On the feasibility of overnight industrial high-fidelity simulations of CSP technologies on modern HPC systems

A.Alsalti-Baldellou[1,2] G.Colomer[3] J.A.Hopman[1] X.Álvarez-Farré[4]
A.Gorobets[5] F.X.Trias[1] C.D.Pérez-Segarra[1] A.Oliva[1]

[1]Heat and Mass Transfer Technological Center, Technical University of Catalonia
[2]Department ICEA, University of Padova, Italy
[3]Termo Fluids SL, http://www.termofluids.com/
[4]High-Performance Computing Team, SURF, The Netherlands
[5]Keldysh Institute of Applied Mathematics of RAS, Russia

9th edition of the European Thermal Sciences Conference
June 10-13 2024 – Bled, Slovenia

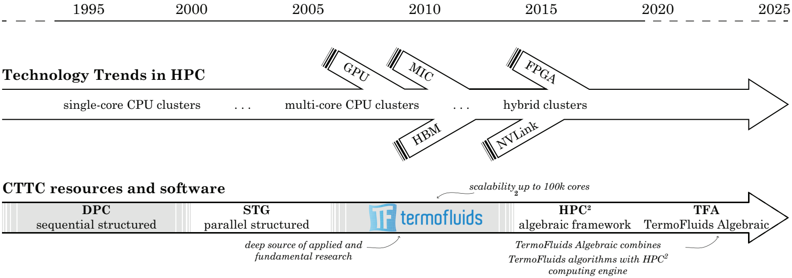
Index

Motivation

Changing landscape...

Rank	System	Cores	Rmax (PFlop/s)	Rpeak (PFlop/s)	Power (kW)
1	**Frontier** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE DOE/SC/Oak Ridge National Laboratory United States	8,699,904	1,194.00	1,679.82	22,703
2	**Supercomputer Fugaku** - Supercomputer Fugaku, A64FX 48C 2.2GHz, Tofu interconnect D, Fujitsu RIKEN Center for Computational Science Japan	7,630,848	442.01	537.21	29,899
3	**LUMI** - HPE Cray EX235a, AMD Optimized 3rd Generation EPYC 64C 2GHz, AMD Instinct MI250X, Slingshot-11, HPE EuroHPC/CSC Finland	2,220,288	309.10	428.70	6,016
4	**Leonardo** - BullSequana XH2000, Xeon Platinum 8358 32C 2.6GHz, NVIDIA A100 SXM4 64 GB, Quad-rail NVIDIA HDR100 Infiniband, Atos EuroHPC/CINECA Italy	1,824,768	238.70	304.47	7,404

Changing landscape, changing codes



Towards performance portability

Stencil-based design

Looping across the mesh performing local operations

Towards performance portability

Stencil-based design

Looping across the mesh performing local operations

- **Pros:** More flexible and compute-intensive, lower memory requirements

Towards performance portability

Stencil-based design

Looping across the mesh performing local operations

- **Pros:** More flexible and compute-intensive, lower memory requirements
- **Cons:** More challenging portability probably relying on "hardware-agnostic" APIs (eg, OpenMP and OpenACC) or libraries (eg, Kokkos and RAJA)

Towards performance portability

Stencil-based design

Looping across the mesh performing local operations

- **Pros:** More flexible and compute-intensive, lower memory requirements
- **Cons:** More challenging portability probably relying on "hardware-agnostic" APIs (eg, OpenMP and OpenACC) or libraries (eg, Kokkos and RAJA)

Algebra-based design

Express discrete operators as sparse matrices and fields as vectors

Towards performance portability

Stencil-based design

Looping across the mesh performing local operations

- **Pros:** More flexible and compute-intensive, lower memory requirements
- **Cons:** More challenging portability probably relying on "hardware-agnostic" APIs (eg, OpenMP and OpenACC) or libraries (eg, Kokkos and RAJA)

Algebra-based design

Express discrete operators as sparse matrices and fields as vectors

- **Pros:** All operations reduce to 3 linear algebra kernels generally available in standard libraries (eg, Intel MKL, cuSPARSE, clSPARSE)

Towards performance portability

Stencil-based design

Looping across the mesh performing local operations

- **Pros:** More flexible and compute-intensive, lower memory requirements
- **Cons:** More challenging portability probably relying on "hardware-agnostic" APIs (eg, OpenMP and OpenACC) or libraries (eg, Kokkos and RAJA)

Algebra-based design

Express discrete operators as sparse matrices and fields as vectors

- **Pros:** All operations reduce to 3 linear algebra kernels generally available in standard libraries (eg, Intel MKL, cuSPARSE, clSPARSE)
- **Cons:** Less compute-intensive, higher memory requirements, requires algorithmic reformulation

Algebra-based design

HPC² library

HPC² library

- Sparse linear algebra code
- Modular design ensuring natural portability

X. Álvarez-Farré et al. (2018). "HPC² – A fully-portable, algebra-based framework for heterogeneous computing. Application to CFD" in *Computers & Fluids*.

A. Alsalti-Baldellou et al. (2023). "Exploiting spatial symmetries for solving Poisson's equation" in *Journal of Computational Physics*.

HPC² library

HPC² library

- Sparse linear algebra code
- Modular design ensuring natural portability
- In C++ and currently supporting MPI+OpenMP, CUDA and OpenCL
- Implements a few highly optimized kernels. Namely:
  - Matrix-vector product
  - Linear combination of vectors
  - Dot product of vectors

X. Álvarez-Farré et al. (2018). "HPC² – A fully-portable, algebra-based framework for heterogeneous computing. Application to CFD" in *Computers & Fluids*.

A. Alsalti-Baldellou et al. (2023). "Exploiting spatial symmetries for solving Poisson's equation" in *Journal of Computational Physics*.

HPC² library

HPC² library

- Sparse linear algebra code
- Modular design ensuring natural portability
- In C++ and currently supporting MPI+OpenMP, CUDA and OpenCL
- Implements a few highly optimized kernels. Namely:
  - Matrix-vector product
  - Linear combination of vectors
  - Dot product of vectors
- Als includes specialized kernels and Poisson solvers

X. Álvarez-Farré et al. (2018). "HPC² – A fully-portable, algebra-based framework for heterogeneous computing. Application to CFD" in *Computers & Fluids*.

A. Alsalti-Baldellou et al. (2023). "Exploiting spatial symmetries for solving Poisson's equation" in *Journal of Computational Physics*.

TFA library

TFA library

- Incompressible CFD simulation code
- Fully-conservative discretisation for collocated unstructured grids

---

F.X. Trias et al. (2014). "Symmetry-preserving discretization of Navier-Stokes equations on collocated unstructured meshes" in *Journal of Computational Physics*.

X. Álvarez-Farré et al. (2021). "A hierarchical parallel implementation for heterogeneous computing. Application to algebra-based CFD simulations on hybrid supercomputers" in *Computers & Fluids*.

TFA library

TFA library

- Incompressible CFD simulation code
- Fully-conservative discretisation for collocated unstructured grids
- Algebra-based, formulated in terms of:
  - Sparse matrix-vector product
  - Linear combination of vectors
  - Dot product of vectors

F.X. Trias et al. (2014). "Symmetry-preserving discretization of Navier-Stokes equations on collocated unstructured meshes" in *Journal of Computational Physics*.

X. Álvarez-Farré et al. (2021). "A hierarchical parallel implementation for heterogeneous computing. Application to algebra-based CFD simulations on hybrid supercomputers" in *Computers & Fluids*.

Addressing the challenges

Algorithmic reformulation

Some extra effort is required to reformulate algebraically certain operations applying "locally".
Recently, it was shown how to effectively implement flux limiters and CFL-like time-steps.

N. Valle et al. (2022). "On the implementation of flux limiters in algebraic frameworks" in *Computer Physics Communications*.

F.X. Trias et al. (2023). "An Efficient Eigenvalue Bounding Method: CFL Condition Revisited" in *SSRN*.

Algorithmic reformulation

Some extra effort is required to reformulate algebraically certain operations applying “locally”.
Recently, it was shown how to effectively implement flux limiters and CFL-like time-steps.

Algebra-based boundary conditions

Virtually all boundary conditions can be expressed as an affine transformation:

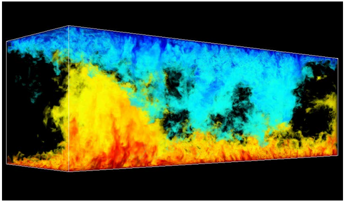
$$\psi_h \to A\psi_h + b_h,$$

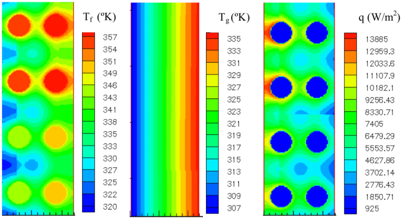
where fluxes are imposed through A and values through b_h

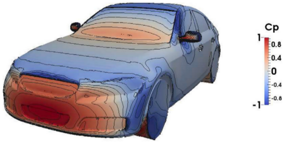
N. Valle et al. (2022). “On the implementation of flux limiters in algebraic frameworks” in *Computer Physics Communications*.

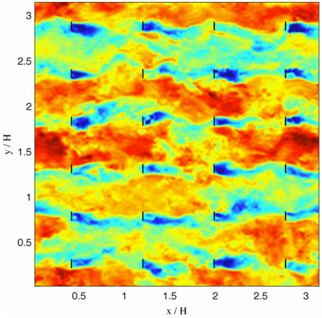
F.X. Trias et al. (2023). “An Efficient Eigenvalue Bounding Method: CFL Condition Revisited” in *SSRN*.

New opportunities: exploiting regular geometries – 1



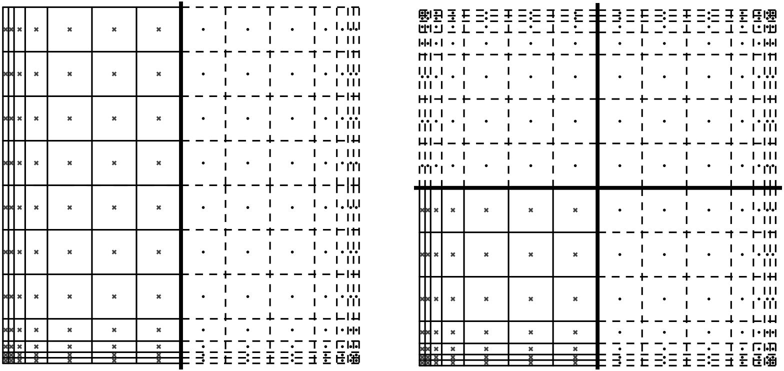






F. Dabbagh et al. (2017) in *Physics of Fluids*
D.E. Aljure et al. (2018) in *Journal of Wind Engineering and Industrial Aerodynamics*
L. Paniagua et al. (2014) in *Numerical Heat Transfer, Part B: Fundamentals*
M. Calaf et al. (2010) in *Physics of Fluids*

New opportunities: exploiting regular geometries – 2

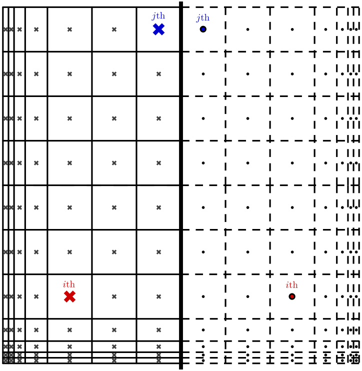


(a) 1 symmetry

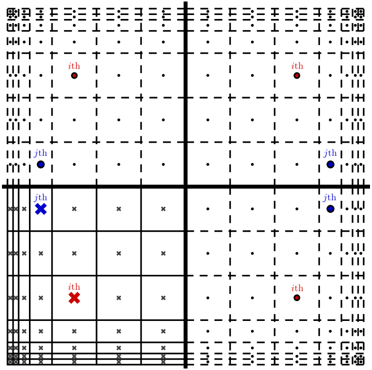
(b) 2 symmetries

Figure: 2D meshes with varying number of symmetries.

New opportunities: exploiting regular geometries – 3



(a) 1 symmetry



(b) 2 symmetries

Figure: "Mirrored" ordering on 2D meshes with a varying no. of symmetries.

New opportunities: exploiting regular geometries – 4

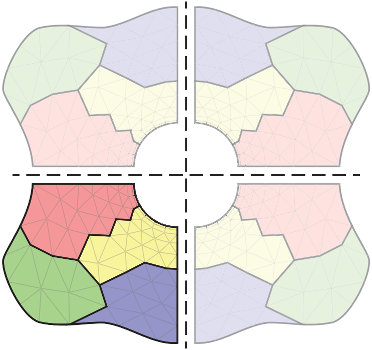


Figure: "Mirrored" partitioning on an unstructured 2D meshes with 2 symmetries.

Computational advantages

On a domain with n_b repeated/mirrored subdomains, virtually all operators satisfy (or a compatible expression):

$$\bar{H} = \mathbb{I}_{n_b} \otimes H, \quad (1)$$

where $\bar{H} \in \mathbb{R}^{n \times n}$ stands for the operator itself and $H \in \mathbb{R}^{n/n_b \times n/n_b}$ for its restriction to the base mesh.

A. Alsalti-Baldellou et al. (2023). "Lighter and faster simulations on domains with symmetries", submitted.

Computational advantages

On a domain with n_b repeated/mirrored subdomains, virtually all operators satisfy (or a compatible expression):

$$\bar{H} = \mathbb{I}_{n_b} \otimes H, \tag{1}$$

where $\bar{H} \in \mathbb{R}^{n \times n}$ stands for the operator itself and $H \in \mathbb{R}^{n/n_b \times n/n_b}$ for its restriction to the base mesh. Then, the standard SpMV on $\bar{H}$:

$$y = \begin{pmatrix} H & & \\ & \ddots & \\ & & H \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_{n_b} \end{pmatrix} \in \mathbb{R}^n, \tag{2}$$

A. Alsalti-Baldellou et al. (2023). "Lighter and faster simulations on domains with symmetries", submitted.

Computational advantages

On a domain with n_b repeated/mirrored subdomains, virtually all operators satisfy (or a compatible expression):

$$\bar{H} = \mathbb{I}_{n_b} \otimes H, \tag{1}$$

where $\bar{H} \in \mathbb{R}^{n \times n}$ stands for the operator itself and $H \in \mathbb{R}^{n/n_b \times n/n_b}$ for its restriction to the base mesh. Then, the standard SpMV on $\bar{H}$:

$$y = \begin{pmatrix} H & & \\ & \ddots & \\ & & H \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_{n_b} \end{pmatrix} \in \mathbb{R}^n, \tag{2}$$

can be replaced with an SpMM on H:

$$(y_1 \dots y_{n_b}) = H(x_1 \dots x_{n_b}) \in \mathbb{R}^{n/n_b \times n_b}. \tag{3}$$

A. Alsalti-Baldellou et al. (2023). "Lighter and faster simulations on domains with symmetries", submitted.

Computational advantages

On a domain with n_b repeated/mirrored subdomains, virtually all operators satisfy (or a compatible expression):

$$\bar{H} = \mathbb{I}_{n_b} \otimes H, \tag{1}$$

where $\bar{H} \in \mathbb{R}^{n \times n}$ stands for the operator itself and $H \in \mathbb{R}^{n/n_b \times n/n_b}$ for its restriction to the base mesh. Then, the standard SpMV on $\bar{H}$:

$$y = \begin{pmatrix} H & & \\ & \ddots & \\ & & H \end{pmatrix} \begin{pmatrix} x_1 \\ \vdots \\ x_{n_b} \end{pmatrix} \in \mathbb{R}^n, \tag{2}$$

can be replaced with an SpMM on H:

$$(y_1 \dots y_{n_b}) = H(x_1 \dots x_{n_b}) \in \mathbb{R}^{n/n_b \times n_b}. \tag{3}$$

SpMV vs SpMM

- SpMM reads H n_b less times
- $\bar{H}$ takes n_b times more memory than H

A. Alsalti-Baldellou et al. (2023). "Lighter and faster simulations on domains with symmetries", submitted.

Numerical results

Exploiting symmetries – 1

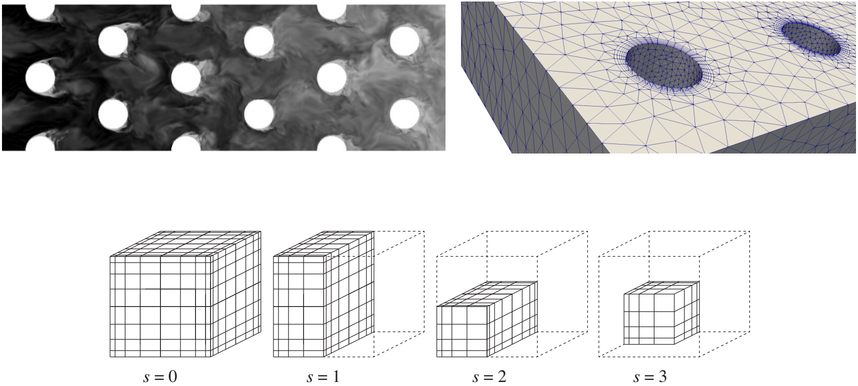
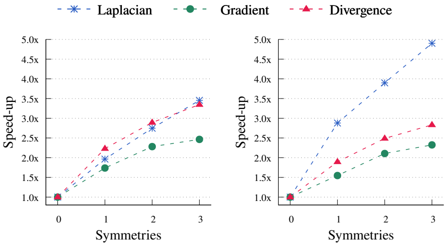
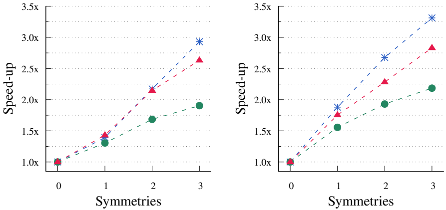


Figure: Top: 17.7M wall-bounded pin matrix heat exchanger. Bottom: 15.5M cubic mesh.

Exploiting symmetries – 2



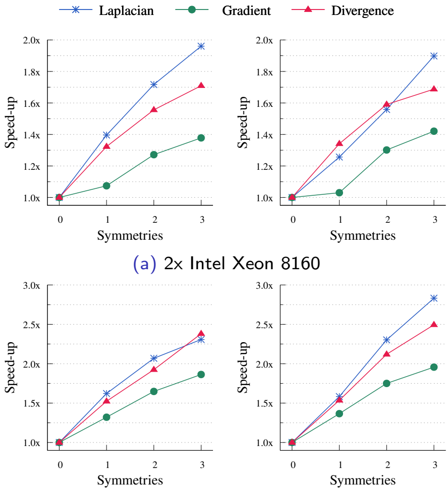
(a) 2x Intel Xeon 8160



(b) NVIDIA RTX A5000

Figure: SpMM speedups on a fixed problem size. Left: structured. Right: unstructured.

Exploiting symmetries – 3

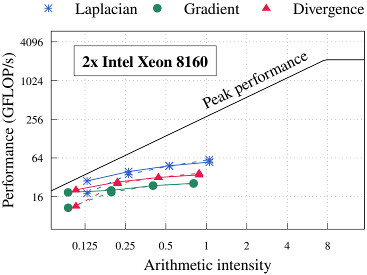


(a) 2x Intel Xeon 8160

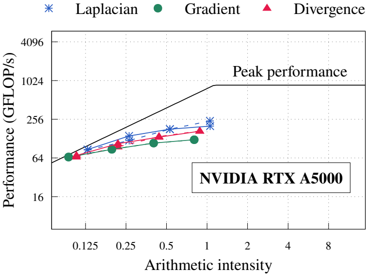
(b) NVIDIA RTX A5000

Figure: SpMM speedups on a fixed base mesh. Left: structured. Right: unstructured.

Exploiting symmetries – 4



(a) 2x Intel Xeon 8160



(b) NVIDIA RTX A5000

Figure: SpMM's roofline analysis. Dashed: fixed problem size. Solid: fixed base mesh.

Exploiting symmetries – 5

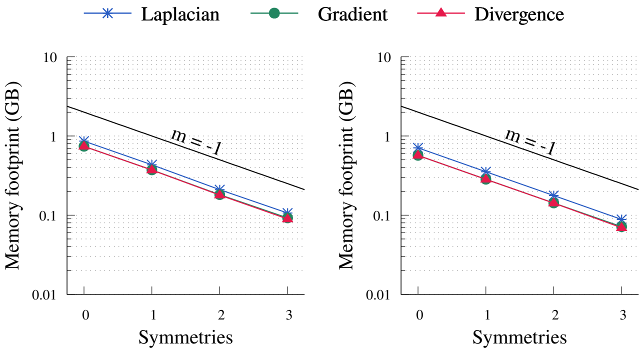


Figure: Operators' memory footprint. Left: structured. Right: unstructured.

Exploiting symmetries – 5

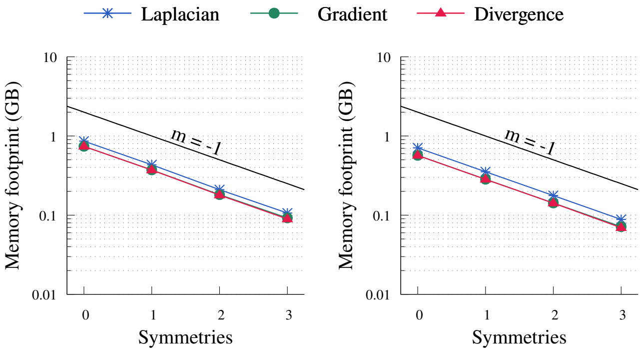
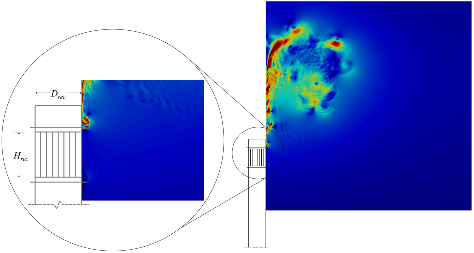


Figure: Operators' memory footprint. Left: structured. Right: unstructured.

More generally: repeated geometries lead to n_b times smaller footprints!

Test-case: CSP central tower receiver



Assumptions for industrial LES

LES limitation to be routinely applied in the industry: to be completed overnight.

- Mesh resolution: 300M-500M grid
- Simulated time period: 150 time units
- Wall-clock time limit: 16 hours

R. Löhner et al. (2011). "Overnight industrial LES for external aerodynamics" in *Computers & Fluids*.

TFA vs OpenFOAM: strong scalability

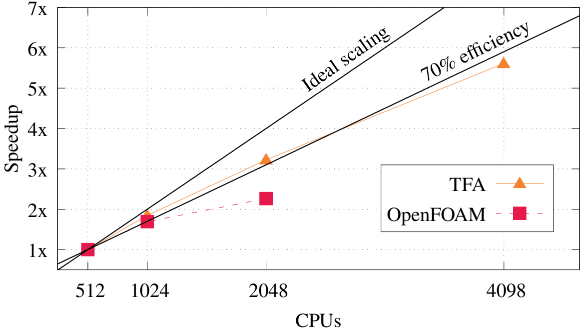


Figure: Scalability of TFA (MPI+OpenMP) vs OpenFOAM (MPI-only) down to 70% efficiency on a 500M CSP structured grid. Ran on AMD EPYC Rome nodes.

Towards overnight LES

Assuming constant Δt, to simulate τ time units, the required time-steps are:

$$n_{\Delta t} = \frac{\tau}{\Delta t}.$$

F.X. Trias et al. (2010). "Direct numerical simulation of a differentially heated cavity of aspect ratio 4 with Rayleigh numbers up to 10^{11} – Part I: Numerical methods and time-averaged flow" in *International Journal of Heat and Mass Transfer*.

Towards overnight LES

Assuming constant Δt, to simulate τ time units, the required time-steps are:

$$n_{\Delta t} = \frac{\tau}{\Delta t}.$$

Recalling that LES are generally convection-dominated, for some correction constant c:

$$\Delta t = \min\left\{\frac{\Delta x_i}{|u_i|}\right\} \simeq \frac{c}{\sqrt[3]{N}},$$

where Δx, u and N stand for the cell length, local velocity and mesh size.

F.X. Trias et al. (2010). "Direct numerical simulation of a differentially heated cavity of aspect ratio 4 with Rayleigh numbers up to 10^{11} – Part I: Numerical methods and time-averaged flow" in *International Journal of Heat and Mass Transfer*.

Towards overnight LES

Assuming constant Δt, to simulate τ time units, the required time-steps are:

$$n_{\Delta t} = \frac{\tau}{\Delta t}.$$

Recalling that LES are generally convection-dominated, for some correction constant c:

$$\Delta t = \min\left\{\frac{\Delta x_i}{|u_i|}\right\} \simeq \frac{c}{\sqrt[3]{N}},$$

where Δx, u and N stand for the cell length, local velocity and mesh size. Let $T_{\Delta t}^{\text{eff}}$ be the wall-clock time per time-step at a given parallel efficiency and on a mesh of size N_{ref}. Then, the wall-clock time of an LES of size N can be approximated as:

$$T_{\text{LES}}(N) \simeq n_{\Delta t}\frac{T_{\Delta t}^{\text{eff}} N}{N_{\text{ref}}} = \frac{\tau T_{\Delta t}^{\text{eff}}}{c N_{\text{ref}}}\sqrt[3]{N^4}.$$

F.X. Trias et al. (2010). "Direct numerical simulation of a differentially heated cavity of aspect ratio 4 with Rayleigh numbers up to 10^{11} – Part I: Numerical methods and time-averaged flow" in *International Journal of Heat and Mass Transfer*.

Towards overnight LES

Assuming constant Δt, to simulate τ time units, the required time-steps are:

$$n_{\Delta t} = \frac{\tau}{\Delta t}.$$

Recalling that LES are generally convection-dominated, for some correction constant c:

$$\Delta t = \min \left\{ \frac{\Delta x_i}{|u_i|} \right\} \simeq \frac{c}{\sqrt[3]{N}},$$

where Δx, u and N stand for the cell length, local velocity and mesh size. Let $T_{\Delta t}^{\text{eff}}$ be the wall-clock time per time-step at a given parallel efficiency and on a mesh of size N_{ref}. Then, the wall-clock time of an LES of size N can be approximated as:

$$T_{\text{LES}}(N) \simeq n_{\Delta t} \frac{T_{\Delta t}^{\text{eff}} N}{N_{\text{ref}}} = \frac{\tau T_{\Delta t}^{\text{eff}}}{c N_{\text{ref}}} \sqrt[3]{N^4}.$$

According to Trias et al. (2010), $c \simeq 0.3$ and after 100 time units the flow starts becoming statistically stationary, so we take $\tau = 150$.

F.X. Trias et al. (2010). "Direct numerical simulation of a differentially heated cavity of aspect ratio 4 with Rayleigh numbers up to 10^{11} – Part I: Numerical methods and time-averaged flow" in *International Journal of Heat and Mass Transfer*.

TFA vs OpenFOAM: strong scalability

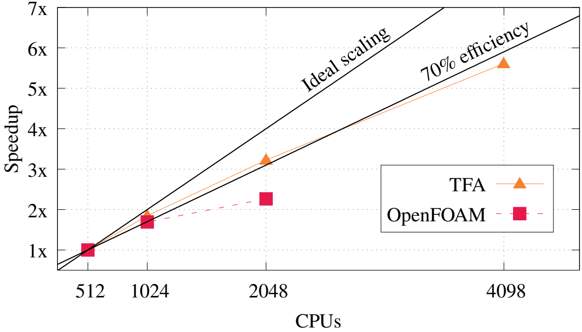


Figure: Scalability of TFA (MPI+OpenMP) vs OpenFOAM (MPI-only) down to 70% efficiency on a 500M CSP structured grid. Ran on AMD EPYC Rome nodes.

Towards overnight LES – 95% parallel efficiency

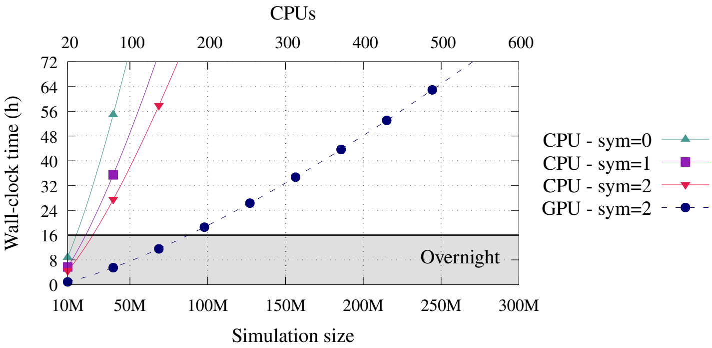


Figure: Estimated largest affordable overnight simulations on a 500M CSP structured grid. Ran on AMD EPYC Rome nodes and assuming a conservative 5x GPU speedup.

Towards overnight LES – 75% parallel efficiency

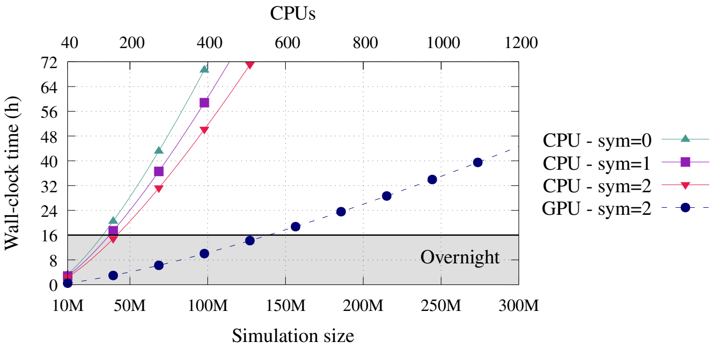


Figure: Estimated largest affordable overnight simulations on a 500M CSP structured grid. Ran on AMD EPYC Rome nodes and assuming a conservative 5x GPU speedup.

Towards overnight LES – 65% parallel efficiency

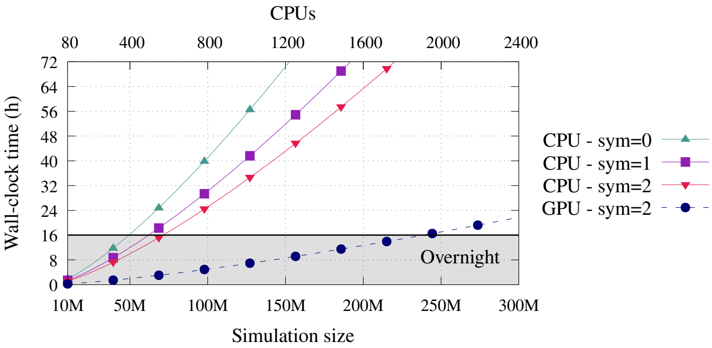


Figure: Estimated largest affordable overnight simulations on a 500M CSP structured grid. Ran on AMD EPYC Rome nodes and assuming a conservative 5x GPU speedup.

Concluding remarks

Conclusions

Summary:

- The algebra-based design allows for easy performance portability

Conclusions

Summary:

- The algebra-based design allows for easy performance portability
- Despite the challenges it poses, it opens the door to new opportunities:
  - Specialised kernels such as SpMM
  - Specialised sparse matrix formats
  - Specialised solvers and preconditioners

Conclusions

Summary:

- The algebra-based design allows for easy performance portability
- Despite the challenges it poses, it opens the door to new opportunities:
  - Specialised kernels such as SpMM
  - Specialised sparse matrix formats
  - Specialised solvers and preconditioners
- Massively parallel accelerators open the door to overnight industrial LES

Conclusions

Summary:

- The algebra-based design allows for easy performance portability
- Despite the challenges it poses, it opens the door to new opportunities:
  - Specialised kernels such as SpMM
  - Specialised sparse matrix formats
  - Specialised solvers and preconditioners
- Massively parallel accelerators open the door to overnight industrial LES

Ongoing work:

- Extend TFA vs OpenFOAM comparison

Conclusions

Summary:

- The algebra-based design allows for easy performance portability
- Despite the challenges it poses, it opens the door to new opportunities:
  - Specialised kernels such as SpMM
  - Specialised sparse matrix formats
  - Specialised solvers and preconditioners
- Massively parallel accelerators open the door to overnight industrial LES

Ongoing work:

- Extend TFA vs OpenFOAM comparison
- Evaluate the gains with GPUs

Conclusions

Summary:

- The algebra-based design allows for easy performance portability
- Despite the challenges it poses, it opens the door to new opportunities:
  - Specialised kernels such as SpMM
  - Specialised sparse matrix formats
  - Specialised solvers and preconditioners
- Massively parallel accelerators open the door to overnight industrial LES

Ongoing work:

- Extend TFA vs OpenFOAM comparison
- Evaluate the gains with GPUs
- Evaluate the gains on regular domains like wind-farms and heat exchangers

Conclusions

Summary:

- The algebra-based design allows for easy performance portability
- Despite the challenges it poses, it opens the door to new opportunities:
  - Specialised kernels such as SpMM
  - Specialised sparse matrix formats
  - Specialised solvers and preconditioners
- Massively parallel accelerators open the door to overnight industrial LES

Ongoing work:

- Extend TFA vs OpenFOAM comparison
- Evaluate the gains with GPUs
- Evaluate the gains on regular domains like wind-farms and heat exchangers
- Evaluate the gains with a novel multigrid reduction framework

Thanks for your attention!