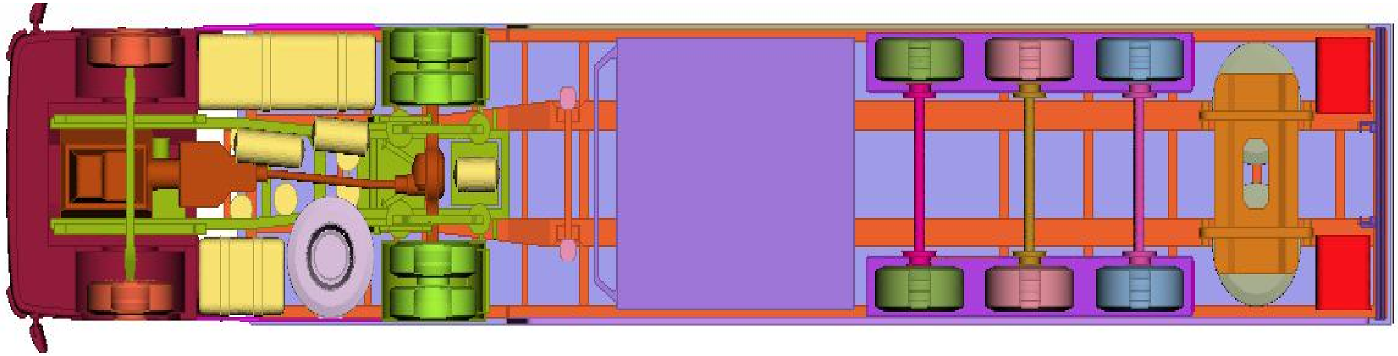
Introduction

- primarily devoted to the development of templates and "recipes" for automated CAD handling, parallel meshing, solving and post-processing that were tailored towards the chosen application
- studied the parallel implementation of code and execution on large parallel clusters for real industrial applications

Bechmaring study



Truck-Trailer CFD model, side view



Truck-Trailer CFD model, bottom view

Open Source CFD Workflow

we set up and analyse an automated process that, using prepared CAD surfaces as input, generates a high resolution CFD mesh fulfilling predefined quality metrics. The process also performs a steady state RANS (Reynolds-averaged Navier-Stokes) simulation, and extracts results and post-processing data. All these are steps implemented in parallel on distributed compute nodes without the need for any intermediate input/output (I/O) or data transfers.

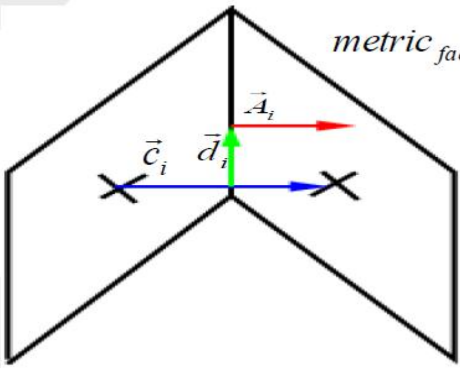
SnappyHexMesh

- **geometry**
  - Definition of geometrical entities
- **castellatedMeshControls**
  - Definition of features, surface and volume refinements
- **snapControls**
  - Definition of parameters for surface snapping
- **addLayersControls**
  - Definition of parameters for boundary layer meshing
- **meshQualityControls**
  - Definition of mesh quality metrics

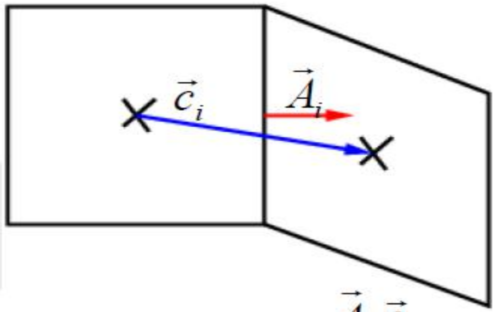
SnappyHexMesh

Face skewness is calculated as the distance from the face centre to the cell-centre to cell-centre face intersection point normalised by the distance from centroid of the cell to centroid of the adjacent cell

$$metric_{face,skewness} = \frac{|\vec{d}_i|}{|\vec{c}_i|}$$

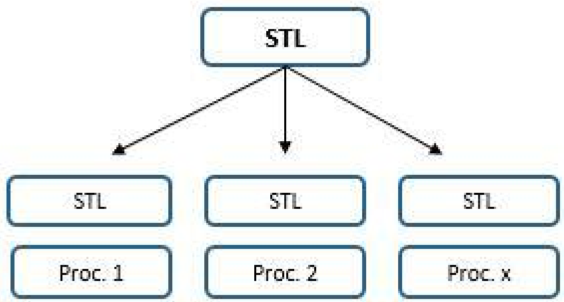


Face orthogonality is calculated as the normalised dot product of the face area vector with a vector from the centroid of the cell to centroid of the adjacent cell



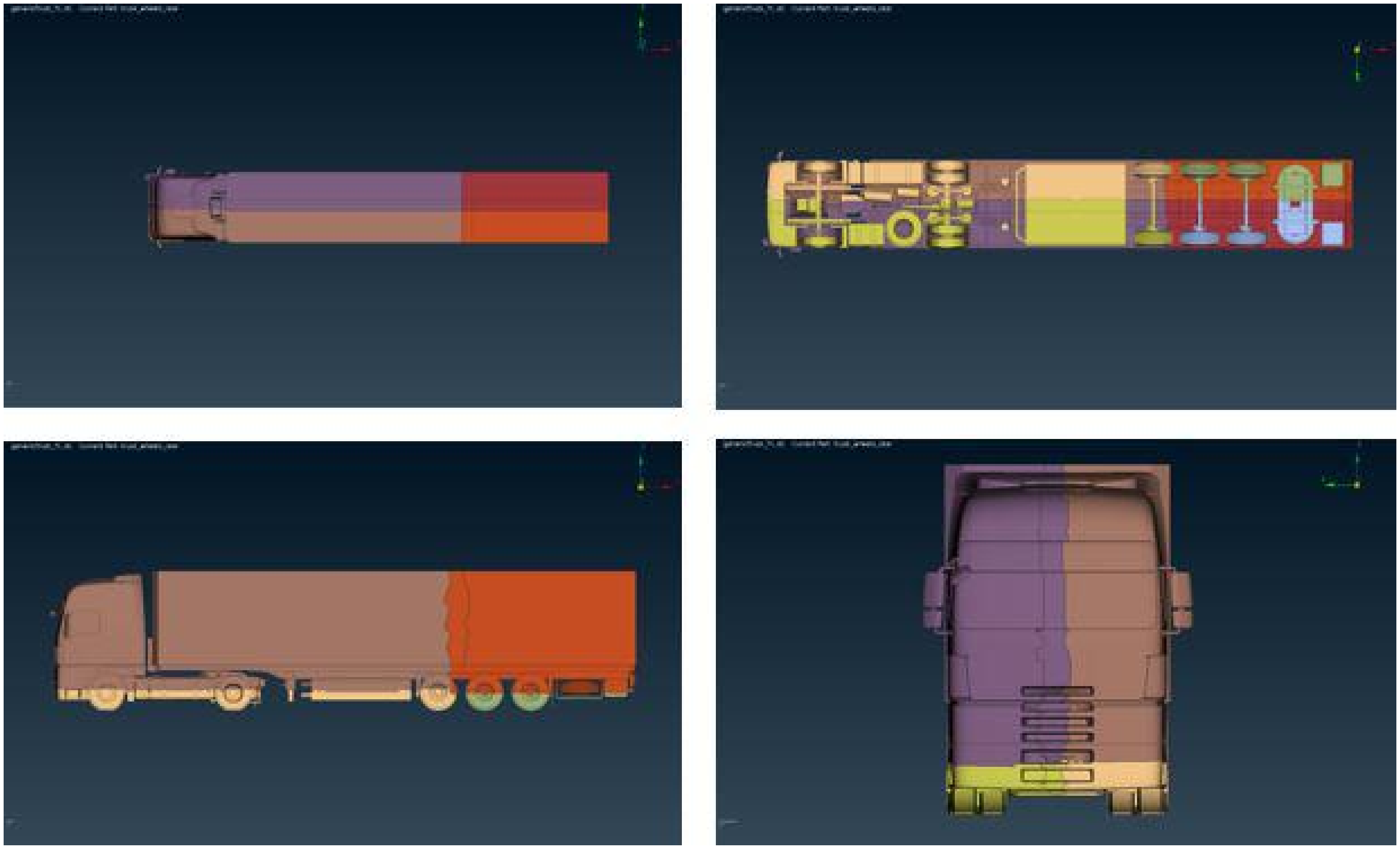
$$metric_{face,orthogonaity} = \frac{\vec{A}_i . \vec{c}_i}{|\vec{A}_i||\vec{c}_i|}$$

SnappyHexMesh

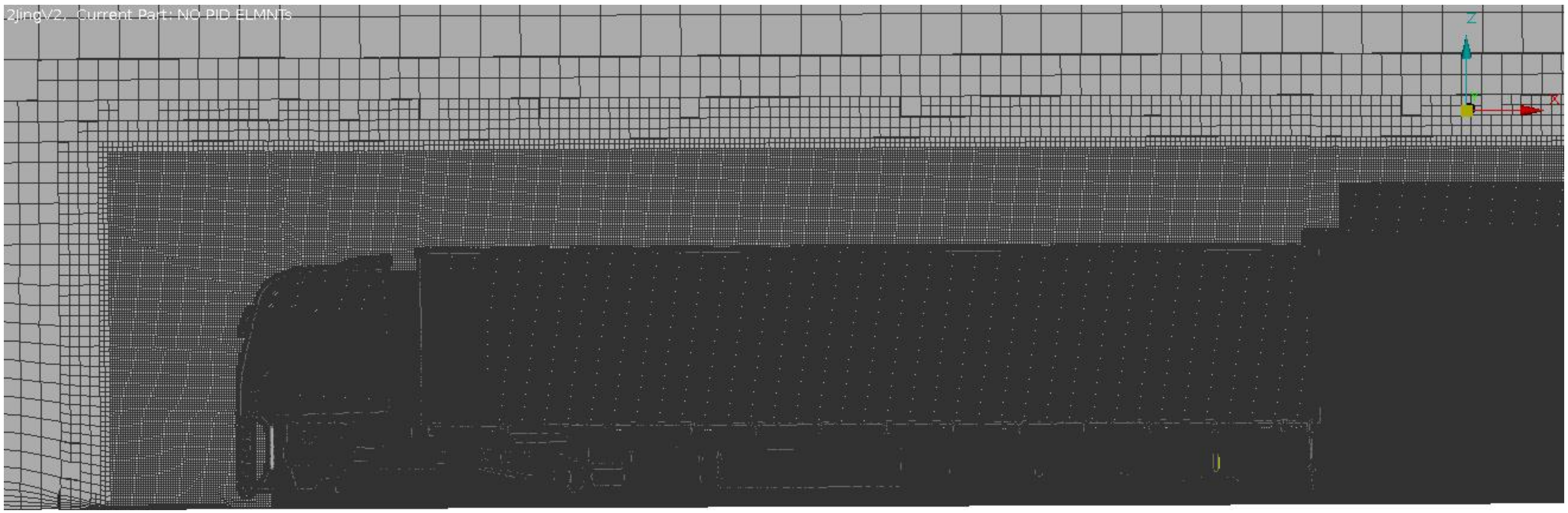


Standard (left) vs. Distributed (right) STL input

SnappyHexMesh



SnappyHexMesh



OpenFoam Solver (simpleFoam)

The Navier-Stokes equations for the incompressible flows are written as

$$\nabla \cdot \mathbf{v}=0$$

$$\partial \mathbf{v} \partial t+\mathbf{v}\cdot \nabla \mathbf{v}=-\nabla p+\nu \Delta \mathbf{v}$$

· Set the boundary conditions.

· Solve the discretized momentum equation to compute the intermediate velocity field.

· Compute the mass fluxes at the cell faces.

· Solve the pressure equation and apply under-relaxation.

· Correct the mass fluxes at the cell faces.

· Correct the velocities on the basis of the new pressure field.

· Update the boundary conditions.

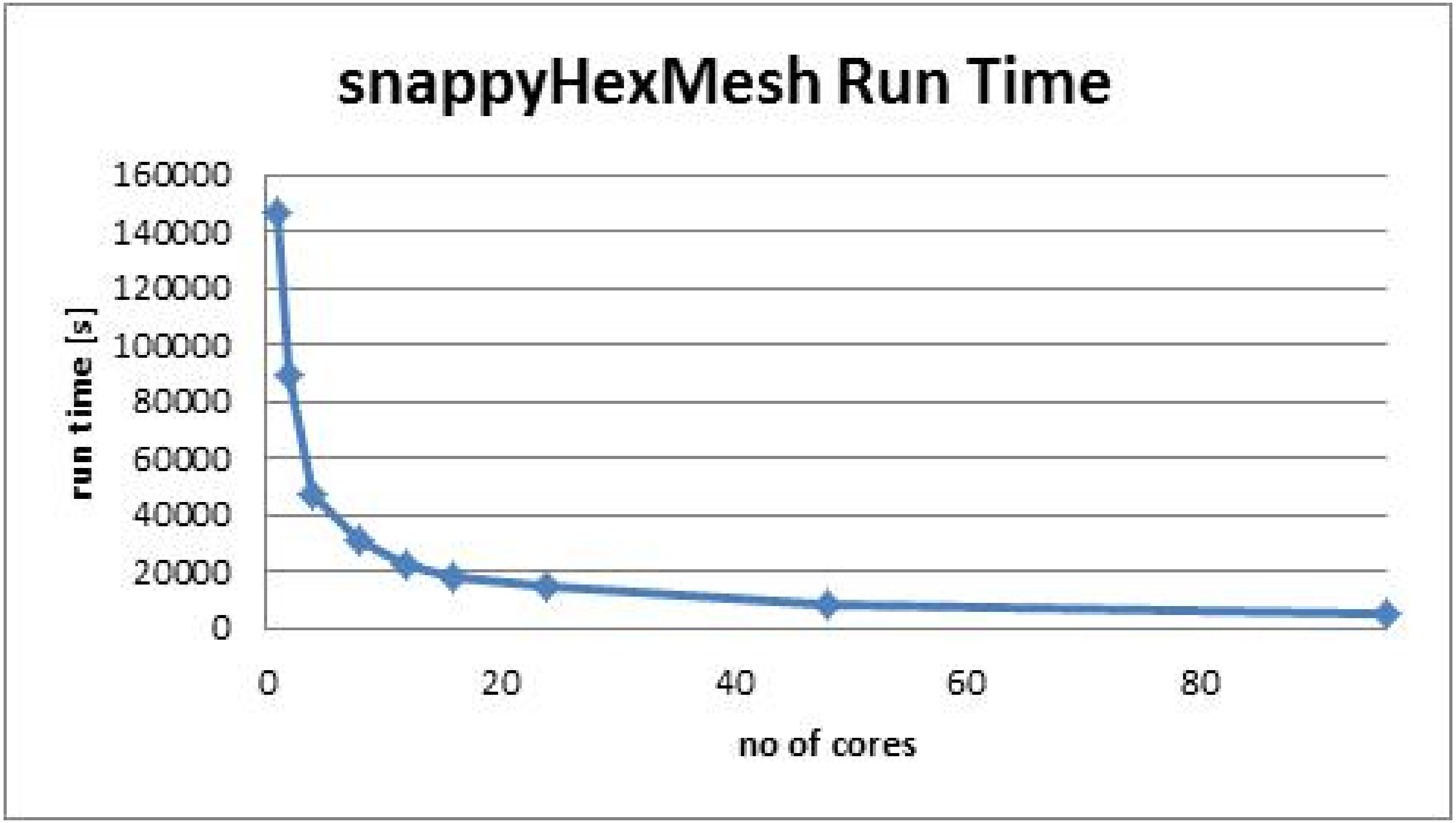
· Repeat till convergence.

Results

Hardware:

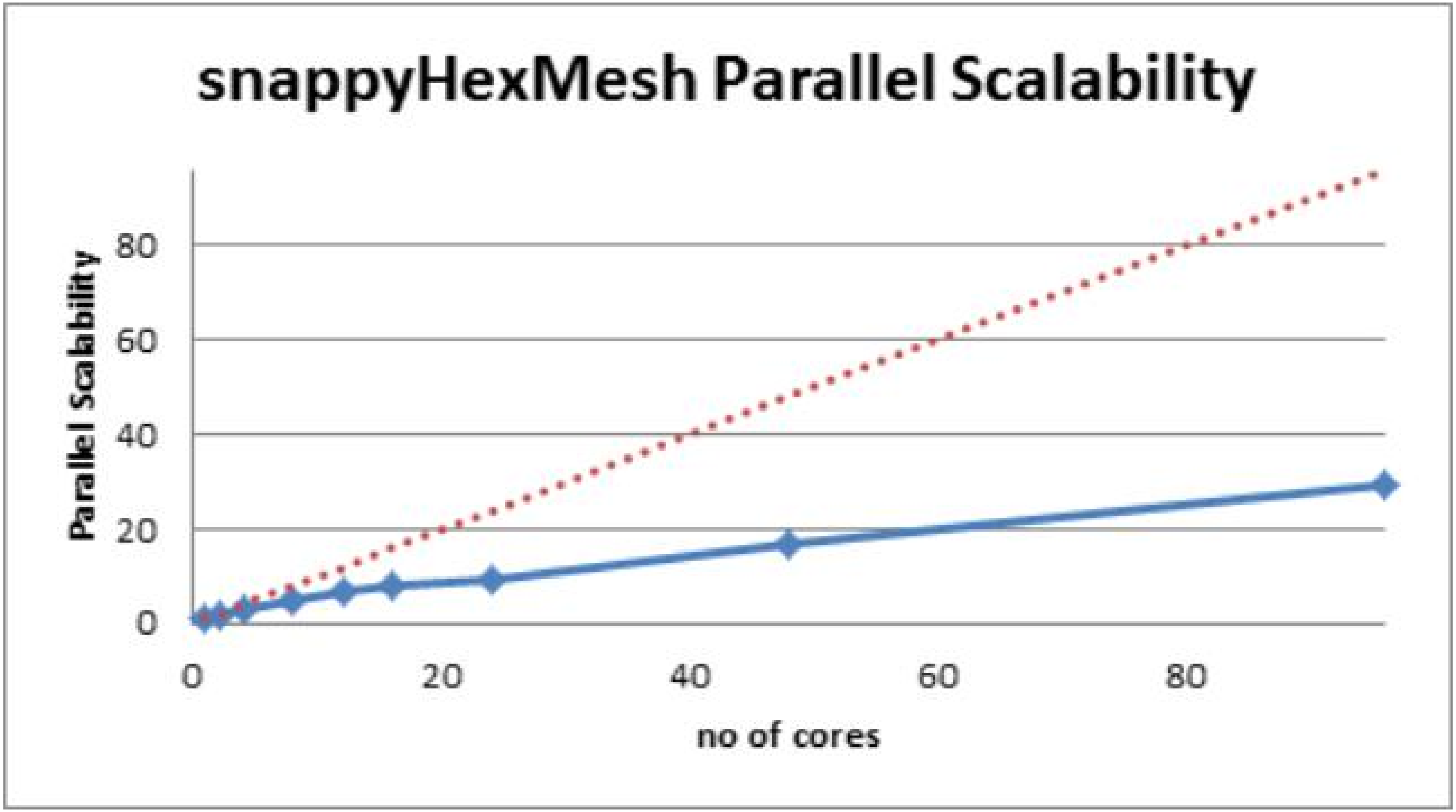
- Beskow: Cray XC40 system with Intel Haswell processors and the Cray Aries interconnection. It consists of 1676 compute nodes, each of which consists of 32 Intel Xeon E5-2698v3 cores
- MareNostrum: 36 racks dedicated to calculations. These racks have a total of 48,448 Intel SandyBridge cores with a frequency of 2.6 GHz and 94.625 TB of total memory

Results

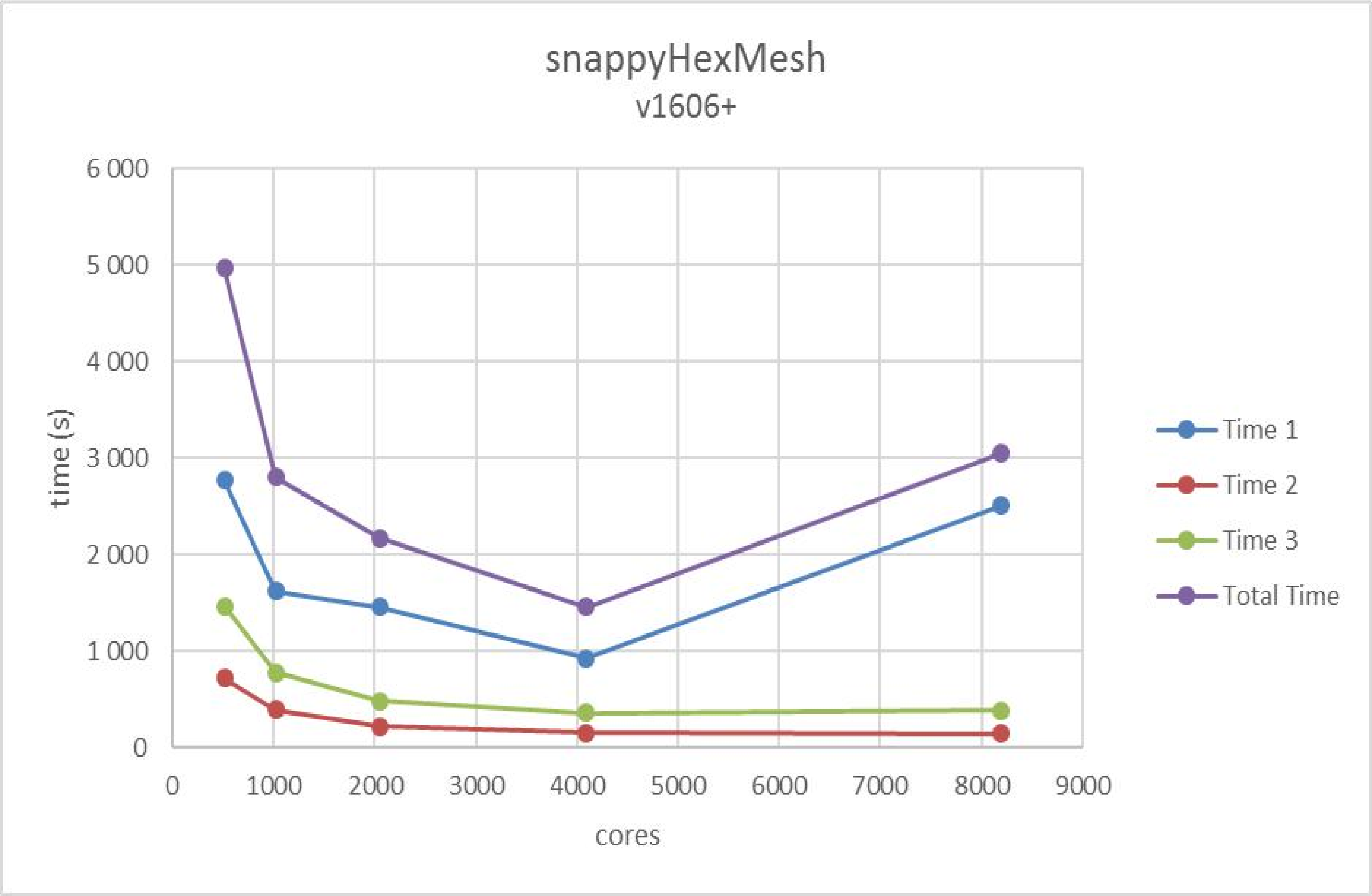


snappyHexMesh Run Time
run time [s]
160000
140000
120000
100000
80000
60000
40000
20000
0
0
20
40
60
80
no of cores

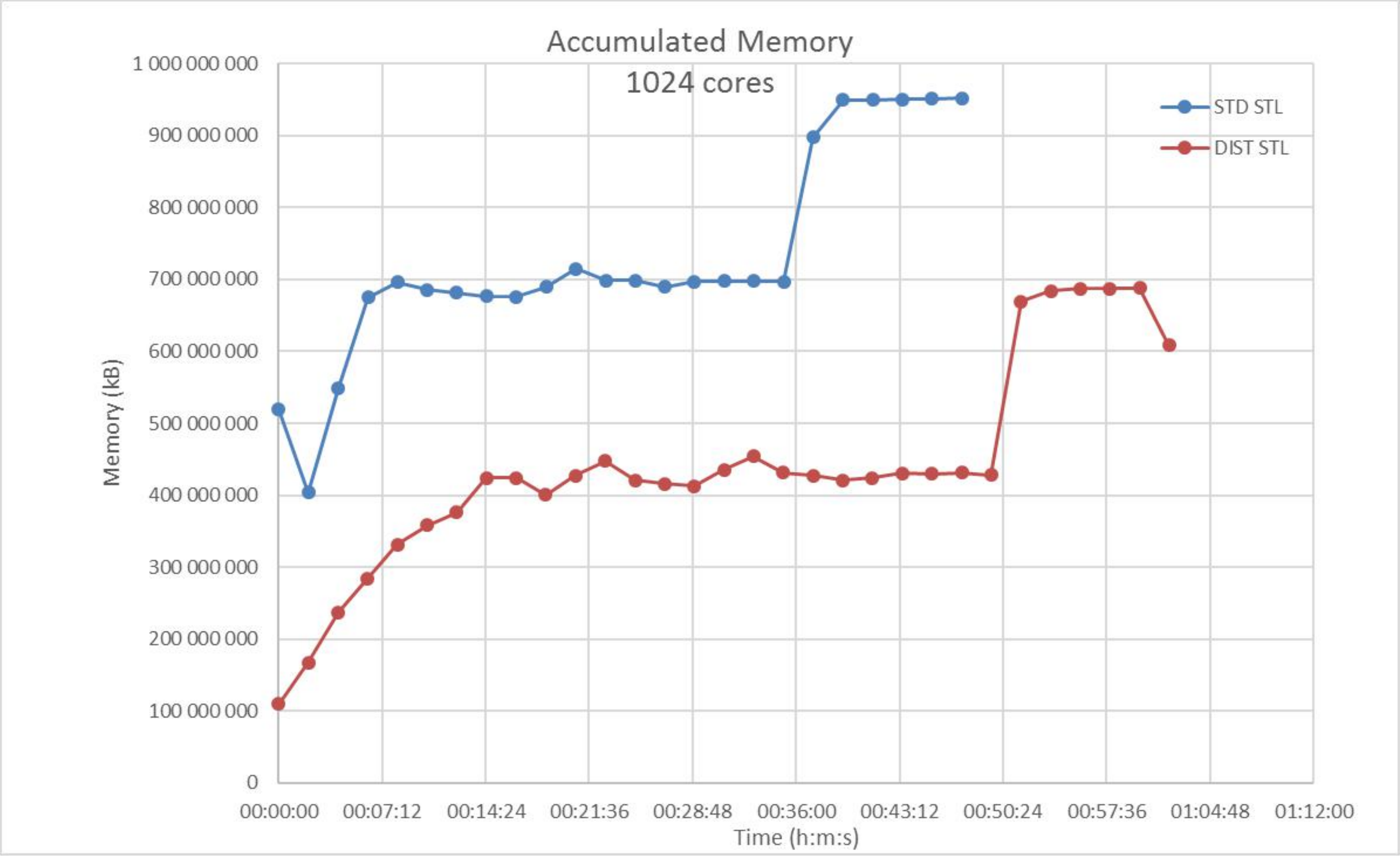
Results


snappyHexMesh Parallel Scalability
Parallel Scabbility
0
20
40
60
80
0
20
40
60
80
no of cores

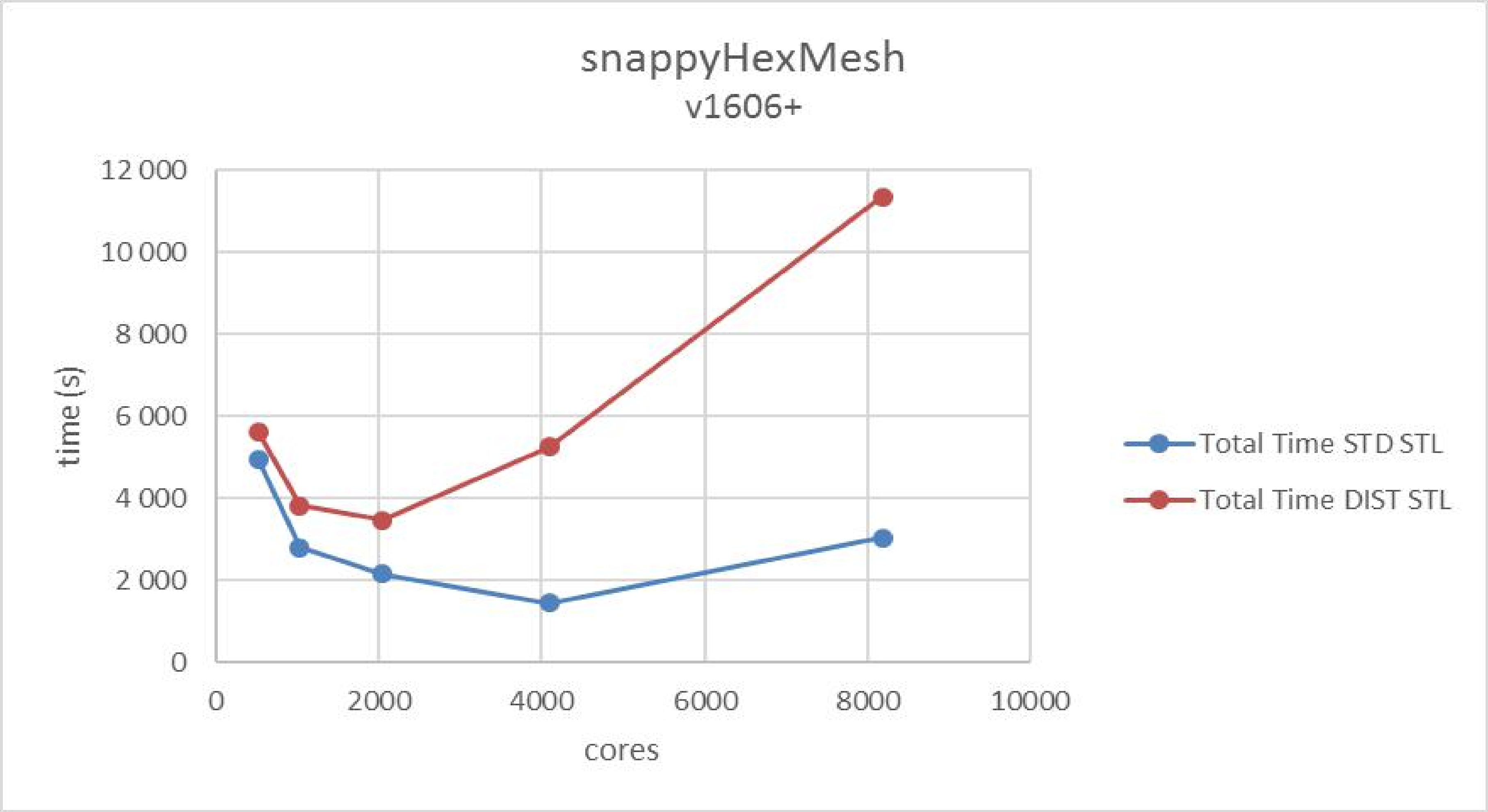
Results



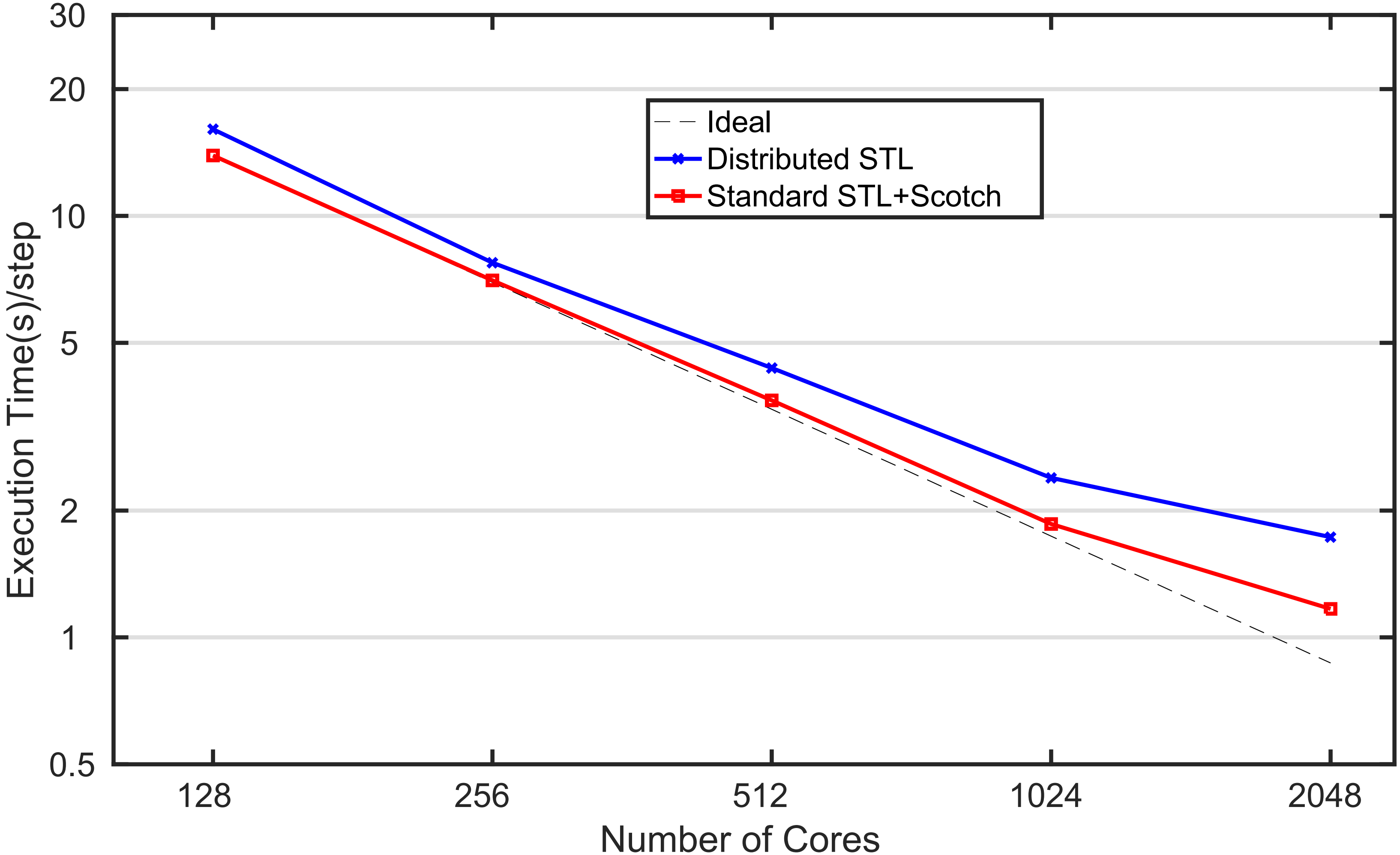
snappyHexMesh
v1606+
6 000
5 000
4 000
3 000
2 000
1 000
0
time (s)
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
cores
Time 1
Time 2
Time 3
Total Time



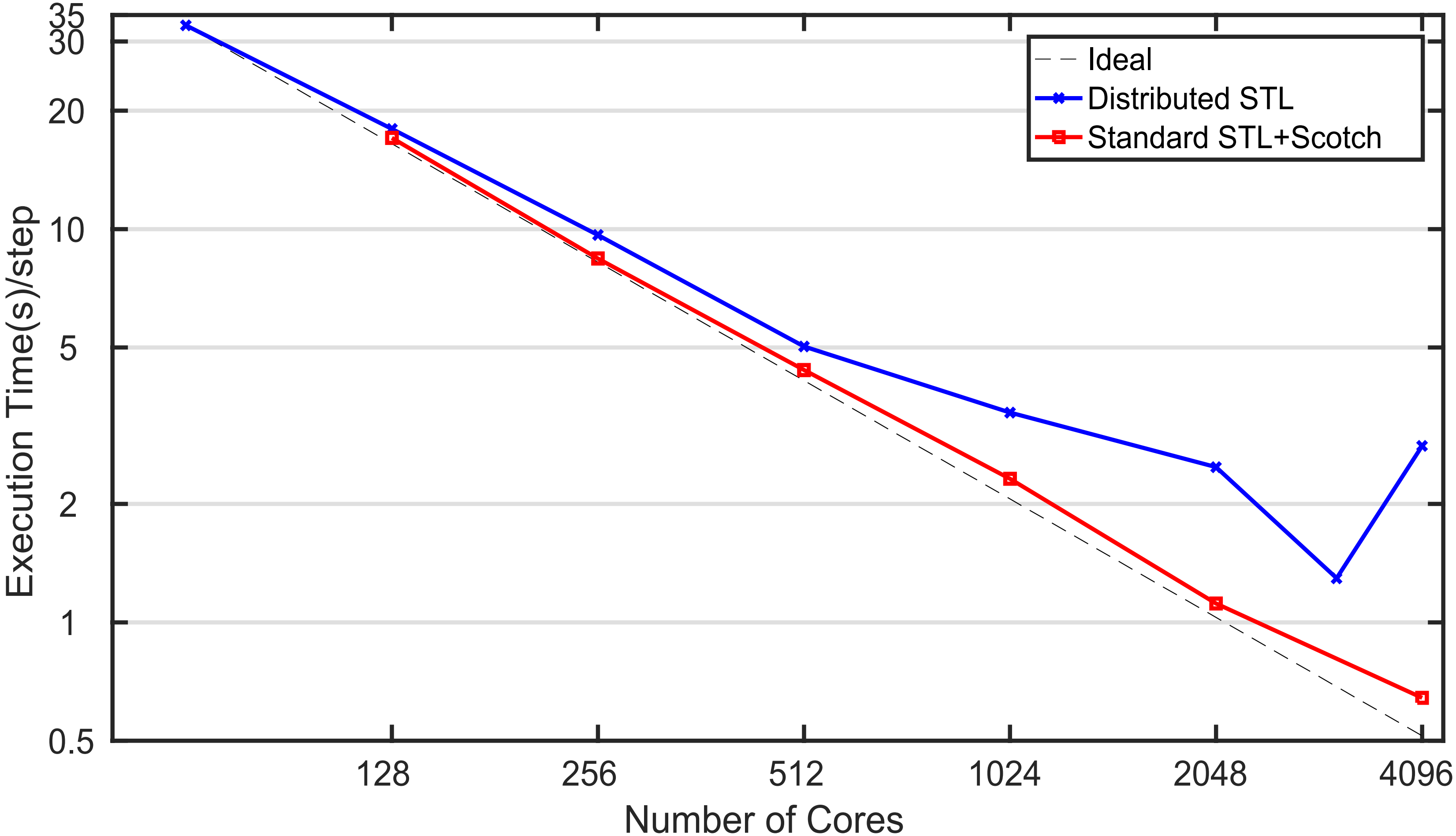
Accumulated memory of snappyHexMesh running on 1024 cores, large case



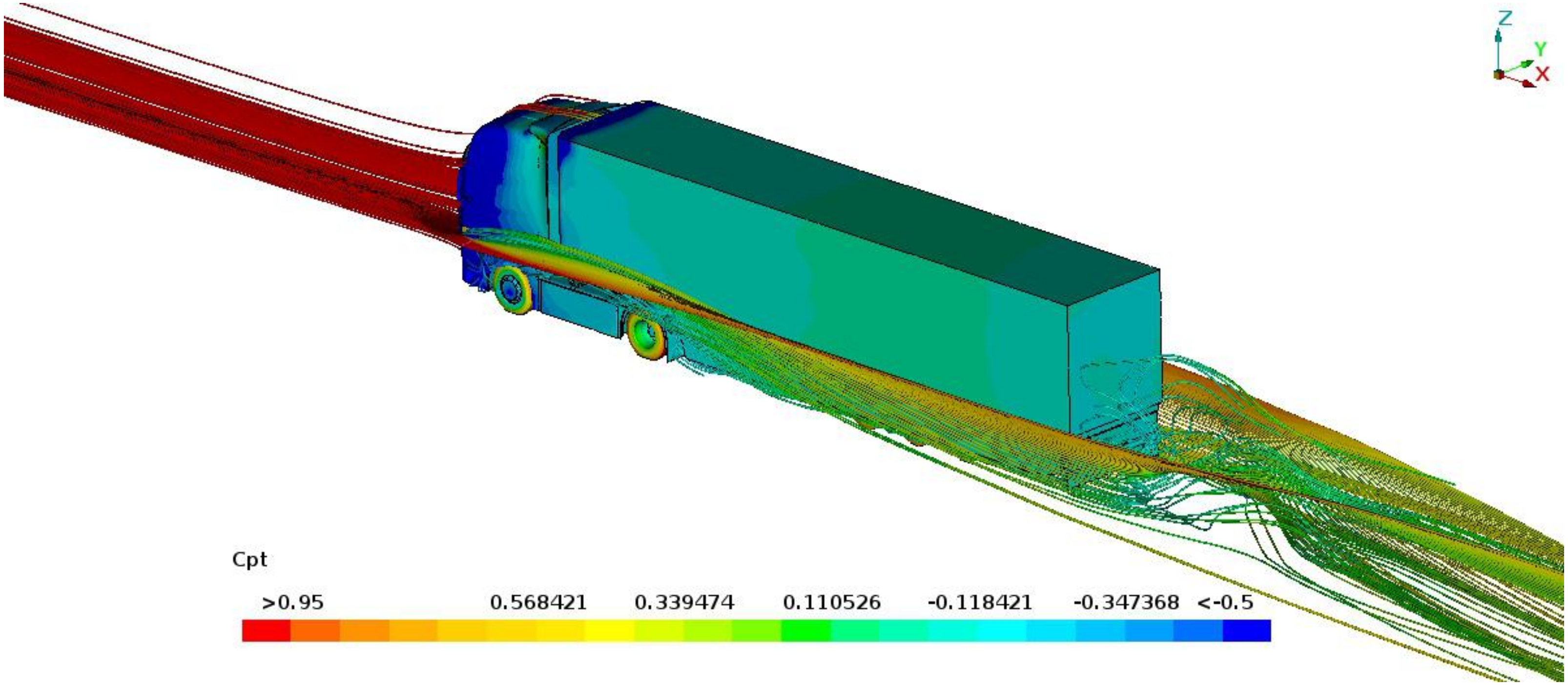
Run time performance of snappyHexMesh STD vs. DIST STL, large case



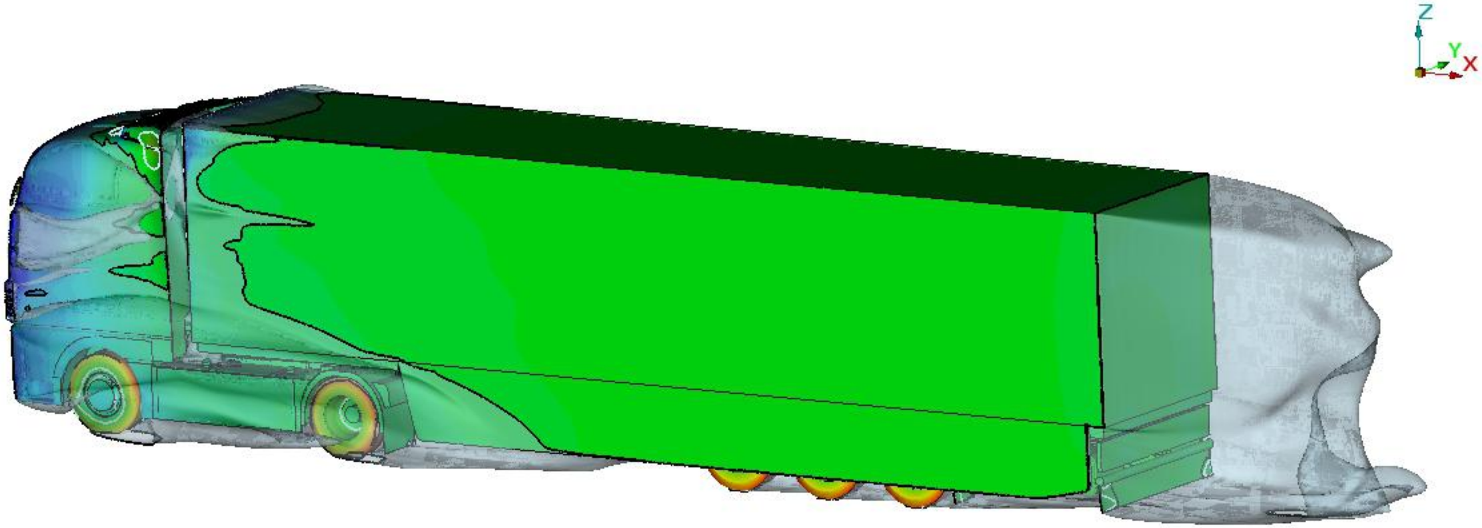
The performance results on MareNostrum



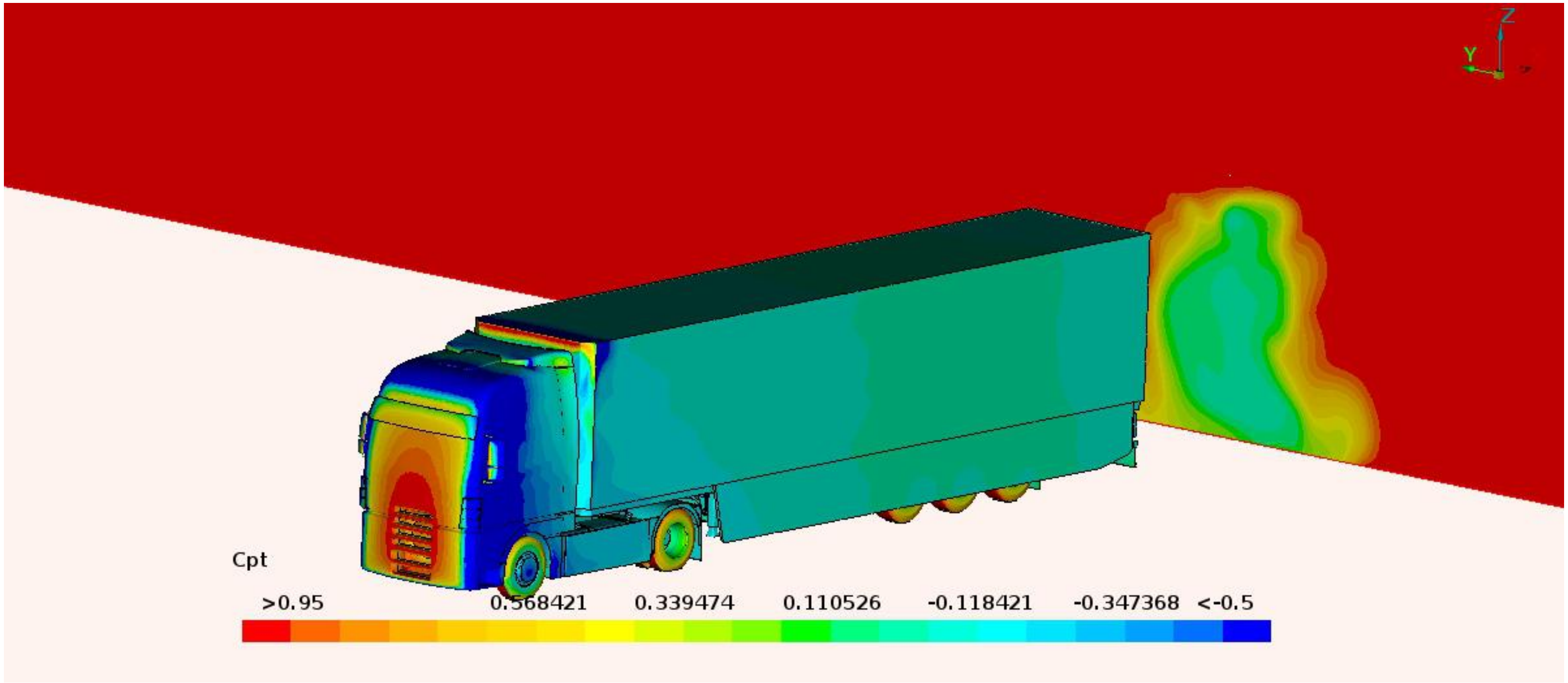
The performance results on Beskow



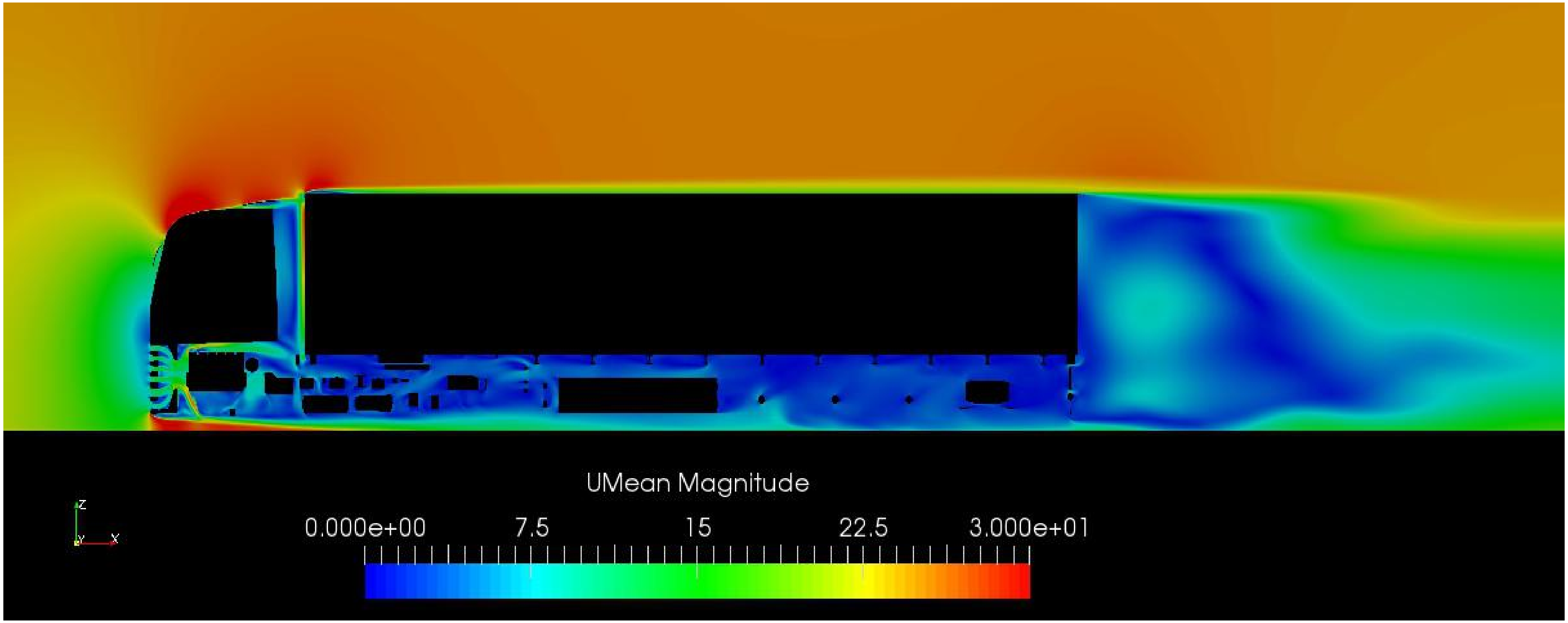
Flow path lines rendered by total pressure



Iso-contours of total pressure



Vehicle surfaces rendered by static pressure and plane rendered by total pressure



Vehicle centreline plane rendered by mean velocity magnitude

Conclusions and discussions

- a (semi) automated CFD simulation process tailored towards aerodynamics predictions for heavy duty semi-trailers was developed.
- A variety of benchmarks related to parallel meshing and flow solving using *OpenFoam* were performed.
- The memory usage and parallel scalability were monitored revealing performance deficits and weaknesses, particularly related to parallel mesh generation on many cores.
- For efficient simulations of large problems requiring many thousands of computational cores there are still critical bottlenecks and deficiencies in current *OpenFoam* distributions that need to be addressed.