Ref. Ares(2023)7688937 - 13/11/2023

HORIZON-EUROHPC-JU-2021-COE-01

EXCELLERAT P2

The European Centre of Excellence for Engineering Applications

Project Number: 101092621

D2.4

Report on the CODA Application Use Case

The EXCELLERAT P2 project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 101092621. The JU receives support from the European Union's Horizon Europe research and innovation programme and Germany, Italy, Slovenia, Spain, Sweden and France.

Work Package:	2	Use-Case Execution
Author(s):	Michael Wagner	DLR
Approved by	Executive Centre Management	10.11.2023
Reviewer	Damien Dosimont	BSC
Reviewer	Adam Peplinski	KTH
Dissemination Level	Public	

Date	Author	Comments	Version	Status
2023-10-24	Michael Wagner	Initial draft	V0.1	Draft
2023-11-01	Michael Wagner	Updated draft after first review round	V0.2	Draft
2023-11-08	Michael Wagner	Final version	V1.0	Final

List of abbreviations

CFD	Computational fluid dynamics
CRM	Common Research Model
FSDM	FlowSimulator DataManager
GASPI	Global Address Space Programming Interface
MDA	multi-disciplinary analysis
MPI	Message Passing Interface
HPC	High performance computing
RANS	Reynolds-averaged Navier-Stokes

Executive Summary

This document presents the progress made in the CODA Application Use Case UC-1 within reporting period 1 covering the first year of the EXCELLERAT P2 project. Based on the detailed roadmap of the workflow development defined in deliverable D2.1, the workflow of the use case is summarised and the achieved progress with respect to the defined workflow, objectives and success criteria is presented.

In summary, the workflow development for UC-1 has progressed according to the schedule defined in deliverable D2.1. Work has been performed on the individual tasks planned for the first year of the project.

Table of Contents

Table of Figures

1 Use Case Introduction

Computational fluid dynamics (CFD) simulations are an increasingly important part of aircraft design. They reduce cost and time of aircraft development and accelerate the introduction of progressive technologies and improvements. Moreover, high-precision CFD simulations are inevitable for assessing future aircraft designs by providing reliable insight into new aircraft technologies and reaching the best overall aircraft performance. They allow to design quieter, safer, and more fuel-efficient planes.

This use case simulates steady airflow at subsonic speed and computes typical characteristics like air velocity and direction, pressure and turbulence (Figure 1). For the use case, the CODA CFD software, in conjunction with the FlowSimulator framework, solves the Reynolds-averaged Navier-Stokes equations (RANS) with a Spalart-Allmaras one-equation turbulence model in its negative form (SAneg). It uses a second-order finite-volume spatial discretisation with an implicit Euler time integration.

The inputs of the use case are aircraft geometries for commercial aviation in the form of unstructured meshes with varying sizes ranging from tens of millions to several billions of elements; typically, with six degrees of freedom per element (ranging up to several hundreds). The aircraft geometries include public models such as the NASA Common Research Model (CRM) and internal models from DLR and Airbus.

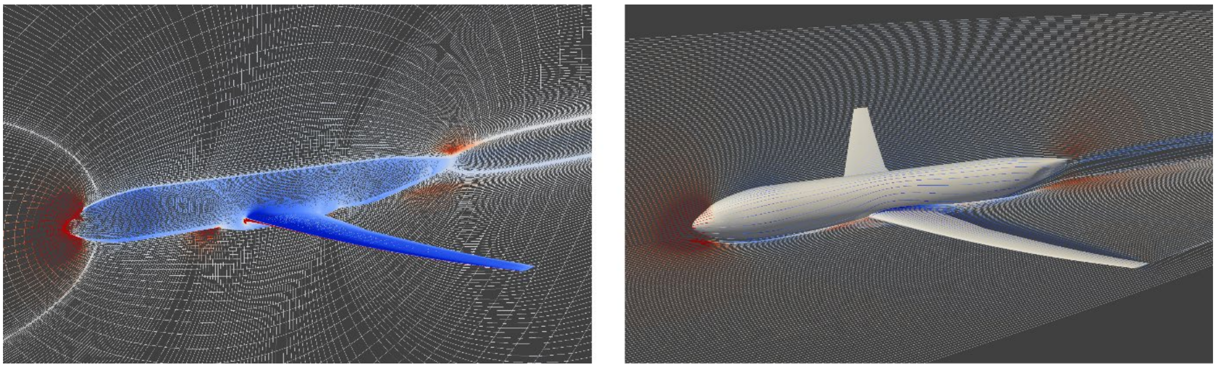

Figure 1: Visualisation of an exemplary use case simulation: aircraft configuration with mesh (left) and airflow around wing and fuselage (right); both with air pressure as colour gradient.

2 Objectives of the Use Case

The objective of the use case is to evaluate and improve the CODA CFD software and the FlowSimulator framework with respect to node and system-level performance and increase their scalability. EXCELLERAT P2 provides the opportunity for early access to pre- and near-exascale architectures and allows to verify our extrapolations from previous strong-scaling tests, which indicated pre-exascale readiness for CODA and FlowSimulator. Real near-exascale testing will show which further performance optimisation is necessary to reach this goal.

Planned external aircraft aerodynamics simulations will require large-scale runs in the range of pre-exascale systems for research and production as well as large sets of medium-scale runs for aerodynamics data production (ensemble runs). Therefore, with the help of the use case, we strive to evaluate and extend the scalability of CODA and FlowSimulator to enable research and production runs at new levels as well as further improve the efficiency of both to allow for aerodynamics data production at medium-scale with high efficiency to reduce time-to-solution and energy consumption on the HPC systems.

3 Workflow Description

The workflow for the use case consists of three software components: the CFD software CODA, the FlowSimulator framework and the sparse linear systems solver Spliss.

CODA is the 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 a next-generation CFD solver for aerodynamic simulations of fully equipped aircraft and is set to replace its predecessor, the TAU CFD package [1], in production in the European aircraft industry and research organizations. Today, TAU has been in production in the European aircraft industry, research organizations and academia for more than 20 years and was, for instance, used for the Airbus A380 and A350 wing design.

The CODA CFD software sets its focus on, first, a flexible and comprehensive parallelization concept suited for current and future HPC systems and, second, on algorithmic efficiency using strong implicit solvers, higher-order spatial discretization via the Discontinuous Galerkin method featuring hp-adaptation in addition to finite volumes with maximum code share, and seamless integration into Python-based multi-disciplinary process chains via FlowSimulator. For that, CODA uses classical domain decomposition to utilise distributed-memory parallelism via Message Passing Interface (MPI) and, additionally, the Global Address Space Programming Interface (GASPI) implementation GPI-2 as an alternative to MPI, which allows for efficient one-sided communication to reduce network traffic and latency. For both, CODA supports overlapping of halo-data communication with computation to hide network latency and further increase scalability. Besides classical domain decomposition, CODA employs a hybrid two-level parallelization to utilize shared-memory parallelism for multi- and many-core architectures. CODA implements sub-domain decomposition, where each domain is further partitioned into sub-domains, each of which being processed by a dedicated software thread that is mapped one-to-one to a hardware thread to maximise data locality. The hybrid approach allows utilising all layers of parallelism and providing a flexible adaption to different hardware architectures [2].

The CODA CFD software is a part of the multi-disciplinary analysis (MDA) framework FlowSimulator [3], which provides plug-ins for all steps of a full aircraft simulation as well as a seamless integration into multi-disciplinary simulations. In particular, CODA uses FlowSimulator's core component, the FlowSimulator DataManager (FSDM) for I/O, where various I/O libraries are supported. FSDM is an open-source software hosted by DLR. FSDM provides the FSMesh class, which is the preferred container for the exchange of data among FlowSimulator plug-ins. FSDM is MPI parallelized and an FSMesh instance is a distributed representation of the data, usually containing information on the geometry, the (computational) mesh, flow fields/solutions, and coupling strategies.

Furthermore, CODA uses the sparse linear system solver Spliss [4] for solving linear equation systems as part of implicit CFD methods. Spliss aims to provide a linear solver library that, on the one hand, is tailored to the requirements of CFD applications but, on the other hand, independent of the particular CFD solver. Focusing on the specific task of solving linear systems allows for integrating more advanced, but also more complex, hardware-specific optimisations while at the same time hiding this complexity from the CFD solver. For instance, Spliss enables the execution of the computationally intensive linear solver on GPUs without the necessity of any code adaption in CODA.

4 Progress achieved

During the reporting period, four main tasks were carried out: First, we assessed the baseline scalability of CODA and FlowSimulator on the largest available partition of DLR's main production system CARA with the NASA common research model in a strong and weak-scaling scenario. Second, we compared the performance of CODA on various upcoming CPU architectures. Third, based on an initial analysis, we identified and solved scalability issues in FlowSimulator. Fourth, we evaluated CODA with Spliss running on GPUs on the Nvidia A100 architecture and compared the performance to the DLR production systems.

Assessment of Baseline Scalability of CODA and FlowSimulator

First, we focused on evaluating the scalability of CODA on CARA with Use Case UC-1. The use case solves the Reynolds-averaged Navier-Stokes equations with a Spalart-Allmaras turbulence model in its negative form (SA-neg). The use case runs on an unstructured mesh from the NASA Common Research Model (CRM) with about 5 million points and 24 million volume elements. The mesh is a relatively small mesh, which has been chosen for a strong-scalability analysis (fixed problem size) of CODA. Production meshes are typically at least 10 times larger and accordingly achieve comparable efficiency on much higher scales. For the weak scalability analysis, we use different mesh sizes from the CRM mesh family ranging from 3 to 192 million elements and solve the use case with an according number of cores.

Figure 2 highlights the scalability of CODA on the CARA HPC system based on the AMD Naples CPU architecture. CODA achieves about 61% parallel efficiency on the largest available partition on CARA with 512 nodes and 32,768 cores in the strong-scaling scenario. In the weak-scaling scenario, a parallel efficiency of 74% was achieved on 32,768 cores.

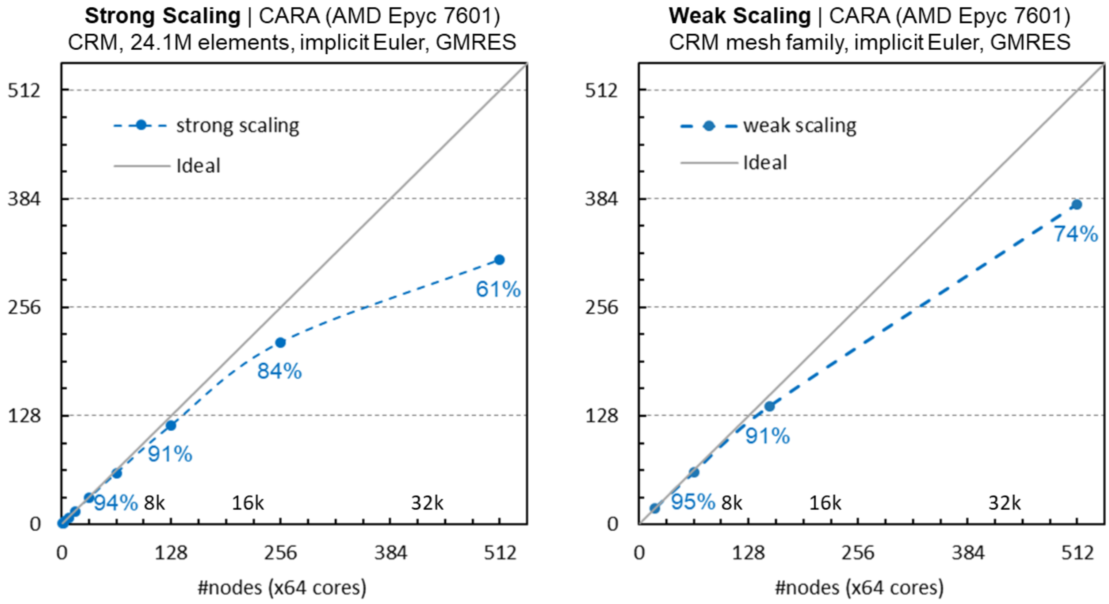

Figure 2: Strong and weak scalability of CODA with UC-1.

Comparison of CODA's Performance on Upcoming CPU Architectures

In a continuous effort to test and evaluate CODA and FlowSimulator on new CPU architectures, so far, we have evaluated the AMD Zen1, Zen2, Zen3 and Zen4 architecture, the Intel Icelake architecture and the ARM-based Graviton2 and Graviton3 architecture. For the evaluation, we use standardised benchmarks and a containerised version of CODA and FlowSimulator, including the use case, on resources in the Germany-based AWS cloud. These measurements allow us to adapt CODA to new architectures during the early-access phase and evaluate which systems offer best performance ahead of deployment to new full-scale HPC systems as well as provide valuable insight for designing DLR's own future HPC systems.

Scalability Improvement of FlowSimulator

Based on an initial analysis, we improved FlowSimulator's support for large meshes and large core counts. We identified and solved multiple issues in the mesh partitioning stage, which consists of a fast pre-partitioner based on recursive coordinate bisection (RCB) and a following graph-partitioner such as ParMETIS. In the FlowSimulator implementation of an RCB, we solved an integer overflow and an out-of-memory error for large meshes (more than 1 billion elements). In the graph-partitioner we solved an out-of-memory error in the graph-extraction phase for large core counts.

In combination with other improvements, we overcame a previous scalability limit at about 8,192 MPI processes that limited the execution of simulations with more than 32,768 cores on the DLR production systems CARA and CARO. We are now able to run successfully on 131,072 cores (the largest partition of the CARO system). In addition to increasing the scalability of FlowSimulator, the pre-processing stage was also drastically accelerated.

Evaluation on Nvidia A100 Architecture

A significant part of computational fluid dynamics simulations is the solving of large sparse linear equation systems resulting from the discretisation of the Reynolds-averaged Navier-Stokes (RANS) equations for implicit time integration methods. CODA computes these linear equation systems via the sparse linear systems solver Spliss. Besides leveraging a wide range of available HPC technologies such as hybrid CPU parallelisation, Spliss allows offloading the computationally intensive linear solver to GPU accelerators while at the same time hiding this complexity from the CFD solver. We evaluated the entire workflow on a GPU system, whereas FlowSimulator and CODA are executed on the CPU part and the linear solver on GPUs.

When comparing the CPU system CARO (AMD Rome) and the Nvidia A100 GPU system Juwels Booster at Jülich Supercomputing Center, the Use Case UC-1 achieves a speedup of up to 8.4 in a node-wise comparison and a speedup of up to 1.9 in a power-equated comparison. The improvements made to establish multi-GPU capabilities in Spliss, which allow for efficient and scalable usage of large GPU systems, and an evaluation of performance and scalability on CPU and GPU systems were published recently [5]. With these improvements, CODA is able to support European Nvidia-based GPU systems such as Leonardo or MareNostrum 5. We currently evaluate the support for AMD-based GPU systems such as LUMI-G.

5 Next Steps

According to the schedule defined in deliverable D2.1, we continue with two tasks: First, the continuous analysis of CODA and FlowSimulator efficiency and capability improvements and, second, the continuous improvement of CODA and FlowSimulator efficiency and capabilities.

The next steps are, first, the evaluation of CODA and FlowSimulator with UC-1 on further HPC systems such as DLR's second production system CARO based on the AMD Rome architecture. Second, we continue the testing and evaluation of upcoming CPU architectures via the AWS cloud. Third, to further increase the scalability of the FlowSimulator pre-processing stage, we plan to develop a hierarchical graph-partitioning approach. Fourth, we evaluate the support of Spliss for further GPU systems, such as systems based on the Nvidia H100 architecture and AMD GPU systems like LUMI-G.

6 References

[1] D. Schwamborn, T. Gerhold and R. Heinrich, "The DLR TAU Code: Recent Applications in Research and Industry", In Proc. of the European Conference on Computational Fluid Dynamics, ECCOMAS CFD, 2006. https://elib.dlr.de/22421

[2] M. Wagner, „Scalability Evaluation of the CFD Solver CODA on the AMD Naples Architecture", In Sustained Simulation Performance, pp. 95-106, 2023. https://doi.org/10.1007/978-3-031-18046-0_7

[3] M. Meinel and G. Einarsson, "The FlowSimulator Framework for Massively Parallel CFD Applications", In PARA 2010. https://elib.dlr.de/67768

[4] O. Krzikalla, A. Rempke, A. Bleh, M. Wagner and T. Gerhold, "Spliss: A Sparse Linear System Solver for Transparent Integration of Emerging HPC Technologies into CFD Solvers and Applications", In New Results in Numerical and Experimental Fluid Mechanics XIII, pp. 635-645, 2021. https://doi.org/10.1007/978-3-030-79561-0_60

[5] J. Mohnke and M. Wagner, "A Look at Performance and Scalability of the GPU Accelerated Sparse Linear System Solver Spliss", In Euro-Par 2023: Parallel Processing. LNCS, vol 14100, 2023. https://doi.org/10.1007/978-3-031-39698-4_43