PERFORMANCE AND SCALABILITY OF THE CODA CFD SOFTWARE

Michael Wagner and many DLR SP-HLR colleagues

International Parallel Tools Workshop, September 19th 2024

Background and Motivation

Numerical Simulation – Key Enabler for Future Aircraft Design

Future aircraft

- Goals: drastic reduction CO2, NOx and noise emissions
- Step changes in aircraft technology and new designs

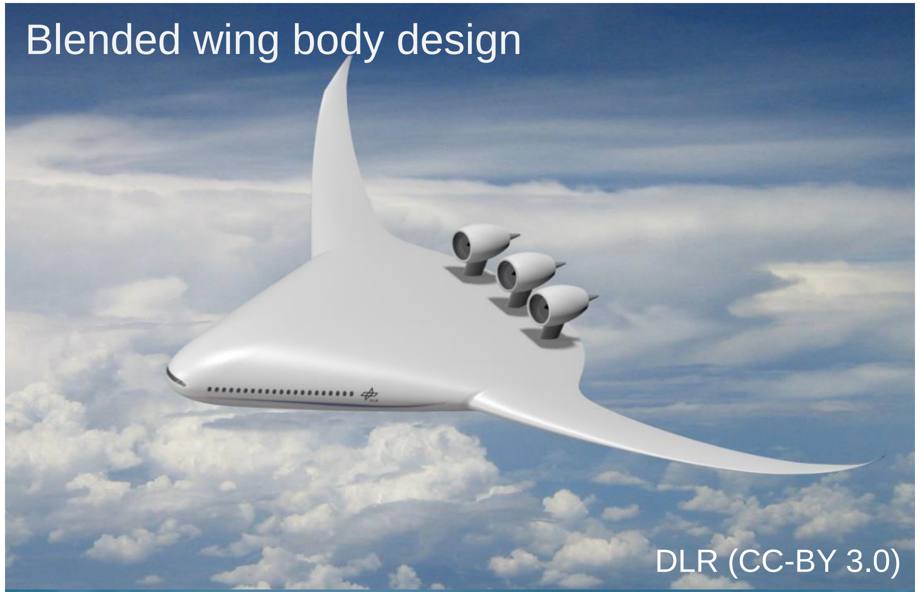
High-fidelity CFD methods indispensable

- Flight characteristics dominated by non-linear effects
- Reliable insight to new aircraft technologies
- High-fidelity CFD simulation of aircraft aerodynamics

Efficient linear system solving important

- CFD requires solving of large linear equation systems
- Linear systems solving makes up majority of time

Further improvement of simulation capabilities, computational efficiency and scalability necessary.



CODA Software Environment

CODA CFD Software

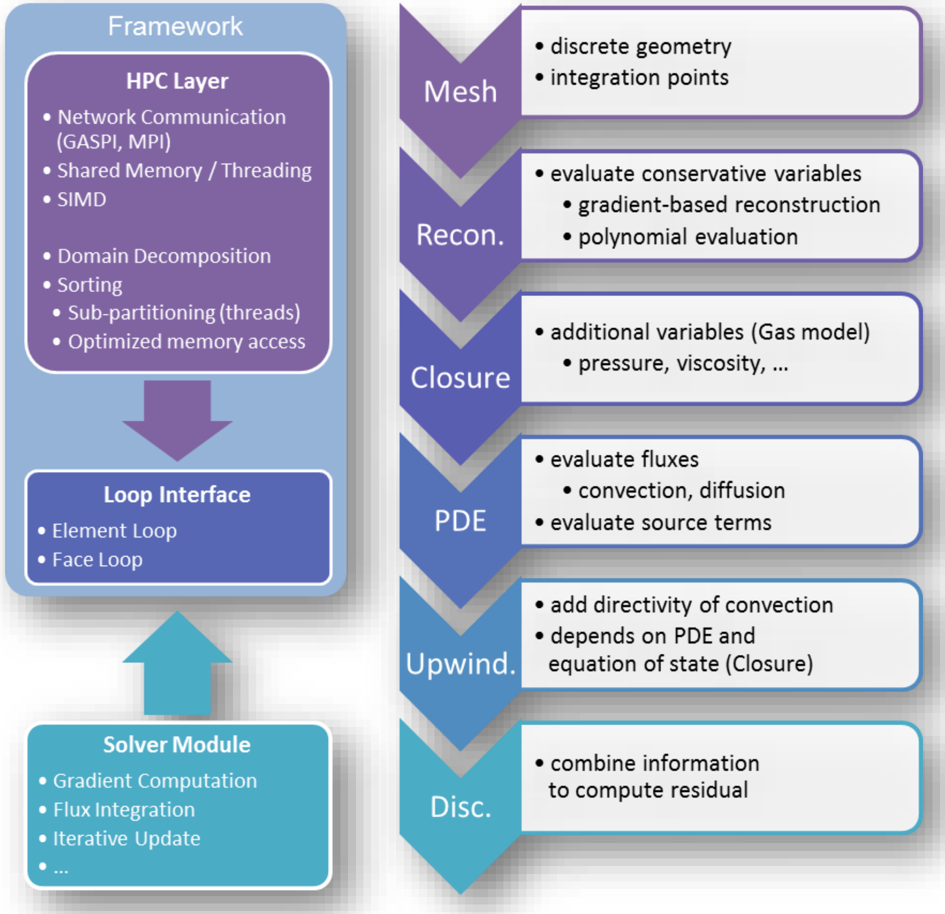
- Collaboration of ONERA, DLR and Airbus*
- 2nd order Finite Volume method and higher-order DG for unstructured grids and compressible flows
- Hybrid parallelization (MPI/GASPI + OpenMP/threads) with overlap of communication & computation
- Seamless integration into multi-disciplinary simulations

FlowSimulator

- Provides plug-ins for all steps of a full aircraft simulation
- FSMesh class for unified data exchange among plug-ins

Spliss: Sparse Linear Systems Solver

- Linear systems solving for implicit methods
- Full HPC support: MPI/GASPI, Threads, SIMD, GPUs



*CODA is the computational fluid dynamics (CFD) software being developed as part of a collaboration between the French Aerospace Lab ONERA, the German Aerospace Center (DLR), Airbus, and their European research partners. CODA is jointly owned by ONERA, DLR and Airbus.

EXCELLERAT (P2 2023 – 2026)

The European Centre of Excellence for Engineering Applications (P1 2019 – 2022)

Preparing European engineering for exascale computing

- 15 partners
- 7 use cases: Alya, AVBP, CODA, m-AIA, Neko, Flew, OpenFoam
- Aerospace & Energy; CFD & Combustion

Cooperation with European engineering and HPC community

- Expertise from other leading-edge engineering codes with similar challenges and problems
- Access to the largest HPC systems in Europe
- Early access and experiences with new hardware and trends

EXCELLERAT Project Targets

Evaluate and demonstrate CODA's and FlowSim's readiness for exascale computing

- Continuous evaluation (and analysis) of CODA/FlowSimulator scalability improvements
- Large scale demonstrator: big mesh + big system
- Evaluation of new systems and emerging technologies

Use case: external aircraft aerodynamics

- Airflow for steady forward flight at subsonic speed
- Reynolds-averaged Navier-Stokes equations (RANS) with Spalart-Allmaras turbulence model (SA-neg)

EXCELLERAT Use Case and Inputs

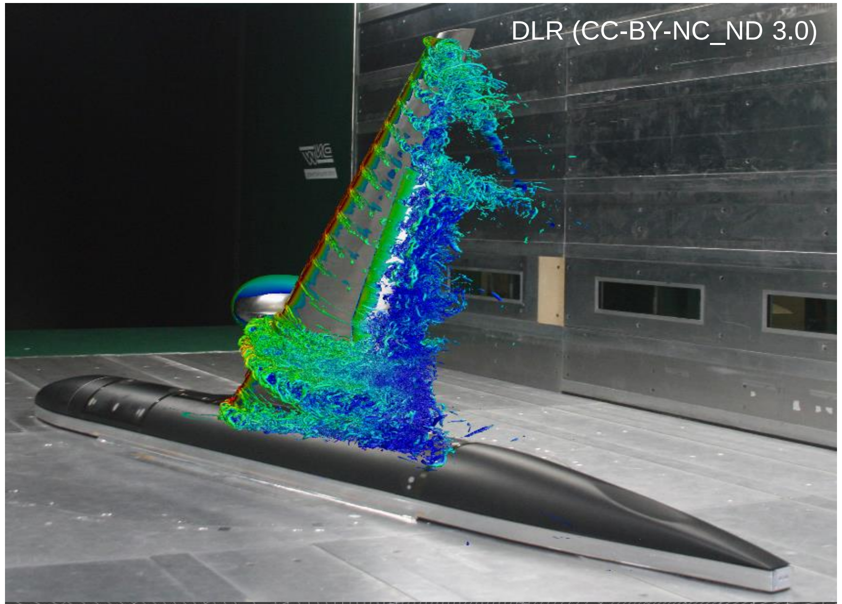
Strong and weak scaling use case

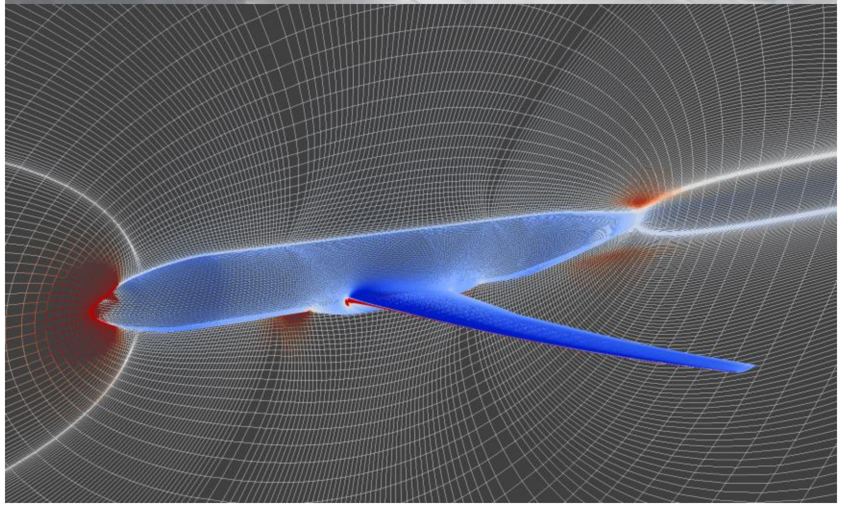
- NASA Common Research Model CRM (wing-body configuration)
- Mesh set with 3, 10, **24**, 81 and 192 million elements
- Practical size to see large scaling effects at smaller scales*
- Public, widely used and well-studied (also experimentally)

Capability demonstrator

- Demonstrate capabilities for big meshes on big systems
- Mesh with about 1 – 5 billion elements
- Upcoming European (pre-)exascale systems: ~500k cores

* within the range of available resources at DLR, i.e. up to 32/64k cores





The CARA and CARO HPC Systems at DLR

CARA (AMD Naples architecture):

- 2168 nodes with 128 GB DDR4 (2666 MHz)
- 2x AMD Epyc 7601 (32 cores; 2,2 GHz) per node
- 145.920 cores delivering 1.7 TFLOP/s
- Infiniband HDR network

CARO (AMD Rome architecture)

- 1354 nodes with 256 GB DDR4 (3200 MHz) RAM
- 2x AMD Epyc 7702 (64 cores; 2,0 GHz) per node
- 174592 cores delivering 3.5 TFLOP/s
- Infiniband HDR network

The CARA and CARO HPC Systems – Comparison

Number of cores:

- CARO (AMD Epyc 7702) has 2x cores (128 vs. 64 per node)

Cache:

- CARO has 4x last-level cache (256 MiB vs 64 MiB), i.e. twice as much per core.
- 16 vs. 8 NUMA domains
- 3 NUMA distances (on die, on socket, 2nd socket)
- 4 cores per die share L3

Memory access:

- 8 memory channels and memory controllers
- Memory controllers: 3200MHz (CARO) vs. 2666MHz (CARA)
- CARO has 1.2x memory bandwidth (191 GiB/s vs. 159 GiB/s) for twice the number of cores.

DLR (CC-BY 3.0)

DLR (CC-BY 3.0)

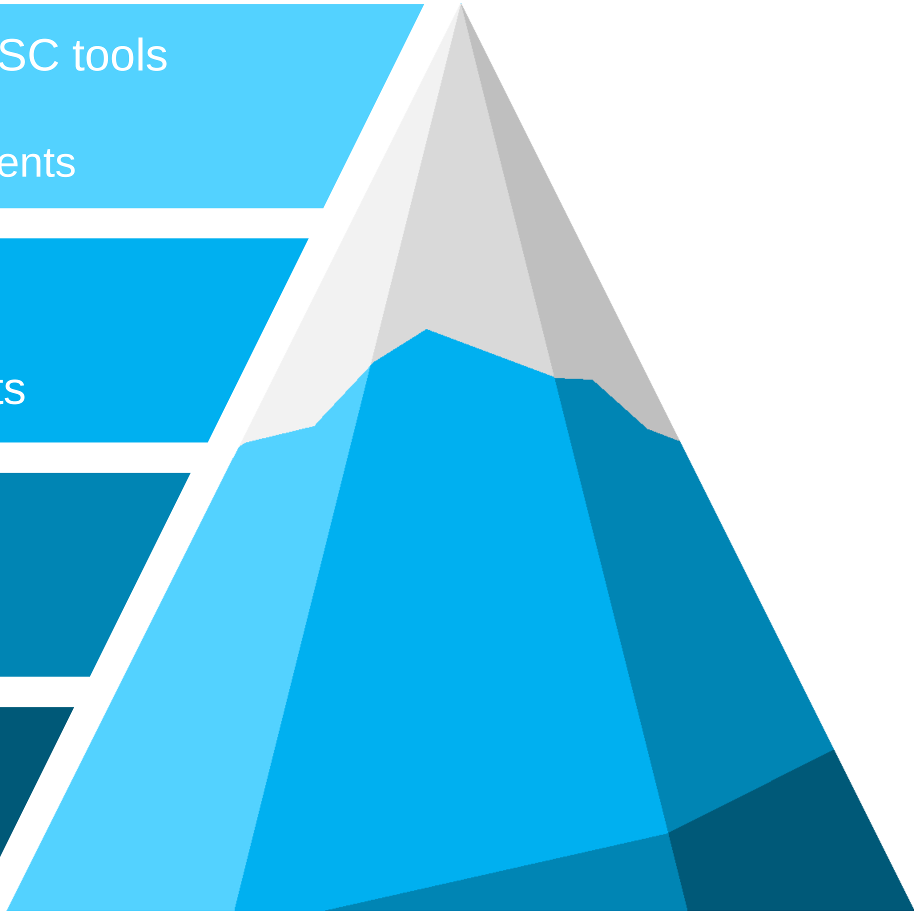
Different Levels of Performance Monitoring and Analysis

System-level performance: Score-P/Vampir, BSC tools
Separate build (spack recipes available)
For devs: identify bottlenecks, validate improvements

Node-level performance: Likwid
Separate build (spack recipes available)
For devs: identify bottlenecks, validate improvements

Component-level tracing: Perfetto
Built-in (enabled via run script)
For adv. users / devs, easy overview of components

Component-level timing: build-in timing output
Always enabled (automatically printed in output)
For users: runtime comparisons, regression tests



Component-level Timing

- Provides quick overview of component timing
- Useful for runtime comparison, regression tests etc.
- (Slurm) output with timings

```
FSMesh::RepartionMeshPARMETIS() 12.2 [s] (wall clock time)
...
Preprocessor::PreprocessMesh() building FaceBasedMeshAdapter 7.24 [s] (wall clock time)
...
Preprocessor::PreprocessMesh() wall distances and / or nearest wall faces 1.67 [min]
(wall clock time)
...
TimeIntegration::Iterate() 15 [h] (wall clock time)
```

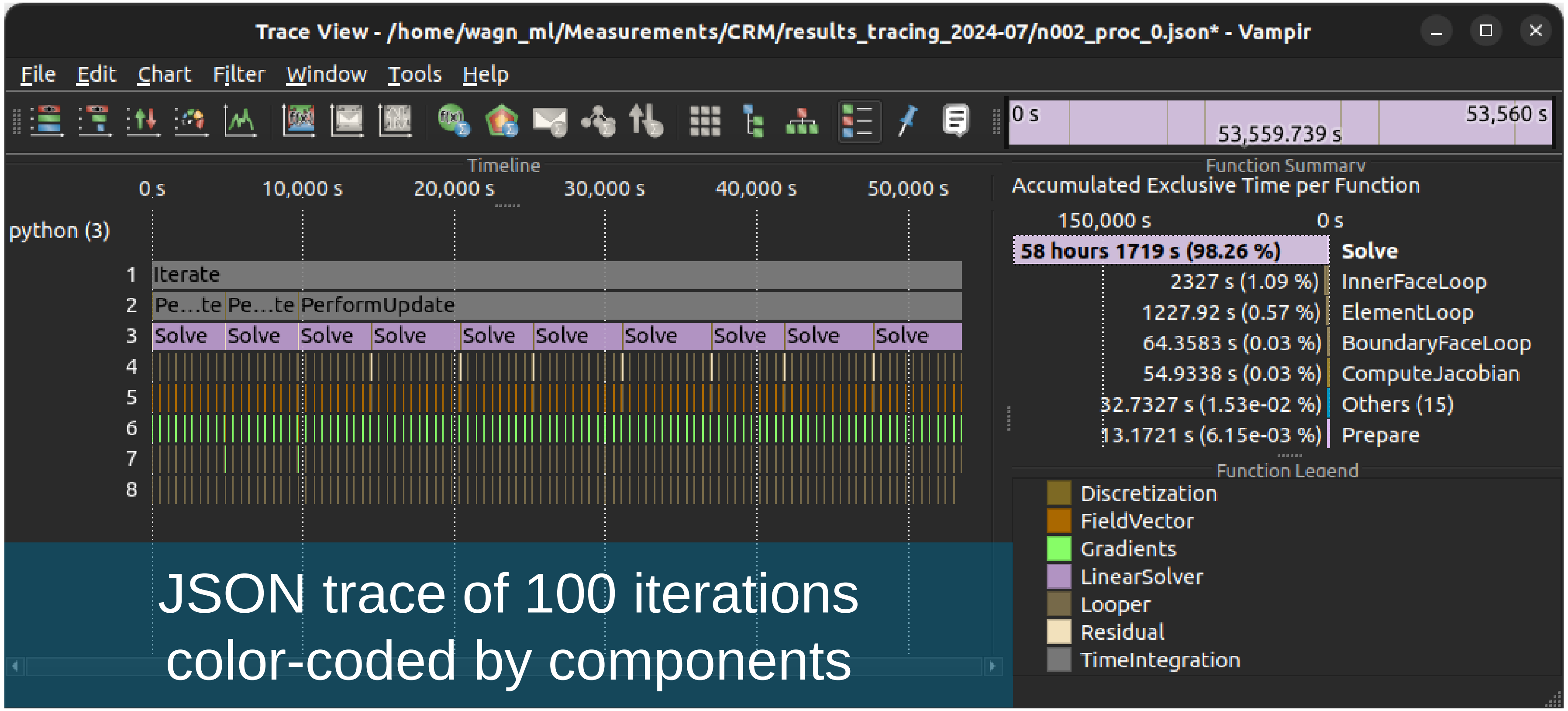
Component-level Tracing with Perfetto

- Provides quick overview of component behavior
- Perfetto instrumentation is shipped with CODA
- Tracing can be enabled via run script

```
tracingSession = TracingSession(…)
…
tracingSession.StartTracing()
timeIntegration.Iterate(…)
tracingSession.StopTracing()
```

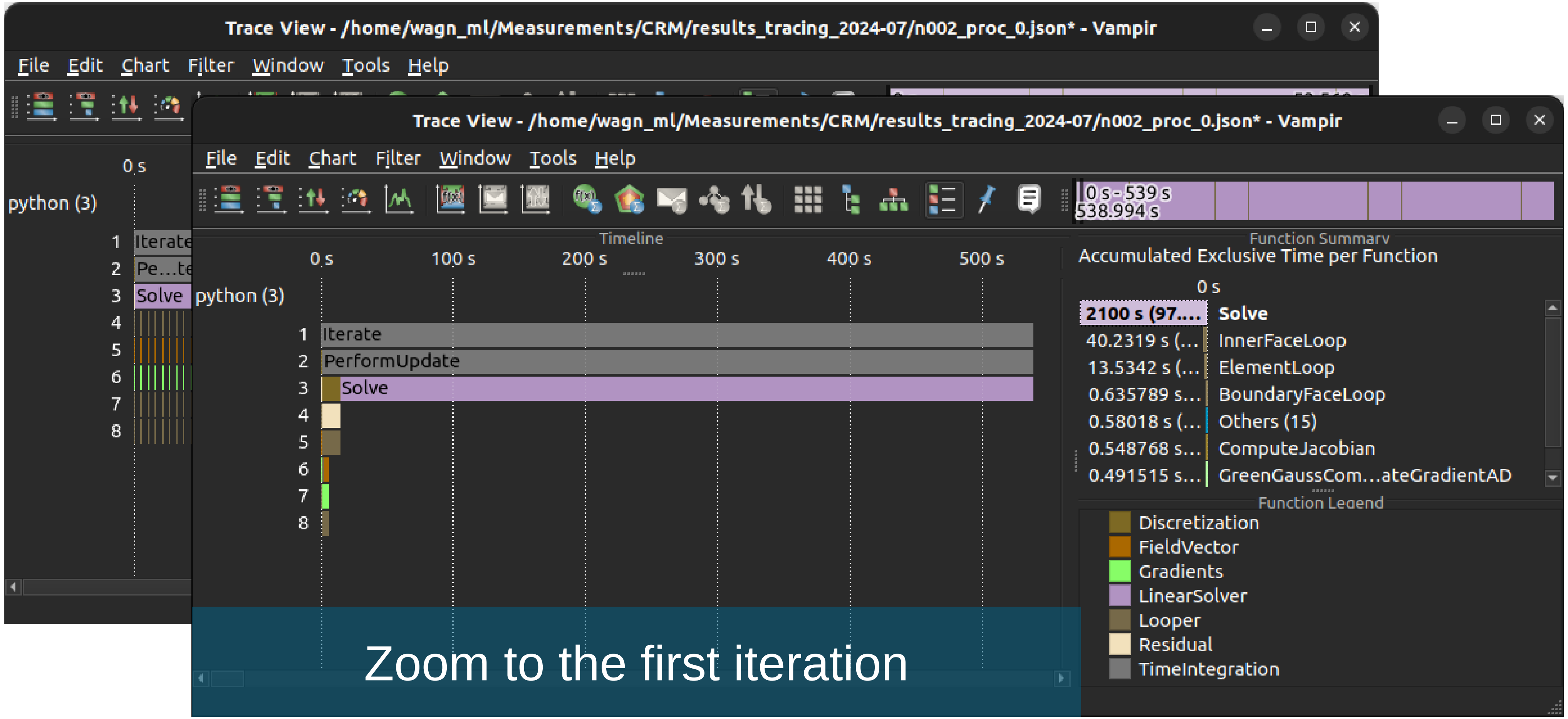
- Perfetto traces can be converted to JSON or OTF2 for viz tools like Vampir

Component-level Tracing with Perfetto (1)

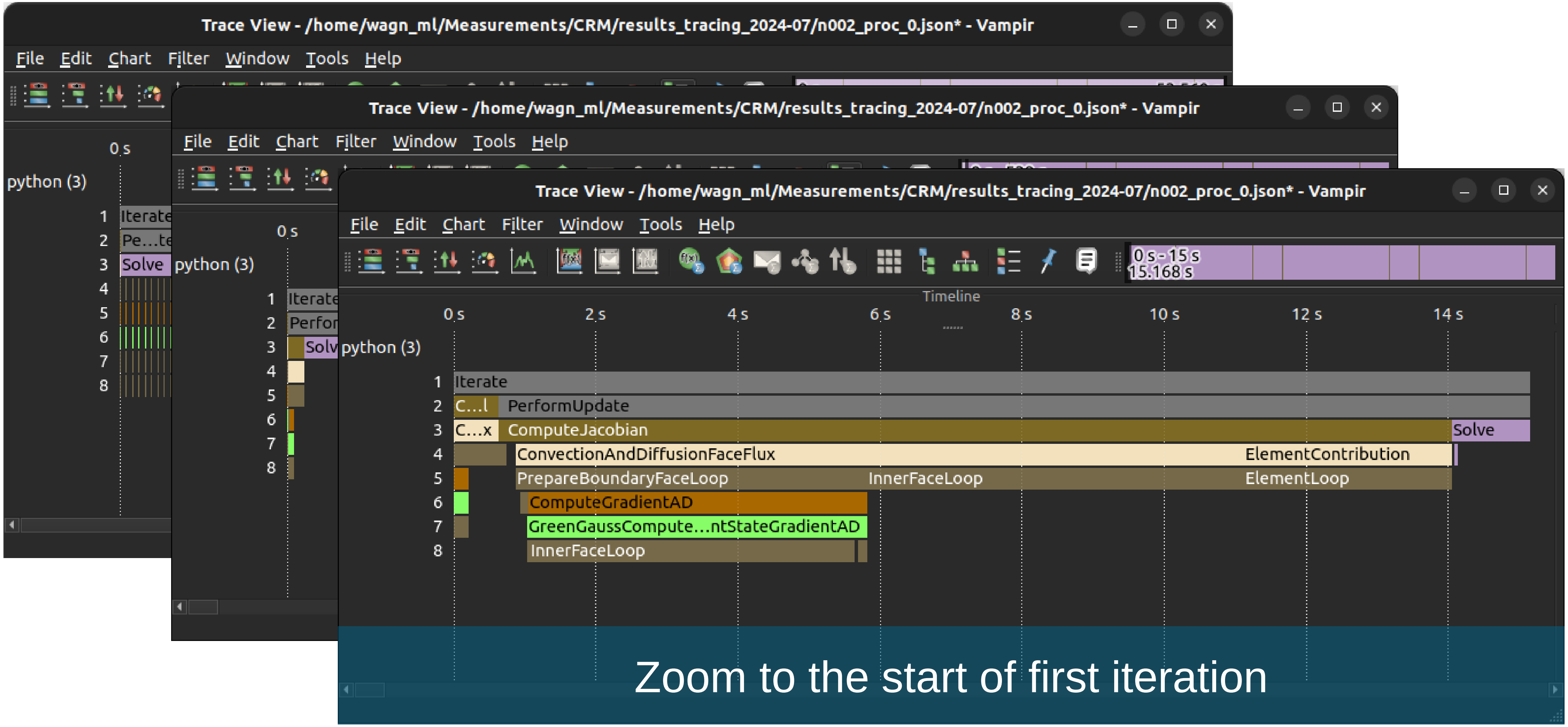


JSON trace of 100 iterations color-coded by components

Component-level Tracing with Perfetto (1)



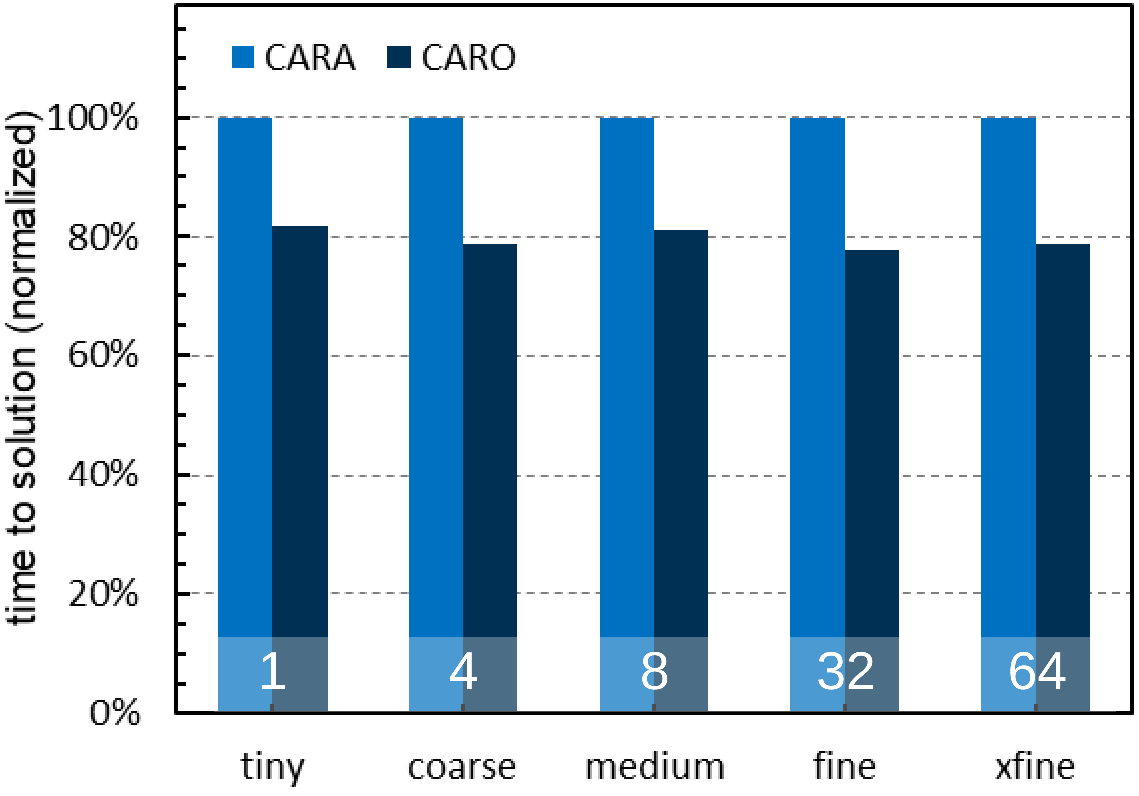
Component-level Tracing with Perfetto (1)



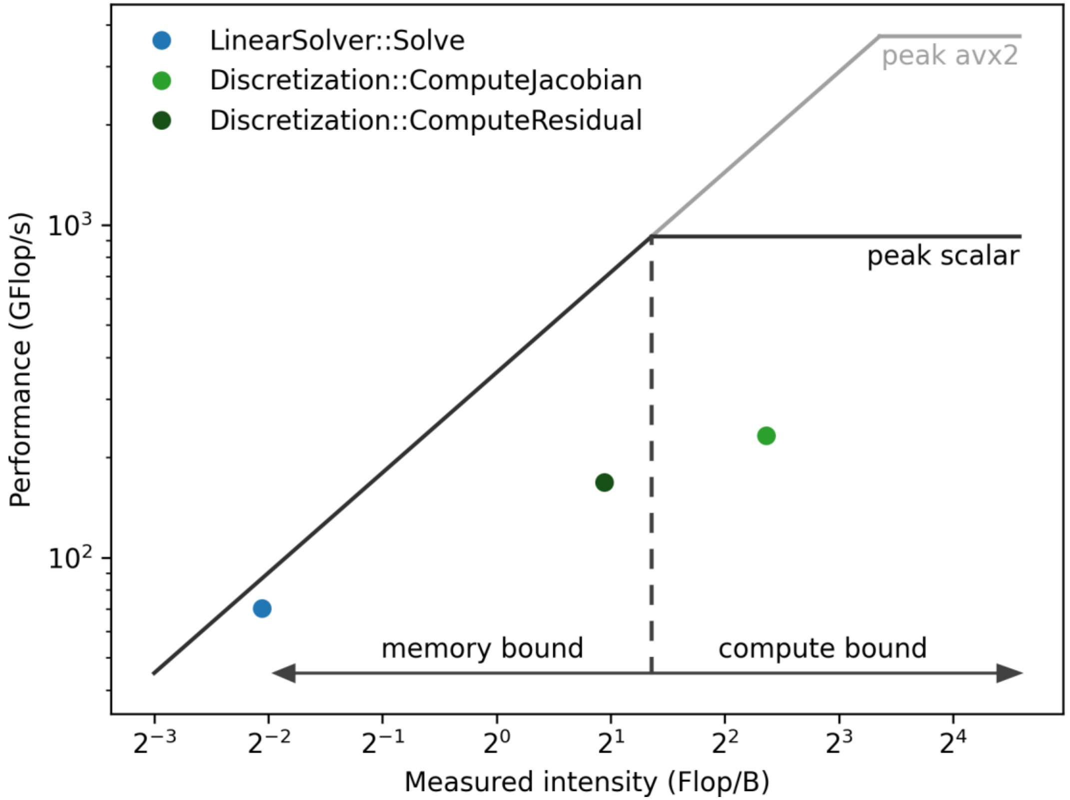
Zoom to the start of first iteration

Node-level Performance with Likwid

Runtime | CRM 3M-192M elem | appropriate node counts



Roofline | CARO (AMD Rome) | CRM 3M elements



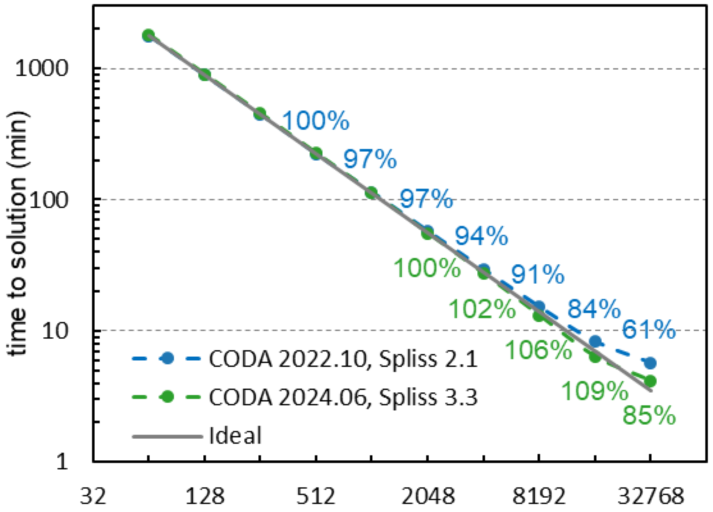
- Although CARO has 2x cores per node, the runtime is only about 1.2 times faster
- Memory-bound on the AMD Naples and Rome architectures
- Basically no benefit from the doubled compute power (2x cores but both with 8 memory controllers)
- 1.2x faster runtime due to increased clock speed of memory controllers / memory bandwidth

Scalability CARA (AMD Naples)

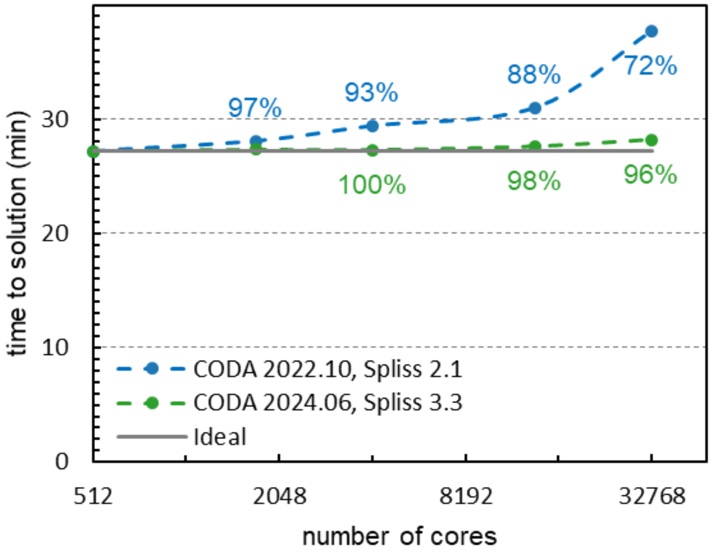
Scalability assessment on DLR's production system CARA

- Strong scaling (CRM, fixed problem size, 24M elements):
  - Scaling from 1 – 512 nodes (largest available partition)
  - Reduce runtime from 1.2 days to 4.2 minutes
  - **Small mesh**: just 730 elements/core @ 32,768 cores
  - Scaling 64 – **32,768 cores: 85% strong scaling efficiency**
  - Small super-linear speedup
- Weak scaling (CRM, fixed workload per core, 3M – 192M elements):
  - Scaling 512 – 32,768 cores: 96% weak scaling efficiency

Runtime | strong scaling 24M elements



Runtime | weak scaling 3M–192M elements

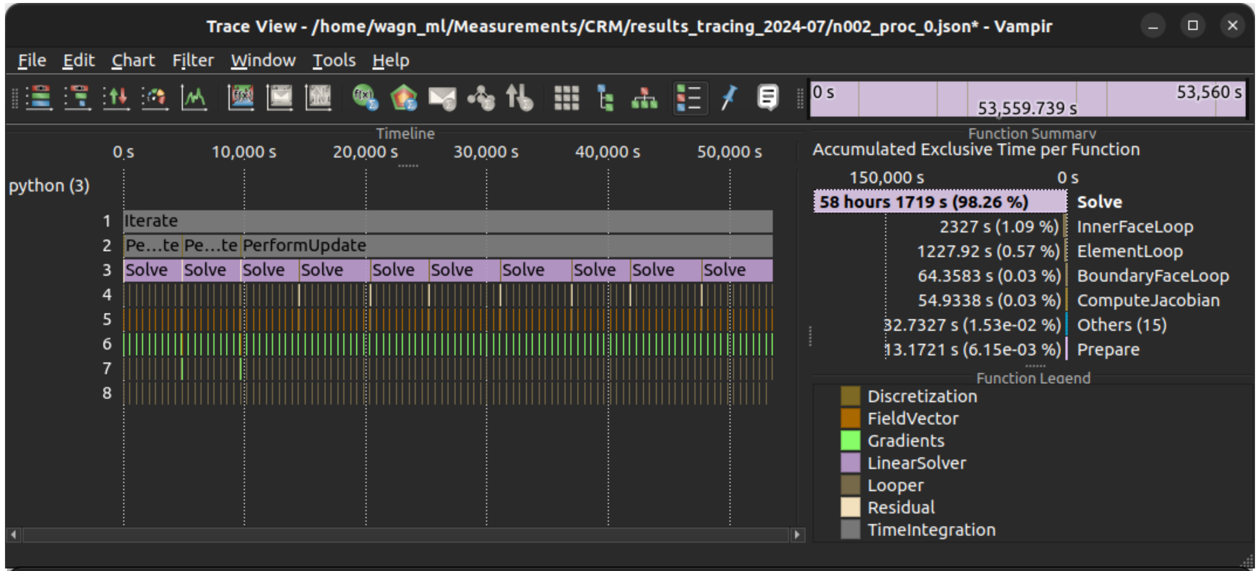


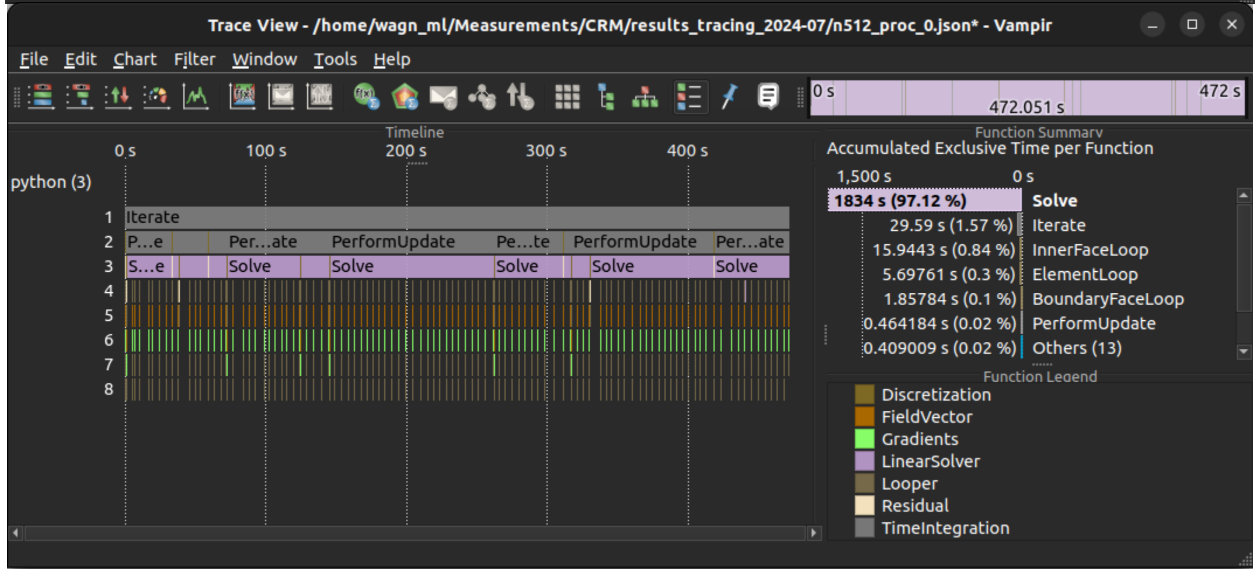
Component-level Tracing with Perfetto: Scalability

Comparing 2 to 512 nodes

- Main time is spent in the linear solver (Spliss)
- Spliss is primarily responsible for scaling
- Spliss part 98.2% to 97.1%
- All components scale similarly well
- Except main Iterate function: increases from 0.005% to 1.6%
- Otherwise limited insight
  - Separate trace per process
  - No MPI, OpenMP etc.

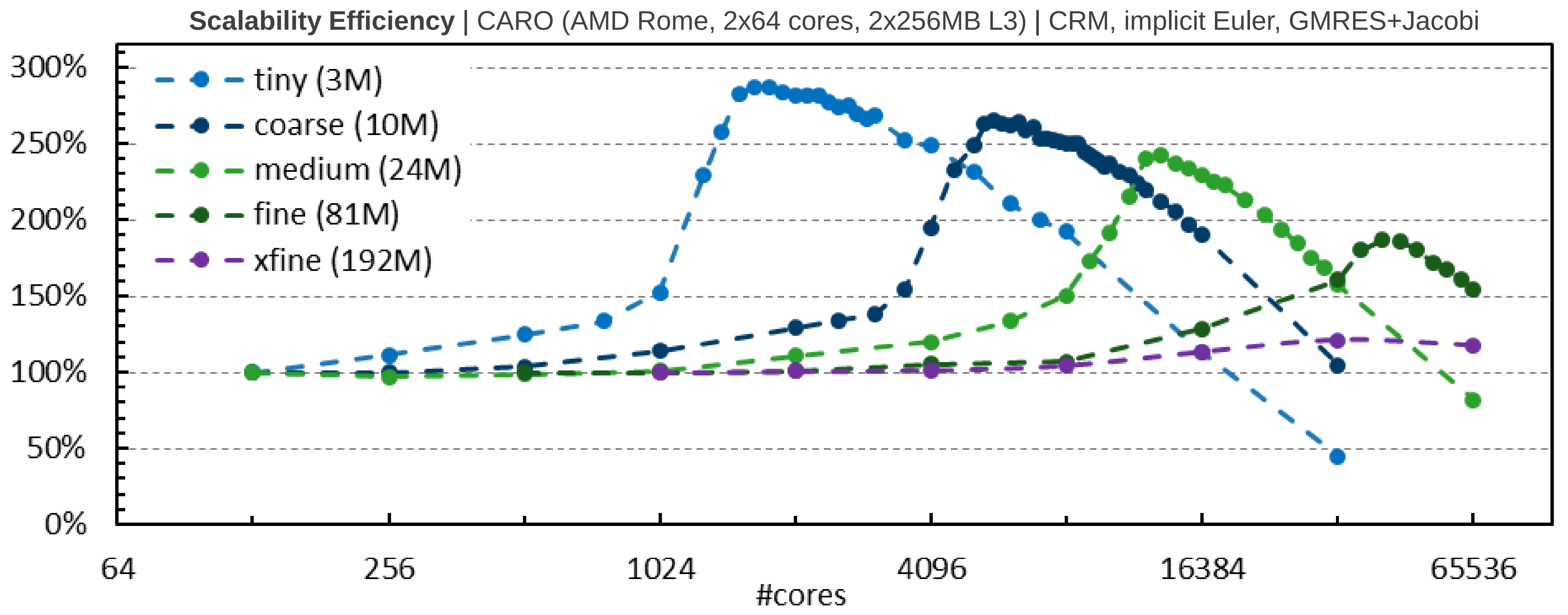
→ Performance tools with MPI/OpenMP support





Scalability Efficiency of CODA on CARO (AMD Rome)

CODA release 2022.10, Spliss release 2.1.0

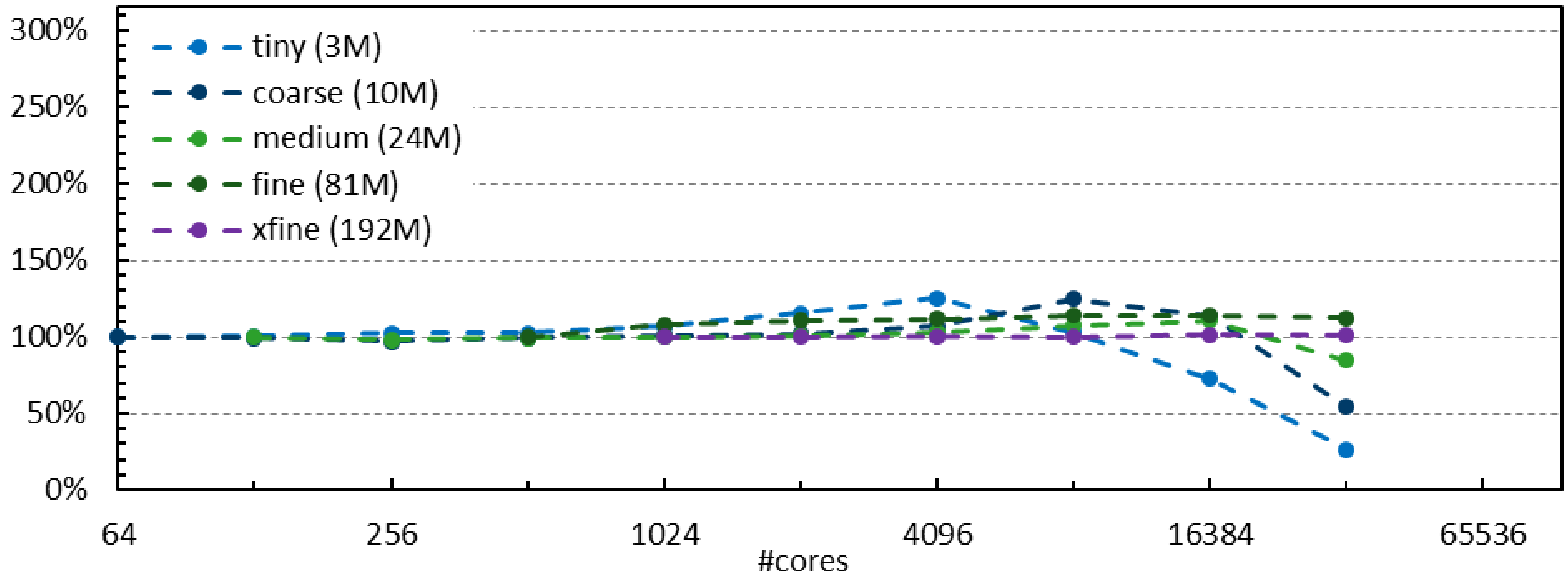


- Significant super-linear speedup: up to 287% scalability efficiency
- Overlapping effects of super-linear speedup and decreasing parallel efficiency
- Peak and general trend comparable for different mesh sizes
- Peak occurs at approx. the same number of elements per core (matching with L3 cache size)

Scalability Efficiency of CODA on CARA (AMD Naples)

CODA release 2024.06, Spliss release 3.3.0

Scalability Efficiency | CARA (AMD Naples, 2x32 cores, 2x64MB L3) | CRM, implicit Euler, GMRES+Jacobi

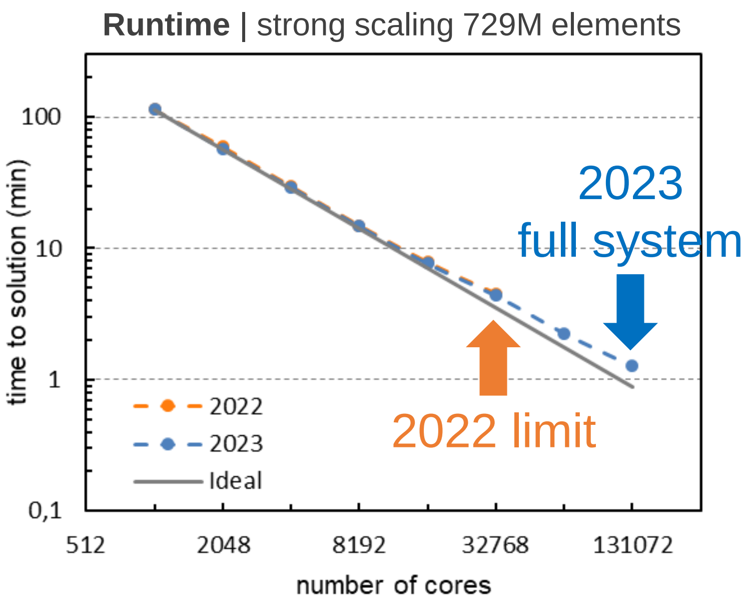


- Small super-linear speedup: up to 125% scalability efficiency
- Peak at double core count / half elements per core (only 2MB L3 cache/core)
- Super-linear speedup much higher on CARO due to higher memory-boundness

Scalability CARO (AMD Rome)

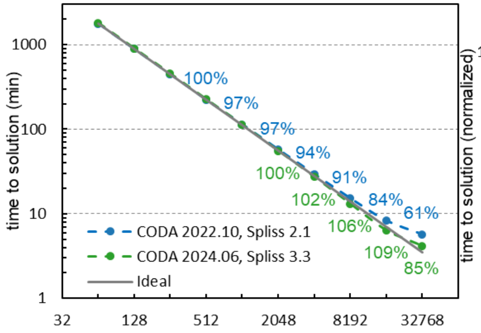
Scalability assessment on DLR's production system CARO

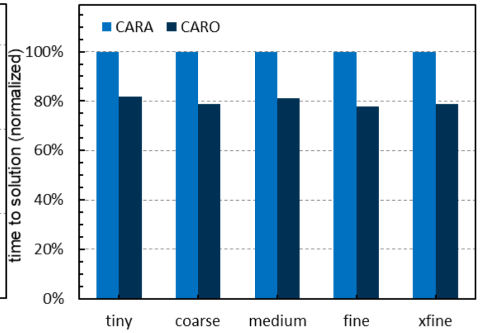
- Super-linear speed-up hinders useful scalability analysis
- Use similar CRM-HL test case with 729M elements
- Strong scaling
  - Scaling from 8 – 1024 nodes (almost full system)
  - Reduced runtime from 1.9 hours to 1.2 minutes
  - Scaling 1024 – **131,072 cores: 70% strong scaling efficiency**
- Several necessary improvements in FlowSimulator to scale to full system
  → More details in next talk

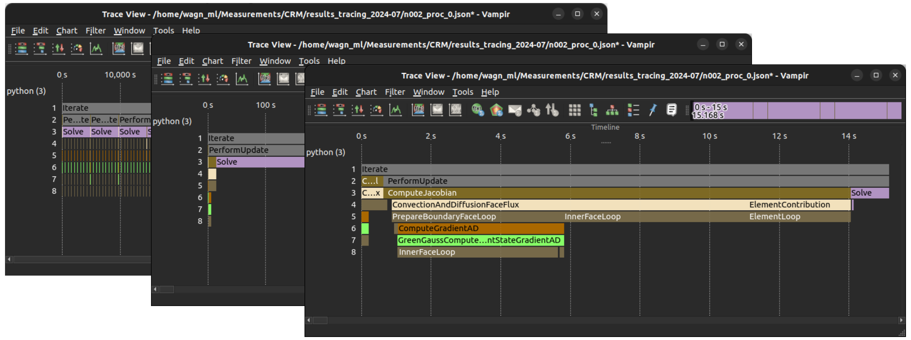


Summary

- Scaling of CODA on DLR systems CARA and CARO
- Comparison of AMD Naples and Rome
  → Important for outlook on Milan and Genoa
- Tools with various levels of detail assist in
  - runtime comparisons, regression tests
  - View component behavior
  - identify bottlenecks, validate improvements
- More to come in the next talk …







Acknowledgments

Funded by the European Union. This work has received funding from the European High Performance Computing Joint Undertaking (JU) and Germany, Italy, Slovenia, Spain, Sweden, and France under grant agreement No 101092621.

The authors gratefully acknowledge the scientific support and HPC resources provided by the German Aerospace Center (DLR). The HPC system CARA is partially funded by “Saxon State Ministry for Economic Affairs, Labour and Transport“ and „Federal Ministry for Economic Affairs and Climate Action”. The HPC system CARO is partially funded by “Ministry of Science and Culture of Lower Saxony“ and „Federal Ministry for Economic Affairs and Climate Action”.

CODA is the computational fluid dynamics (CFD) software being developed as part of a collaboration between the French Aerospace Lab ONERA, the German Aerospace Center (DLR), Airbus, and their European research partners. CODA is jointly owned by ONERA, DLR and Airbus.