ACCELERATED CFD SIMULATIONS AT DLR

Jonathan Fenske, Daniel Molka, Ronny Tschüter, Michael Wagner
Institute of Software Methods for Product Virtualization (SP)
German Aerospace Center (DLR)

Outline

- DLR HPC Systems
- CFD Simulation Software
- Benchmarks of CFD Simulation Runtime Improvements
- Hierarchical Partitioning
- Summary & Outlook

DLR HPC SYSTEMS

HPC Systems operated by DLR-SP

HPC cluster CARO, Göttingen

© Johannes Biermann, GWDG

HPC cluster CARA, Dresden

© Robert Gommlich

HPC Systems operated by DLR-SP

HPC cluster CARA, Dresden

Credit: DLR (CC BY-NC-ND 3.0)

- 2,168 CPU nodes with 2 AMD EPYC 7601 (Zen)
- 664 CPU nodes with 2 AMD EPYC 7702 (Zen 2)
- 10 GPU nodes with 4 Nvidia A100 and 2 AMD EPYC 7702
- 17 PB Lustre file system

HPC cluster CARO, Göttingen

Credit: DLR (CC BY-NC-ND 3.0)

- 1,364 CPU nodes with 2 AMD EPYC 7702 (Zen 2)
- 8.4 PB Lustre file system

HPC Systems operated by DLR-SP

HPC cluster CARA, Dresden

Credit: DLR (CC BY-NC-ND 3.0)

New system to be installed in 2025

- 770 CPU nodes with 2 AMD EPYC 9555 (Zen 5)
- 30 GPU nodes with 4 Nvidia H100 and 2 Intel Xeon 6530
- 30 PB IBM Storage Scale

Continued operation

- 664 CPU nodes with 2 AMD EPYC 7702 (Zen 2)
- 10 GPU nodes with 4 Nvidia A100 and 2 AMD EPYC 7702

HPC cluster CARO, Göttingen

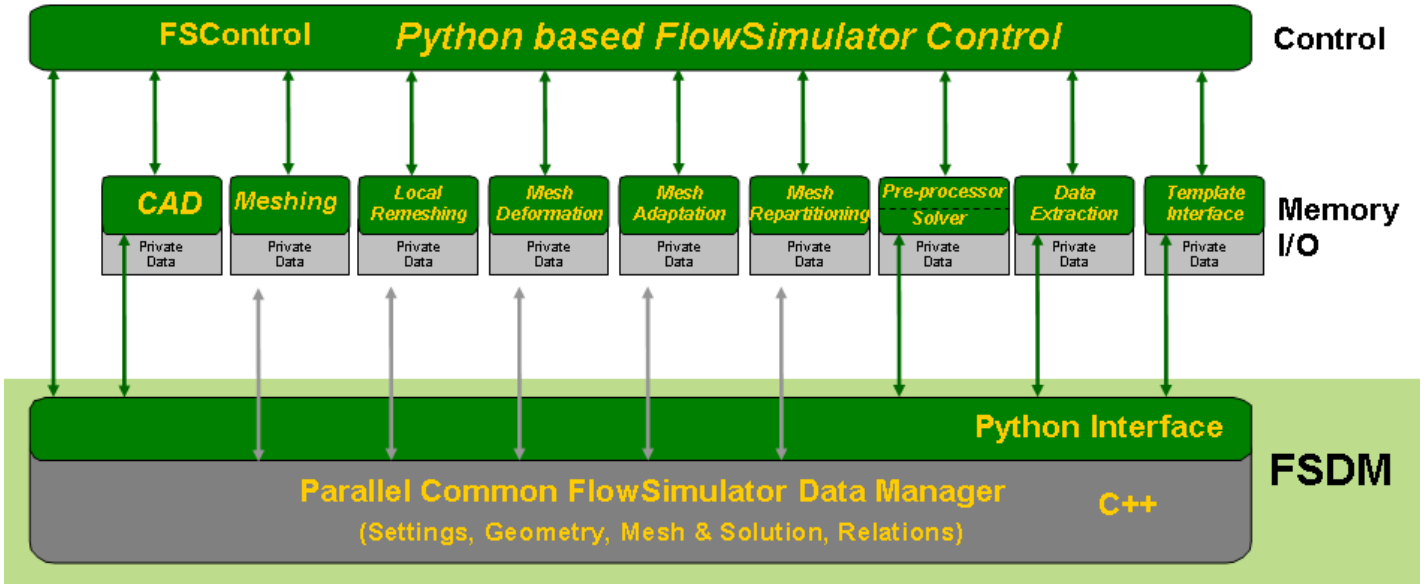
Credit: DLR (CC BY-NC-ND 3.0)

- 1,364 CPU nodes with 2 AMD EPYC 7702 (Zen 2)
- 8.4 PB Lustre file system

CFD SIMULATION SOFTWARE

FlowSimulator Framework

- Data manager for parallel data management for CFD simulations:
  - Importing/exporting data
  - Partitioning data (ParMETIS & Zoltan via PlugIn)
- Solvers or other mesh operations can be provided via plugin



CFD Software by ONERA, DLR, and Airbus (CODA)

- Next generation CFD solver employing finite volume and discontinuous Galerkin methods
- Hybrid MPI+threads approach
- Developed with a focus on scalability to enable efficient scale resolving simulations
- Uses FlowSimulator framework

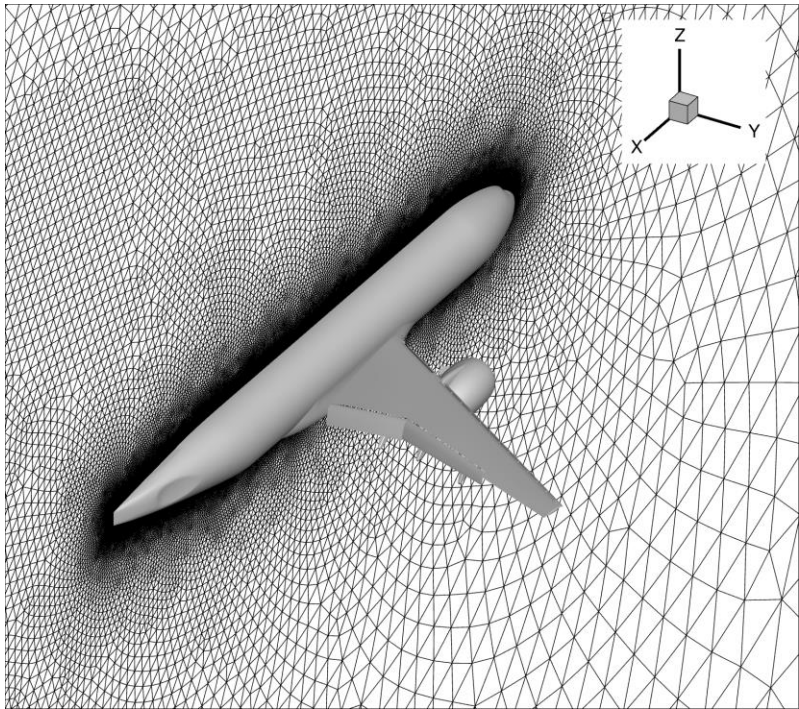
BENCHMARKS OF CFD SIMULATION RUNTIME IMPROVEMENTS

Test Cases

- NASA‘s Common Research Model (CRM) mesh family with 3M-192M cells
- Weak scaling benchmark, i.e., test runs with different resolutions of same mesh and scale the number of processes with the number of elements
- Strong scaling benchmark, i.e., test runs with the same mesh with different numbers of processes
- 100 GMRES iterations with Jacobi preconditioning
- Comparison of older and more recent CODA versions
- Tests on CARA with Zen architecture

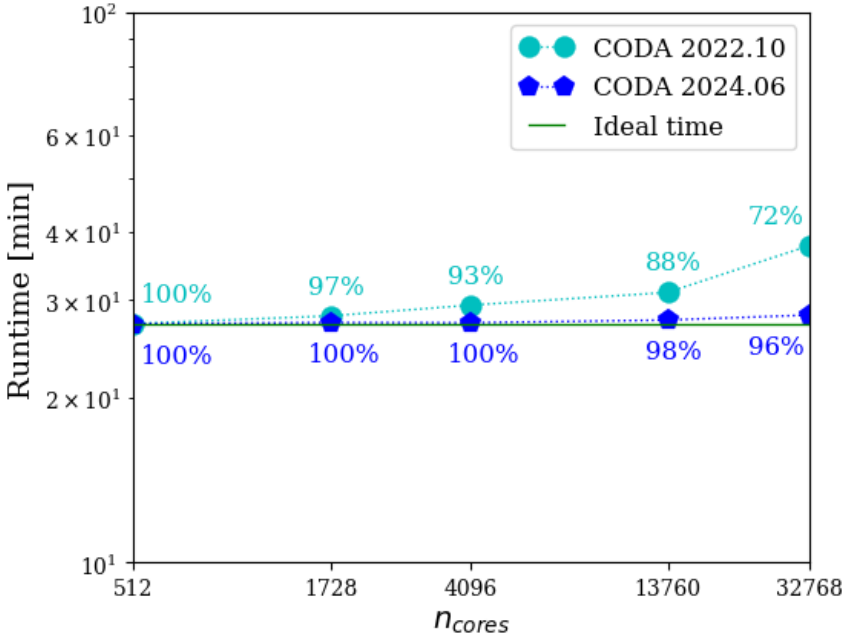
  (2x32 cores per node) and hybrid MPI+threads

  with 16 tasks per node (4 threads per task)



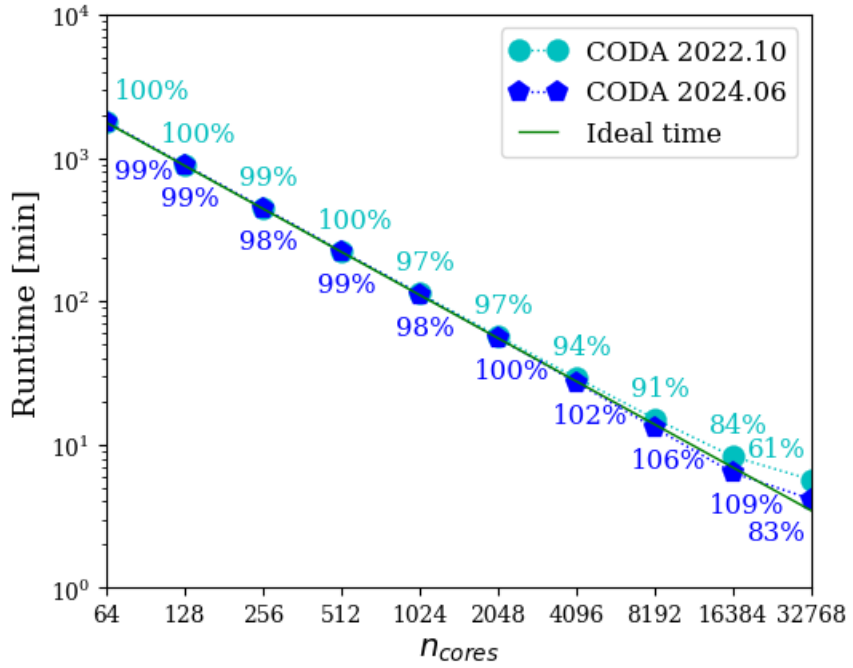
Benchmarks

Weak Scaling (3M-192M cells)*



- Improved weak scaling:
  - Before: 72% weak scaling efficiency
  - Now: 96% weak scaling efficiency

Strong Scaling (24M cells)*



- Improved strong scaling:
  - Before: 61% strong scaling efficiency
  - Now: 83% strong scaling efficiency

*Data provided by Michael Wagner

HIERARCHICAL PARTITIONING

Hierarchical Partitioning

- Partitioning data on multiple hardware hierarchy levels:
  - First partition data among compute nodes, then recursively on lower hierarchy levels within compute nodes, …
  - Benefit: reduced inter node communication time, domain decomposition for less processes at once
  - Drawback: potentially higher load imbalance
- Implementation in FlowSimulator:
  - FlowSimulator plugin
  - 2-level hierarchical partitioner (nodes and processes)
  - In the respective hierarchy level partitioning with third party partitioner (currently ParMETIS and Zoltan)

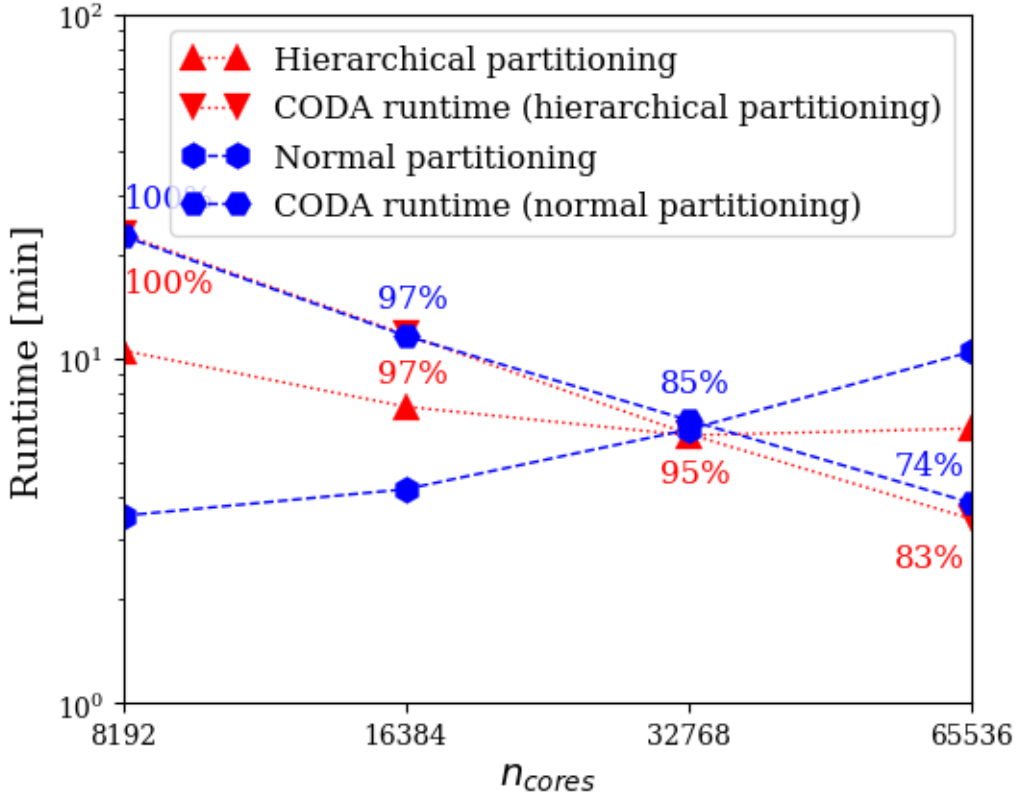
Test Case

- CRM mesh with 1.3B cells
- Strong scaling benchmark, i.e., test runs with the same mesh with different numbers of processes
- 100 Jacobi iterations with LU preconditioning
- Comparison of CODA and partitioning runtimes after direct partitioning with ParMETIS vs. hierarchical partitioning (also internally with ParMETIS)
- Tests on CARO with Zen 2 architecture (2x64 cores per node) and hybrid MPI+threads with 32 tasks per node (4 threads per task)

Results

Strong Scaling of Partitioning and CODA

- CODA runtimes better after hierarchical partitioning for a lot of processes
- For fewer processes, hierarchical partitioning is slower
- For more processes, hierarchical partitioning is faster
- For pure MPI, ParMETIS is limited to fewer than 256 compute nodes (32,768 cores) while hierarchical partitioning isn‘t (not shown here)



SUMMARY & OUTLOOK

Summary & Outlook

Summary

- CODA can already do parallel simulations with good parallel efficiency
- Hierarchical partitioning improves scalability even further
- Hierarchical partitioning potentially enables even larger simulations

Outlook

- Further improvement of parallel efficiency
- Running even larger simulations
- More focus on GPUs