

Article

Performance Comparison of CFD Microbenchmarks on Diverse HPC Architectures

Flavio C. C. Galeazzo ^{1,*} , Marta Garcia-Gasulla ^{2,*} , Elisabetta Boella ³ , Josep Pocurull ² , Sergey Lesnik ⁴ , Henrik Rusche ⁴ , Simone Bnà ⁵ , Matteo Cerminara ⁶ , Federico Brogi ⁶ , Filippo Marchetti ³, Daniele Gregori ³, R. Gregor Weiß ¹  and Andreas Ruopp ¹ 

- ¹ High Performance Computing Center Stuttgart (HLRS), University of Stuttgart, Nobelstr. 19, 70569 Stuttgart, Germany; gregor.weiss@hlrs.de (R.G.W.); ruopp@hlrs.de (A.R.)
- ² Barcelona Supercomputing Center (BSC), Pl. Eusebi Guell 3, 08034 Barcelona, Spain; josep.pocurull@bsc.es
- ³ E4 Computer Engineering, Via Martiri della Libertà 66, 42019 Scandiano, Italy; elisabetta.boella@e4company.com (E.B.); filippo.marchetti-ext@e4company.com (F.M.); daniele.gregori@e4company.com (D.G.)
- ⁴ WIKKI GmbH, Ziegelbergsweg 68, 38855 Wernigerode, Germany
- ⁵ HPC Department, CINECA, Bologna Office, Via Magnanelli 6/3, 40133 Bologna, Italy; simone.bna@cineca.it
- ⁶ Istituto Nazionale di Geofisica e Vulcanologia, Via Cesare Battisti 53, 56125 Pisa, Italy; matteo.cerminara@ingv.it (M.C.); federico.brogi@ingv.it (F.B.)
- * Correspondence: flavio.galeazzo@hlrs.de (F.C.C.G.); marta.garcia@bsc.es (M.G.-G.)

Abstract: OpenFOAM is a CFD software widely used in both industry and academia. The exaFOAM project aims at enhancing the HPC scalability of OpenFOAM, while identifying its current bottlenecks and proposing ways to overcome them. For the assessment of the software components and the code profiling during the code development, lightweight but significant benchmarks should be used. The answer was to develop microbenchmarks, with a small memory footprint and short runtime. The name microbenchmark does not mean that they have been prepared to be the smallest possible test cases, as they have been developed to fit in a compute node, which usually has dozens of compute cores. The microbenchmarks cover a broad band of applications: incompressible and compressible flow, combustion, viscoelastic flow and adjoint optimization. All benchmarks are part of the OpenFOAM HPC Technical Committee repository and are fully accessible. The performance using HPC systems with Intel and AMD processors (x86_64 architecture) and Arm processors (aarch64 architecture) have been benchmarked. For the workloads in this study, the mean performance with the AMD CPU is 62% higher than with Arm and 42% higher than with Intel. The AMD processor seems particularly suited resulting in an overall shorter time-to-solution.

Keywords: CFD; OpenFOAM; computer performance; benchmark



Citation: Galeazzo, F.C.C.; Garcia-Gasulla, M.; Boella, E.; Pocurull, J.; Lesnik, S.; Rusche, H.; Bnà, S.; Cerminara, M.; Brogi, F.; Marchetti, F.; et al. Performance Comparison of CFD Microbenchmarks on Diverse HPC Architectures. *Computers* **2024**, *13*, 115. <https://doi.org/10.3390/computers13050115>

Academic Editor: Yan Liu

Received: 5 March 2024

Revised: 16 April 2024

Accepted: 23 April 2024

Published: 7 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The CFD software OpenFOAM [1] is widely used both in industry and academia. It combines the needs of both worlds, where support for robust numerics and complex geometries are required in real-world engineering practice while a broad range of physical models can be designed, developed, and investigated in a user-friendly environment to extend the numerical modeling capabilities beyond industrial problems. Therefore, OpenFOAM is an object-oriented library that enables extensible ‘user coding’ to re-use individual components of meshing, linear system solvers, and discretization schemes in top-level solvers of complex physical models mapping partial differential equations in software [2]. OpenFOAM’s applicability ranges from aerodynamic [3] and fluid flow [4] over manufacturing [5] to complex chemical engineering problems [6]. Proven useful on a range of compute resources to date, bottlenecks of OpenFOAM arise in current HPC systems identified by the exaFOAM project [7]. Given the high demand of OpenFOAM’s versatility

the exaFOAM project aims at overcoming the current bottlenecks through algorithmic improvements and enhanced HPC scalability.

To demonstrate the improvements in performance and scalability of OpenFOAM, extreme-scale demonstrators (HPC Grand Challenges) and industrial-ready applications have been prepared. However, these test cases are too computationally expensive to assess the software components and the code profiling during the development phases. To solve this problem, less computationally demanding benchmarks have been derived, called microbenchmarks (MBs), with a smaller memory footprint and shorter runtime. The name microbenchmark does not mean that they have been prepared to be the smallest possible test cases, as they have been developed to fit in an HPC compute node, which usually has dozens of compute cores.

The microbenchmarks have been designed to be performance proxies of the HPC Grand Challenges and the industrial test cases, representing the full simulation but demanding far less computational time. All benchmarks are part of the OpenFOAM HPC Technical Committee repository [8] and are publicly available. This set of benchmarks is built as an open and shared entry point to the OpenFOAM community, which can be used as a reference to compare the solution of different physical problems in different hardware architectures.

As representatives of the current generation of HPC architectures, CPUs from Intel, AMD, and Arm have been used in the tests. To directly compare the performance of different test cases in these architectures, a new metric is proposed (FVOPS, finite volumes solved per second). This metric facilitated the comparison between the architectures and the performance with different grid sizes. The most distinct difference between the hardware platforms is the larger amount of L3 cache of the AMD system (see Section 2), which proved to be a differential and resulted in an overall shorter time-to-solution compared to Intel and Arm systems.

2. Materials and Methods

The performance of OpenFOAM versions v2212 and foam-extend 4.1 on different HPC architectures and processor types has been compared, in particular with the x86_64 and aarch64 architectures. In the case of x86_64, both Intel and AMD processors were employed, while in the case of aarch64, Arm processors were used. Table 1 shows a summary of the HPC architectures and processor types utilized for the tests. The manufacturer of the Intel server is Intel Corporation, Santa Clara, CA, USA; of the Arm server is Foxconn, New Taipei City, Taiwan; and of the AMD server is Supermicro, San Jose, CA, USA.

Table 1. Summary of the architectures and processor types utilized for the tests.

Architecture	CPU Model	Frequency	Cores/Node	Memory/Node	L3 Cache/Node
x86_64	Intel(R) Xeon(R) Gold 6226R (Cascade Lake)	2.9 GHz	32	192 GB	44 MB
x86_64	AMD EPYC 7313 16-Core Processor (Milan)	3.3 GHz	32	256 GB	256 MB
aarch64	Arm Neoverse-N1	3.0 GHz	256	512 GB	32 MB

To investigate not only the most appropriate hardware architecture, but also how each benchmark scales in each node type, all benchmarks have been deployed on all systems, and the experiments used an incremental number of MPI processes until reaching the maximum number of cores of each platform.

From the collection of microbenchmarks published by the exaFOAM project in the OpenFOAM HPC technical committee repository [8], 11 have been considered in this work, listed in Table 2 along with their solvers and information about grid characteristics. These MBs were chosen because of their importance for industry applications. Another parameter taken into account in the selection of MBs was the size of the grid. Since the MBs were run on one compute node using from 2 to 32 MPI tasks, the problem size should fit on the node

memory and the computation should be carried out in a reasonable time on the smallest number of processors considered.

For completeness, a short description of each microbenchmark follows in the next subsections. The interested reader is pointed to the OpenFOAM HPC Technical Committee repository [8] where the test cases are fully described, including detailed testing configurations, data collection, and analysis methods, along with instructions on how to run them.

Table 2. Summary of the OpenFOAM MBs considered in this study, with application fields. Mesh generation types are blockMesh (BM), snappyHexMesh (SHM) and cfMesh (CFM).

	Microbenchmarks	Top-Level Solver	Mesh Generation—Cell Count—Cell Type
MB1	Cavity 3D - incompressible laminar flow	icoFoam	BM—8 M—Hexahedra
MB2	Compressible starting square jet - compressible flow, LES turbulence modeling	rhoPimpleFoam	BM—2 M—Hexahedra
MB4	DLR-JHC burner - turbulent combustion with detailed chemistry and LES turbulence modeling	reactingFoam	BM—400 k—Hexahedra
MB5	ERCOFTAC Conical diffuser - incompressible flow, RANS turbulence modeling	simpleFoam	BM—3 M—Hexahedra
MB6	Two cylinders in line - unsteady adjoint optimization, 2D laminar flow	adjointOptimisationFoam	BM—24,500—Hexahedra
MB8	Rotating Wheel - incompressible, external flow with rotating wheels, DDES turbulence modeling	pimpleFoam	SHM—20 M—Polyhedra
MB9	High-lift airfoil - 2D compressible flow, DDES turbulence modeling	rhoPimpleFoam	SHM—20 M—Polyhedra
MB11	Pitz&Daily Combustor - turbulent combustion, LES turbulence modeling	XiFoam	BM—200 k—Hexahedra
MB12	Model Wind Farm - incompressible flow, wind turbines using actuator disc model, LES turbulence modeling	pimpleFoam	BM—8 M—Hexahedra
MB17	1D Aeroacoustic Wave Train - 1D compressible laminar flow	rhoPimpleFoam 1D	BM—0.05 M—Hexahedra
MB19	Viscoelastic polymer melt flow - 3D laminar, viscoelastic flow	viscoelasticFluidFoam	CFM—1 M—Polyhedra

2.1. MB1 Cavity 3D

The flow in a cavity, which is driven by the upper cavity boundary (lid), is one of the most basic and widely spread test cases for the CFD programs (see Figure 1). It is easy to set up and validate because of the simple geometry. According to AbdelMigid et al. [9], the setup has been used by researchers for more than 50 years to benchmark fluid dynamics codes. The available literature on the topic includes numerical experiments for a wide range of computational algorithms and flow regimes, e.g., Ghia et al. [10] provide comprehensive results for the Reynolds number ranging from 100 to 10,000. In most cases, the numerical setups are 2D whereas experimental measurements are carried out in 3D as performed by Koseff et al. [11].

The present setup is based on the white paper by Bnà et al. [12]; additional information can be found in Brogi et al. [13]. Thus, scaling results are available.

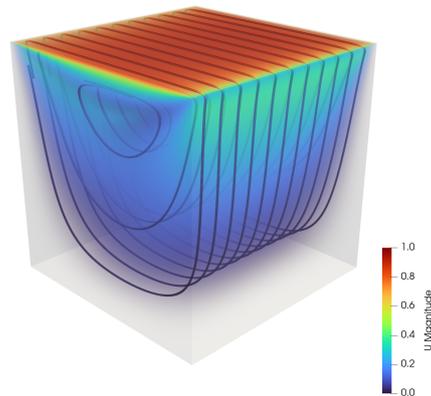


Figure 1. Volume rendering colored by velocity magnitude of the lid-driven flow in a cavity.

2.2. MB2 Compressible Starting Square Jet

A high-speed injection of a warm gas (330 K) into a static atmosphere with a slightly lower temperature (300 K) is considered (see Figure 2). For simplicity, the jet fluid enters the atmospheric box through a square inlet (0.0635 m) resolved with a homogeneous structured mesh (no grid stretching). Given the low gas viscosity (1.8×10^{-5} Pa s) and high velocity (364 m/s) of the injected fluid, the jet flow is unsteady and compressible, characterized by high Mach number (>1 locally) and Reynolds number. Non-linear instabilities in the jet shear layer (like Kelvin–Helmholtz), vortexes, pressure waves and a turbulent buoyant plume typically develop with these conditions. In the simulations, the jet fluid, as it enters the atmospheric box, forms a vortex and pushes the static air outwards producing an intense pressure wave followed by smaller transients that propagate radially (see Figure 2). More information can be found in Brogi et al. [13].

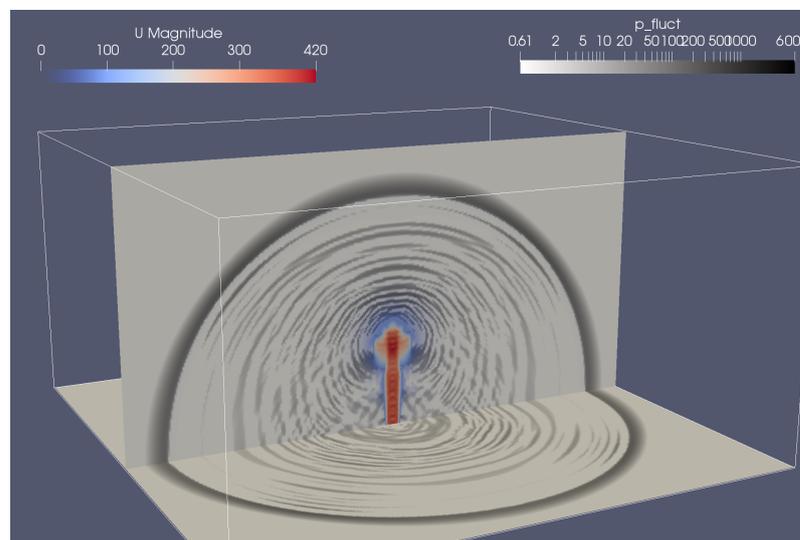


Figure 2. Screenshot of LES simulation of the supersonic starting jet performed with rhoPimpleFOAM solver. The magnitude of the velocity field (in color) and pressure fluctuations (black and white) are shown.

2.3. MB4 DLR-JHC Burner

The combustion chamber investigated by the DLR Institute of Combustion Technology is simulated. Detailed information about the experimental setup is reported in the literature [14]. A pre-chamber exists below a top vessel. The top vessel is considered in the current analysis; the pre-chamber is neglected. The top of the considered chamber is open. A jet of fuel is introduced in the vessel; the region is initially filled with vitiated air

coming from a preliminary combustion process between air and H_2 which takes place in the pre-chamber. The vitiated air is accounted as a homogeneous mixture because of the perfect mixing occurring in the pre-chamber. In the top vessel, the hot vitiated air reacts with the injected methane and generates a steady lift flame. Figure 3 shows simulation results. Experiments are publicly available from the DLR Institute of Combustion Technology. The $L \times W \times H$ rectangular-shaped region is three-dimensional. Due to the introduced simplification, a quarter of the domain can be considered; symmetry planes are identified on two of the resulting sides. A small centered nozzle injects the fuel, while air with lean premixed hydrogen combustion products is introduced at the bottom of the mesh in the surrounding region.

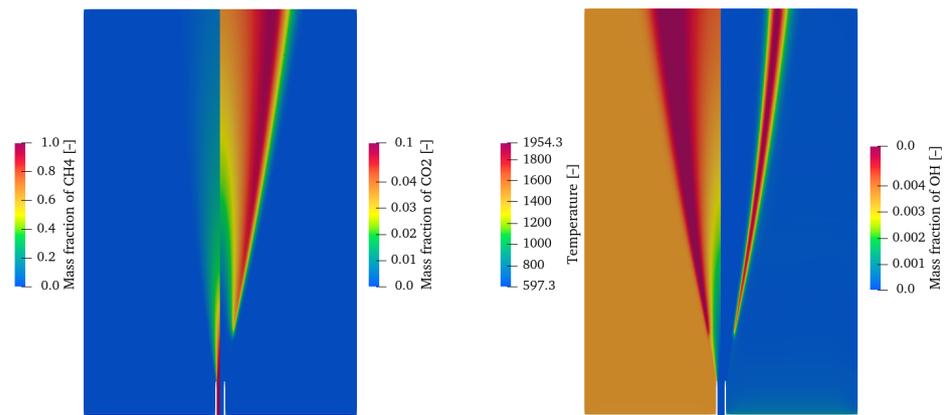


Figure 3. Simulation results of the DLR-JHC burner showing mass fractions of CH_4 , CO_2 , OH and temperature in K.

2.4. MB5 ERCOFTAC Conical Diffuser

The case stems originally from ERCOFTAC (European Research Community On Flow, Turbulence and Combustion). A detailed description is available [15] and experimental data [16] as well. A diffuser is designed to reduce the flow velocity and therefore increase the fluid pressure without causing significant pressure loss; see Figure 4. This is carried out by increasing the cross-section area, e.g., by introducing a conical segment. The latter is characterized by a so-called divergence angle, which is the angle between the opposite walls in the axial cross-section through the conical part. Due to this geometry, the flow tends to separate at the diffuser wall, which is undesirable since it causes losses. However, the behavior can be counteracted by the inclusion of a swirling flow component in the flow entering the diffuser. But if the swirl is too strong, a recirculation along the center axis occurs, which reduces the pressure recovery.

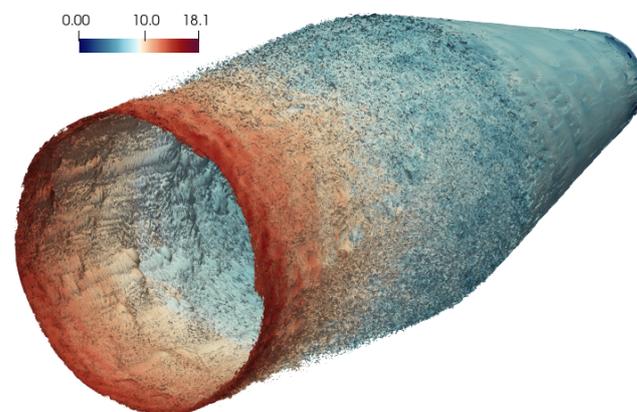


Figure 4. Isovolume of the vorticity magnitude colored by velocity magnitude of the ERCOFTAC conical diffuser, showing that the velocity field forms eddies in the boundary layer.

2.5. Mb6 Two Cylinders in Line

This microbenchmark case deals with the flow around two equal-sized cylinders in a tandem (in-line) arrangement [17,18]. In this benchmark, a von Karman vortex street appears downstream of the cylinders when the spacing L between them (diacenter) is greater than $3D$, where D is the cylinder diameter [17]; see Figure 5. Its goal is to showcase the memory savings when using a (lossy) compression scheme for computing sensitivity derivatives for unsteady flows when compared with the full storage of the flow time history. When used with unsteady flows, the adjoint equations are integrated backward in time, requiring the instantaneous flow fields to be available at each time step of the adjoint solver, which noticeably increases storage requirements in large-scale problems. To avoid extreme treatments, such as the full storage of the computed flow fields or their re-computation from scratch during the solution of the adjoint equations, and to reduce the overhead re-evaluation incurred by the widely used checkpointing technique [19], lossy compression techniques are used. Using lossy compression, the re-computation cost is expected to be reduced by efficiently compressing the checkpoints, so that more can fit within the available memory, or even eliminate the need for checkpointing (and flow re-computations) if the entire compressed flow series can be stored in memory. The compression strategies will be assessed based on their effectiveness in data reduction, computational overhead, and accuracy of the computed sensitivity derivatives.

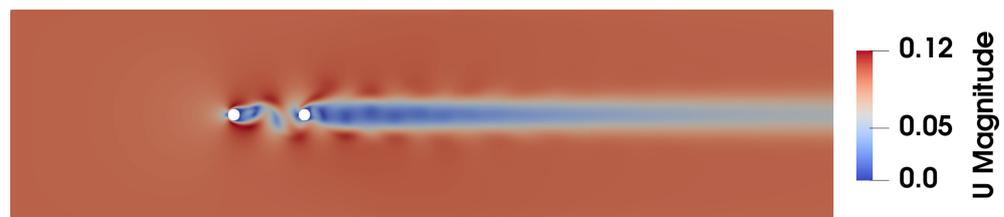


Figure 5. Two cylinders in tandem with an instantaneous depiction of the magnitude of the primal velocity.

2.6. MB8 Rotating Wheel

This microbenchmark aims to represent an industrial application on a small scale using the arbitrary mesh interface (i.e., ACMI) in OpenFOAM. The case consists of a single isolated rotating front left wheel of the DrivAer full-scale car model in the variant introduced by Ford [20]; see Figure 6. The CAD data were taken from case 2 of the 2nd automotive CFD prediction workshop [21] and the parts used for this microbenchmark are shown in Figure 6. The positioning of the wheel axis is inherited from the full-scale case. The domain inlet is positioned at $x = -2500$ mm, such that the domain lengths upstream and downstream of the wheel are approximately $3.4D$ and $9.7D$ with D being the outer tire diameter. The domain size is chosen arbitrarily but ensures adequate distance to boundaries for numerical stability and resolution of the turbulent flow downstream of the wheel. Figure 7 shows velocity contours of a snapshot after the simulation of about 10 full rotations.



Figure 6. DrivAer geometry with the indication of the selected wheel.

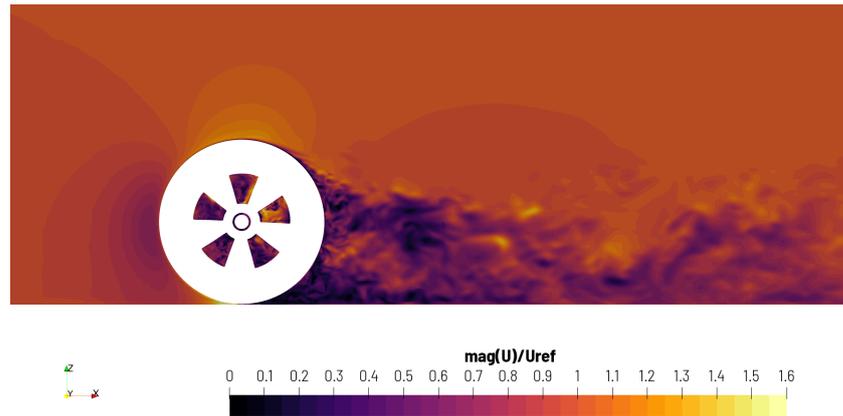


Figure 7. Velocity contours of a snapshot after about 10 full rotations.

2.7. MB9 High-Lift Airfoil

The MB9 microbenchmark is intended for preparatory work building up to the simulation of the exaFOAM Grand Challenge test case of the High Lift Common Research Model (CRM-HL) [22], a full aircraft configuration with deployed high-lift devices using wall-modelled LES (WMLES). MB9 is a two-dimensional, three-element high-lift wing configuration. The configuration is simulated with the scale-resolving IDDES model [23], which exhibits WMLES functionality in regions of resolved near-wall turbulence (e.g., on the suction side of the main element). After a review of various public-domain test cases, the 30P30N case was selected, which was studied extensively with WMLES in the 4th AIAA CFD High Lift Prediction Workshop (HLPW-4) and has experimental data available [24]. An instantaneous snapshot of the results is shown in Figure 8.

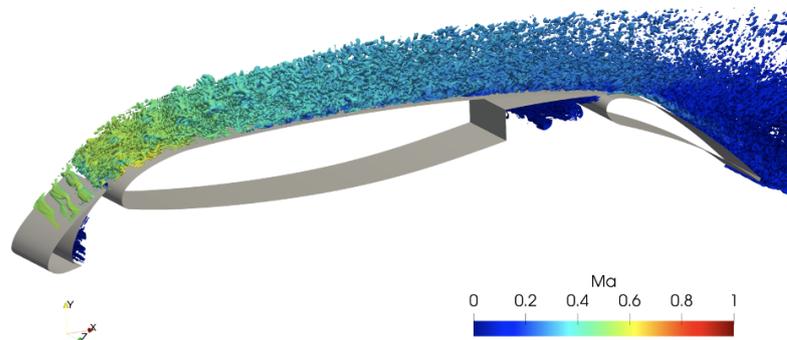


Figure 8. Instantaneous snapshot of iso-surfaces of the Q criterion (with $Q = 5 \times 10^6 \text{ s}^{-2}$) colored by local Mach number, showing the flow over the high-lift airfoil.

2.8. MB11 Pitz&Daily Combustor

The case is based on the experiment carried out by Pitz and Daily [25], who measured a combustion flow formed at a backward-facing step. The goal of the work was to study the turbulent shear layer during a combustion process in conditions similar to those of industrial and aircraft gas turbine combustors. The premixed combustion is stabilized by recirculation of hot products which are mixed with cold reactants in a turbulent shear layer. The setup with a backward-facing step is one of the simplest configurations reproducing these conditions; see Figure 9.

The dimensions used in the numerical setup are kept the same as in the OpenFOAM tutorial cases; however, they are in a 3D configuration. These deviate from the geometry described in the experimental setup by several mm. Furthermore, there are no dimensions of the contraction section available in the original paper [25]. It is assumed that these deviations do not introduce a significant influence on the final results. An additional geometry feature that matters is the ramp upstream of the test section. Calculations with a

short straight inlet section resulted in a stable flow demonstrating no fluctuations. This is probably due to the insufficient development of the boundary layer up to the edge of the step. Thus, the inclusion of the ramp is essential for the results.

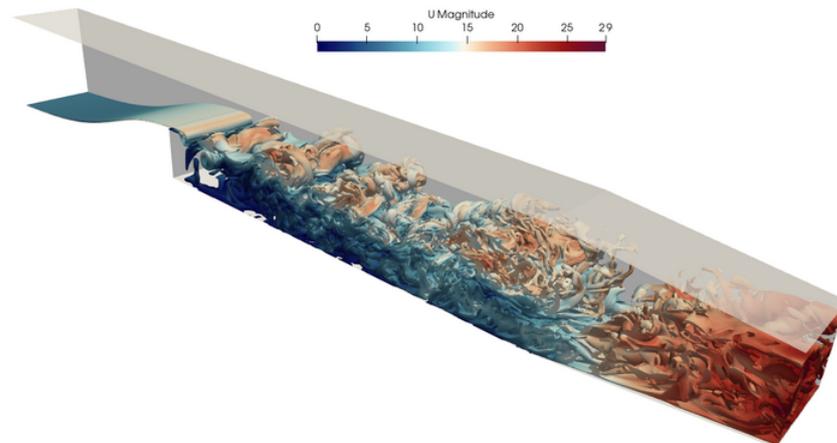


Figure 9. Isovolume of the vorticity magnitude colored by velocity magnitude of the Pitz&Daily combustor.

2.9. MB12 Model Wind Farm

Large-eddy simulations (LES) are a prominent tool for performing high-fidelity simulations of wind turbine wakes and wind farm flows. LES can capture the three-dimensional unsteady character of the flow around wind turbines and the wake flow interaction that occurs in wind farms. However, the influence of the wind turbines on the flow has to be modeled, as is still not feasible to use body-fitted meshes or immersed boundary methods to fully resolve the blades.

To study the turbine wakes generated by wind turbines, LES simulations using the Actuator Disc Model (ADM) are compared with the wind tunnel experimental data of the Saint Anthony Falls Laboratory at the University of Minnesota. Two complementary setups are studied: a single wind turbine, described in references [26,27], and a wind farm with 30 wind turbines, described in references [28,29]. Figure 10 shows a plot of the wind farm simulation, along with the position of the 30 wind turbines that form the wind farm.

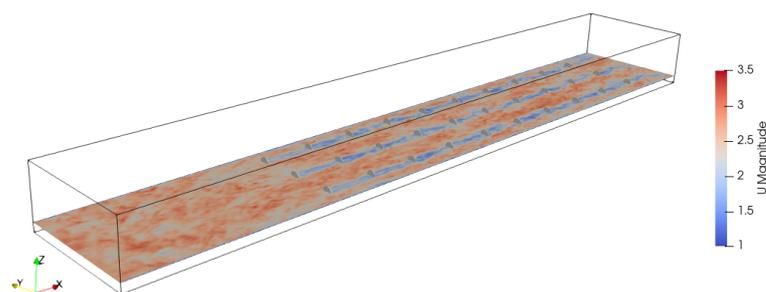


Figure 10. Contour plot of the velocity magnitude, wind farm.

2.10. MB17 1D Aeroacoustic Wave Train

This test case models a one-dimensional phenomenon solved in 1D of a periodic flow emanating from a boundary (pressure oscillation) using the OpenFOAM transient compressible equation of state in laminar flow. It is, therefore, a direct simulation of noise propagation. The aeroacoustic wave train is applied to the aerodynamics industry and is relevant to the automotive, aerospace, energy, and environmental industry sectors. The interest in HPC towards exascale is to provide a further subset of physics that can be quickly profiled and tested for scaling, and can be used to assess the solver capabilities towards the ‘fine mesh limit’.

A 2.5 m length one-dimensional domain is used for wave propagation at 3000 Hz (also corresponding to a typical frequency of aeroacoustics excitation, and within the peak human hearing range). The working fluid is air and the compressible ideal gas equation of the state is solved to simulate wave propagation. Acoustics damping is used to suppress spurious numerical artifacts and boundary wave reflection.

2.11. MB19 Viscoelastic Polymer Melt Flow

The long computational time required to perform a numerical simulation of profile extrusion forming, considering realistic (viscoelastic) constitutive models, is incompatible with usual industrial requirements. This microbenchmark case study represents a typical profile extrusion problem and aims at assessing the solver viscoelasticFluidFoam available in foam-extend 4.1. The geometry resembles a typical profile extrusion die, with a circular inlet with a radius of 12.5 mm which connects to the extruder, and a rectangular outlet that allows manufacturing a profile with a rectangular cross-section of 15 mm \times 2 mm. The middle of the channel comprises a convergent zone, which performs the transition between the circular inlet and the rectangular outlet. The numerical distribution of velocity and stress is shown in Figure 11. Due to symmetry, just a quarter of the geometry is considered for the numerical studies.

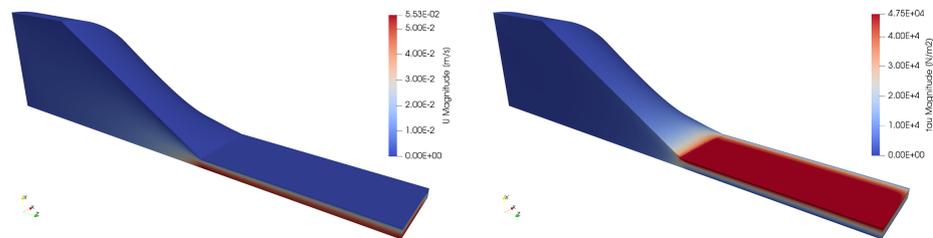


Figure 11. Simulation results showing velocity magnitude and stress tensor magnitude of the viscoelastic polymer melt flow.

3. Results

The tests have been performed using the HPC architectures reported in Table 1. From the 19 microbenchmarks provided by the exaFOAM project [7], 11 MBs have been selected (see Table 2). OpenFOAM has specific solvers for a class of physical problems to solve, and they have been chosen due to their importance for industrial and academic applications.

The majority of the benchmark runs were performed with OpenFOAM version v2212, only MB19 used the viscoelasticFluidFoam solver from foam-extend 4.1. Both versions were compiled with the GNU Compiler Collection version 8.5.0 and OpenMPI version 4.1.4. Figure 12 shows the execution time per time step or iteration versus the number of ranks (MPI tasks) for the different MBs and architectures considered. The execution time was computed excluding the first time step or iteration, as often it includes initialization operations. Each numerical experiment was repeated five times and the reported execution time is the average. By observing the results of Figure 12, it can be noticed that the execution time decreases by increasing the number of ranks. This occurs in general for all MBs and architectures with few exceptions. In the case of MB1, the execution time on x86_64 architectures decreases up to 24 cores, but then it slightly increases when running on 32 cores. A similar trend is observed on AMD for MB2, MB11, MB12, and MB19, and on Arm for MB11. Somehow, unexpected behavior is obtained on the AMD for MB4 and on the Arm for MB5. In the first case, the execution time decreases from 2 to 8 processors, but then when using 16 and 24 processors a longer execution time is obtained. In the latter case, the execution time with 8 or 16 ranks is approximately the same. Figure 12 also indicates that when a few ranks (<8) are launched, better performance in terms of execution time per iteration is obtained on AMD, where for some MBs the execution time is almost half of the other architectures. Differences tend to be less evident when employing larger numbers of ranks. Another consideration is that the performance, in terms of absolute execution time, is comparable for most MBs using Intel and Arm processors (x86_64 and aarch64 architectures).

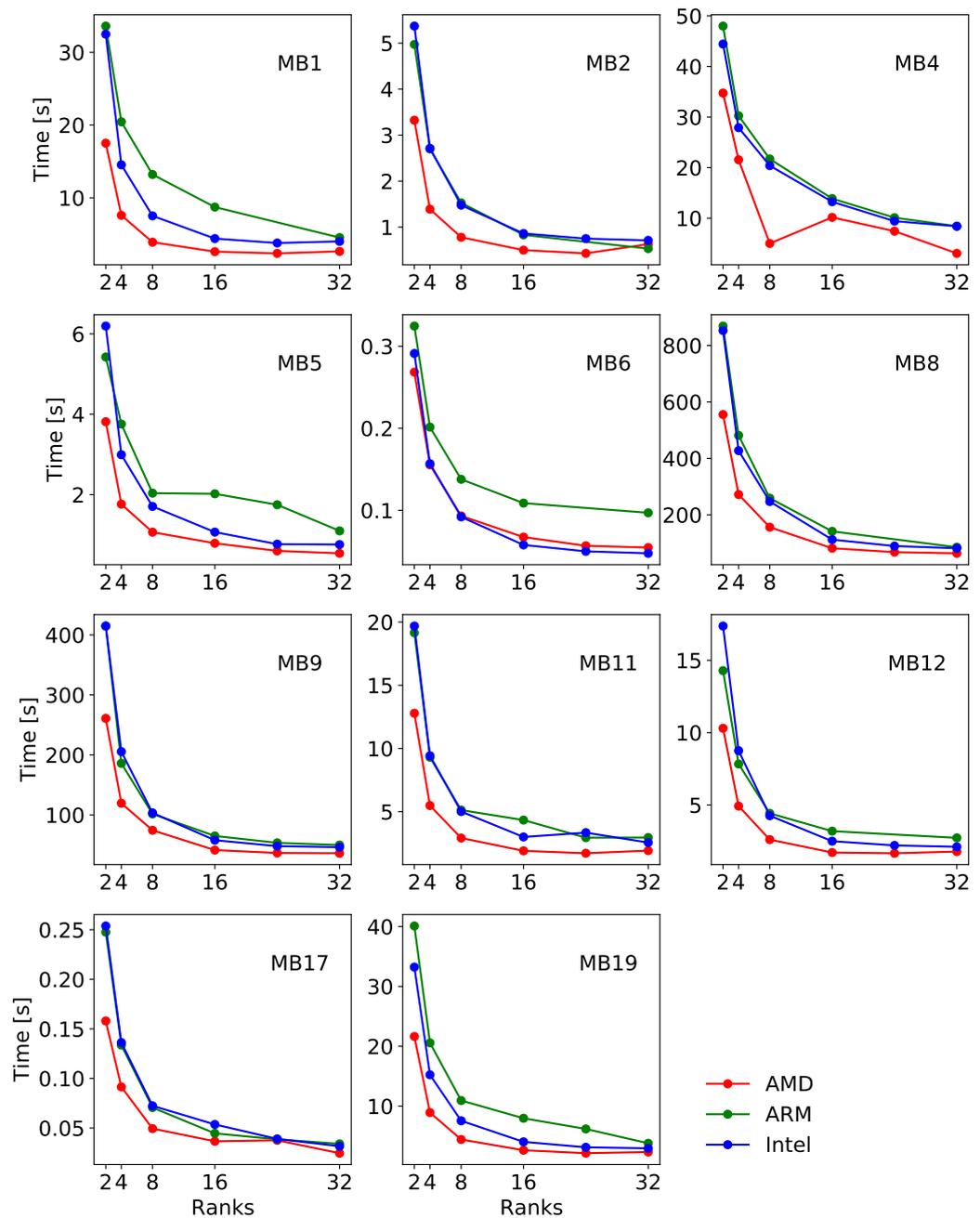


Figure 12. Execution time per OpenFOAM iteration vs. number of ranks (MPI tasks) for different microbenchmarks.

The strong scaling speedup (Figure 13) and efficiency (Figure 14) on the different node types have been calculated based on the data collected on the execution time. Several MBs show superlinear speedups on x86_64 architectures with efficiency values that exceed 1 (MBs 1, 2, 4, 5, 8, 9, 11, 12, and 19) especially when moving from two to four or eight cores. This phenomenon is more pronounced on the AMD processor. When using Arm processors, the efficiency exceeds 1 only in the case of MB9. The reason behind the superlinearity is currently being investigated. One hypothesis is that the larger L3 cache of the AMD CPU (128 MB for AMD, 22MB for Intel, and 4MB for Arm) promotes this behavior.

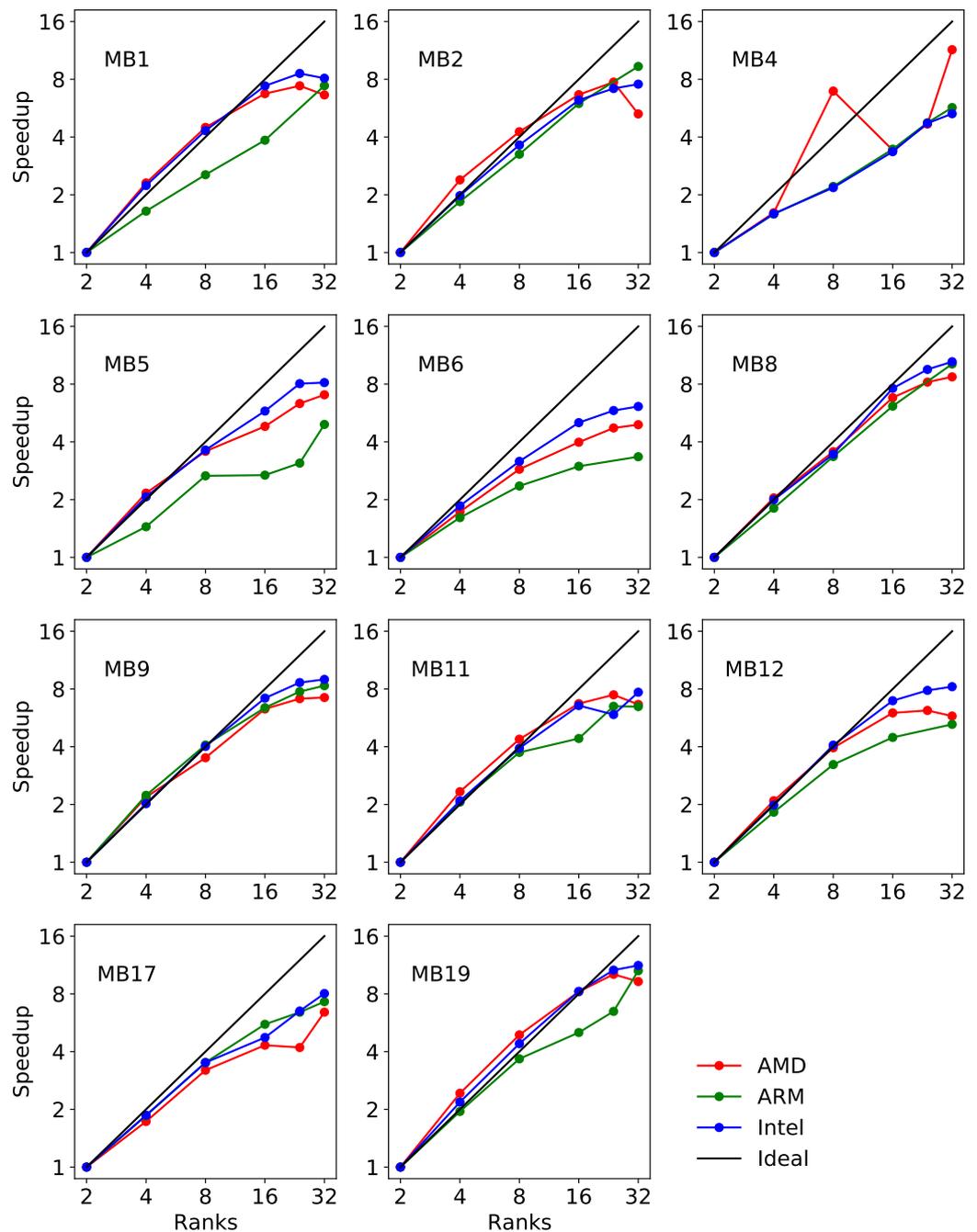


Figure 13. Speedup for different microbenchmarks in OpenFOAM strong scaling tests.

For all the MBs, we observe a departure from the ideal speedup with efficiency often dropping under 50% when launching more than eight ranks. OpenFOAM is a well-known memory-bound application; therefore, when reaching memory bandwidth saturation, launching more MPI tasks does not provide many benefits, as the CPUs are still waiting for data. In general, the x86_64 architectures seem to show better parallel efficiency on all the MBs and up to a larger number of cores, with few exceptions.

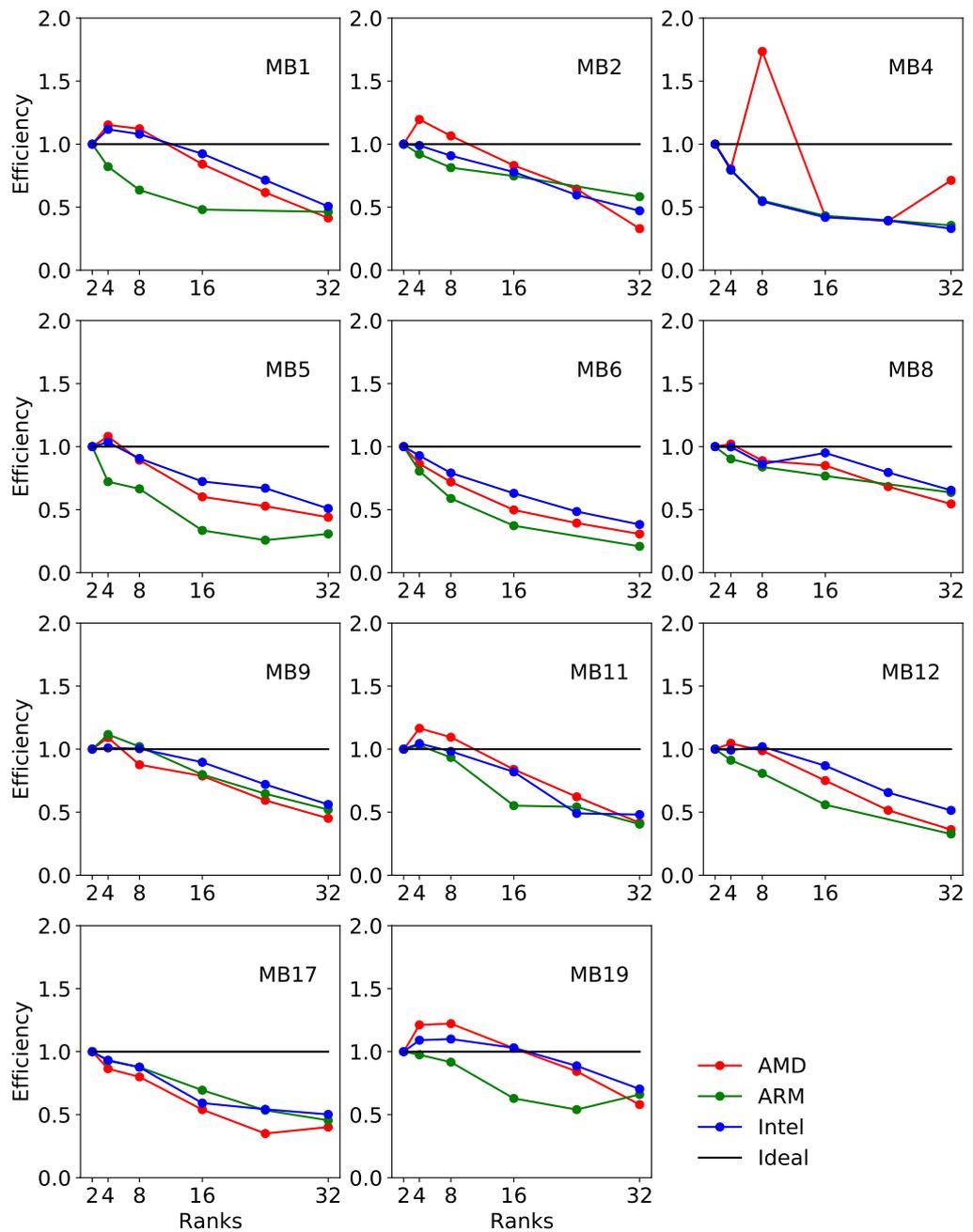


Figure 14. Efficiency for different microbenchmarks in OpenFOAM strong scaling tests. The black line represents the ideal efficiency.

Finally, Figure 15 reports the FVOPS (finite volumes solved per second) metric [30] for the same numerical experiments. This metric is defined as the number of finite volume elements in the grid divided by the execution time of a time step or iteration. The FVOPS metric depends on all parameters of the simulation, including grid size, partitioning, parallel efficiency, type of solver, and number of variables being solved. However, with fixed parameters, this metric allows the direct comparison of the performance on different systems, even with different grid sizes. The plots in Figure 15 show in many cases local maxima which indicates the optimal number of grid points per core per MB and architecture. It is interesting to notice that these local maxima occur at different values of grid element per rank when utilizing different processor types.

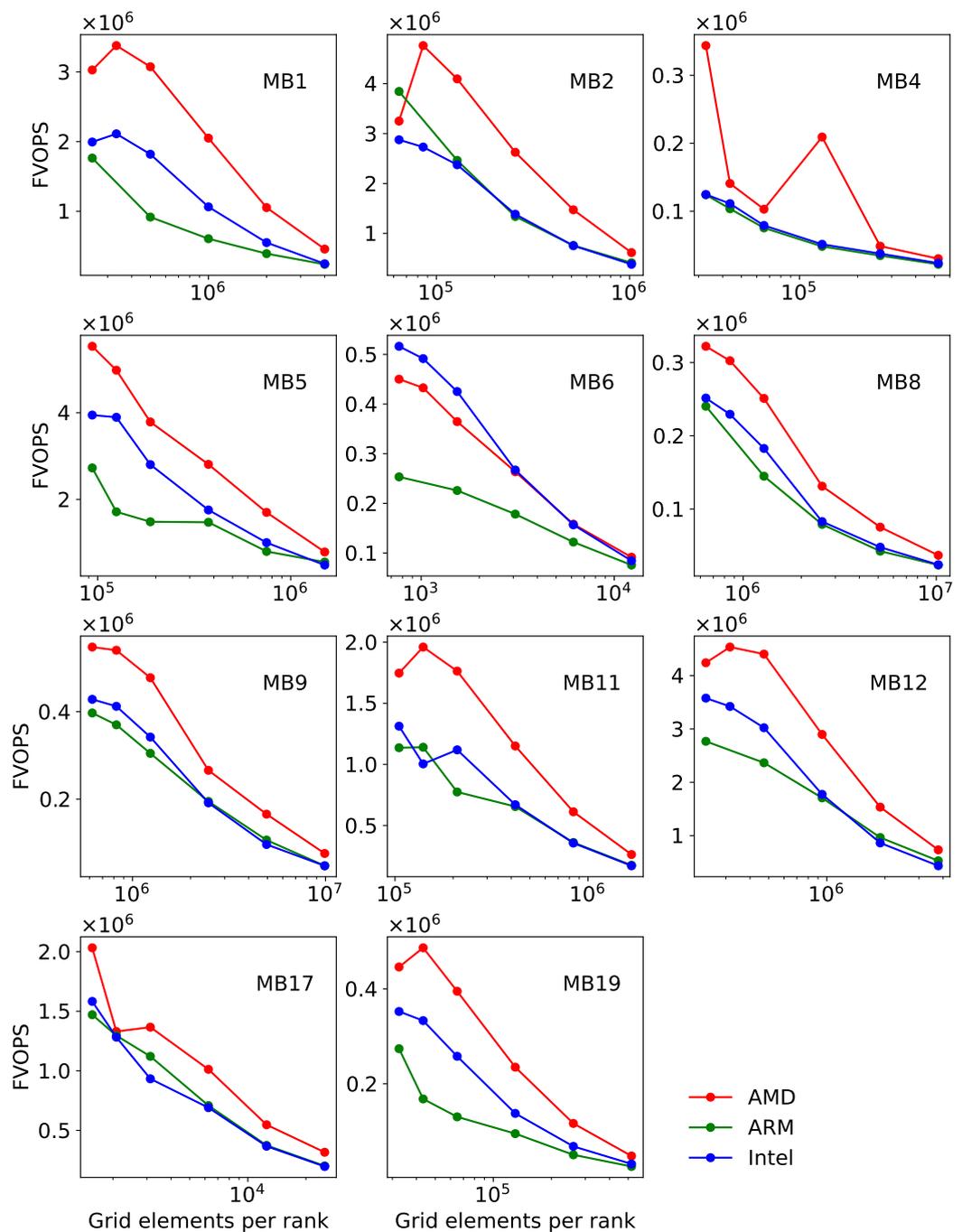


Figure 15. FVOPS metric vs. grid elements per rank for different microbenchmarks. The black line represents the ideal efficiency.

4. Discussion

Calculating a mean FVOPS value across all MBs, using the FVOPS value with the optimal number of grid points per core for each test case and architecture, allows a straightforward comparison of the different architectures. The mean FVOPS for AMD is 2.21×10^6 , for Arm is 1.36×10^6 , and for Intel is 1.55×10^6 . The mean performance with AMD CPUs is 62% higher than with Arm and 42% higher than with Intel.

To illustrate the performance of the different MBs using the different architectures, Figure 16 shows the same results as Figure 15 using box plots summing up the results for all grid sizes for each MB and architecture. The difference in performance using different test cases becomes evident. Figures 15 and 16 also corroborate that the studied MBs

seem particularly suited for AMD processors, with which shorter time-to-solution values were recorded.

OpenFOAM is a memory-bound application, but the difference in performance cannot be explained by the memory bandwidth alone, as the three architectures have similar memory systems of type DDR4, with the AMD and Arm having eight channels and the Intel having six channels. It is argued that the better utilization especially of the L3 cache is the main reason for the performance difference, as the L3 cache varies greatly between the architectures: the AMD EPYC node has 256 MB in comparison to the Arm node with 32 MB and the Intel node with 44 MB. The analysis demanded careful profiling, described in detail in the preprint [30]. It is found that the performance variation is linked to the grid size that each core has to solve. With a large grid size, the main memory access limits the performance, and the performance of all architectures is similar (see, e.g., Figure 15). When using more cores to solve the same problem, the grid size per core decreases to the point that most of the problem fits in the L3 cache, which translates into less access to the main memory and, consequently, more useful computation by the compute cores. When using even more cores, the time per time step can become very short, and the MPI latency decreases the performance. The conclusion is that the larger L3 cache of the nodes with AMD EPYC CPUs compared to Arm and Intel is responsible for better performance.

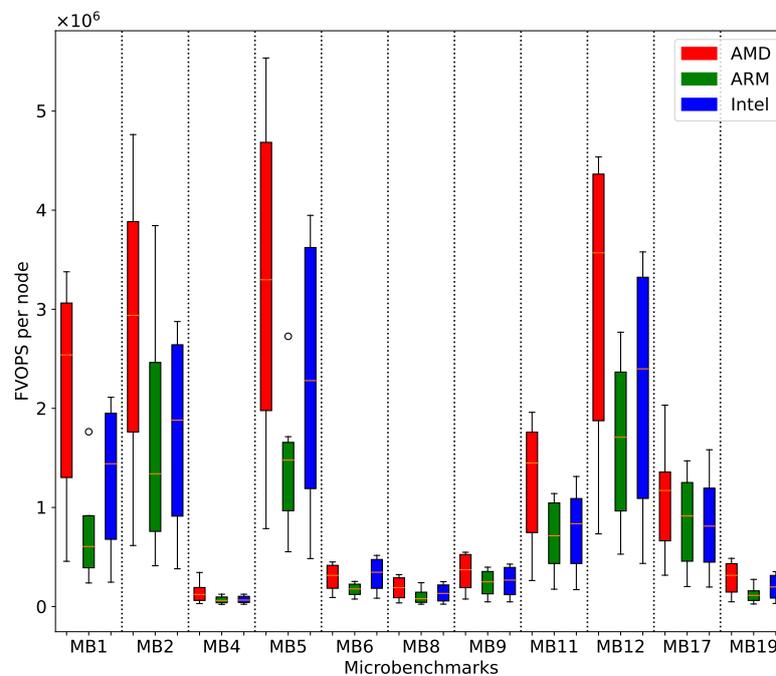


Figure 16. Box plots of the FVOPS metric with all grid sizes for different microbenchmarks.

5. Conclusions

A series of microbenchmarks (MBs) developed by the exaFOAM project has been introduced, having a broad range of applications: incompressible and compressible flow, combustion, viscoelastic flow, and adjoint optimization. The MBs have a relatively small memory footprint and short runtime, making them suitable to fill an HPC compute node and work as performance proxies of larger simulations.

Processors from Intel and AMD (x86_64 architecture) and Arm (aarch64 architecture) have been used. The Intel and Arm systems provided similar performance, despite the different architectures. However, the AMD processor seems particularly suited for the workloads in this study, with an overall shorter time-to-solution. The mean performance with AMD CPUs is 62% higher than with Arm and 42% higher than with Intel. The performance difference can be explained by the larger L3 cache on the AMD platform, as

the AMD EPYC node has 256 MB of L3 cache, while the Arm node has 32 MB and the Intel node has 44 MB.

The microbenchmarks are published in the OpenFOAM HPC Committee repository [8] to serve as benchmarks of different physical problems in various hardware architectures.

Author Contributions: Conceptualization, F.C.C.G. and M.G.-G.; funding acquisition, M.G.-G., D.G. and A.R.; investigation, E.B., J.P., S.L., H.R., S.B., M.C., F.B. and F.M.; methodology, F.C.C.G.; resources, A.R.; writing—original draft, F.C.C.G. and E.B.; writing—review & editing, M.G.-G. and R.G.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work is carried out in the scope of the exaFOAM project, which has received funding from the German Federal Ministry of Education and Research and the European High-Performance Computing Joint Undertaking (JU) under grant agreements No. 16HPC022K and No. 956416, respectively. The JU receives support from the European Union’s Horizon 2020 research and innovation programme and France, Germany, Spain, Italy, Croatia, Greece, and Portugal.

Data Availability Statement: All microbenchmarks are published in the OpenFOAM HPC Committee repository [8].

Acknowledgments: The following contributions are acknowledged: Federico Ghioldi and Federico Piscaglia for MB4; Andreas-Stefanos Margetis, Evangelos Papoutsis-Kiachagias and Kyriakos Gianakoglou for MB6; Julius Bergmann, Hendrik Hetmann and Felix Kramer for MB8 and MB9; Ricardo Costa, Bruno Martins, Gabriel Marcos Magalhães and João Miguel Nóbrega for MB19.

Conflicts of Interest: Authors Elisabetta Boella, Filippo Marchetti and Daniele Gregori were employed by the company E4 Computer Engineering, and authors Henrik Rusche and Sergey Lesnik were employed by the company WIKKI GmbH. The remaining authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

ACMI	Arbitrarily Coupled Mesh Interface
ADM	Actuator Disk Model
BM	blockMesh
CAD	Computer Aided Design
CFD	Computational Fluid Dynamics
CFM	cfMesh
DLR	Deutsches Zentrum für Luft- und Raumfahrt
ERCOFTAC	European Research Community on Flow, Turbulence and Combustion
HPC	High-performance computing
MBs	Microbenchmarks
SHM	snappyHexMesh

References

1. Weller, H.G.; Tabor, G.; Jasak, H.; Fureby, C. A tensorial approach to computational continuum mechanics using object orientated techniques. *Comput. Phys.* **1998**, *12*, 620–631. [CrossRef]
2. Jasak, H. OpenFOAM: Open source CFD in research and industry. *Int. J. Nav. Archit. Ocean. Eng.* **2009**, *1*, 89–94.
3. Segui, M.; Abel, F.R.; Botez, R.M.; Botez, R.M.; Ceruti, A. High-fidelity aerodynamic modeling of an aircraft using OpenFoam: Application on the CRJ700. *Aeronaut. J.* **2021**, *126*, 585–606 [CrossRef]
4. Pedro, R.; Gamez-Montero, J.; Raush, G.; Codina, E. Method for Fluid Flow Simulation of a Gerotor Pump Using OpenFOAM. *J. Fluids Eng.* **2017**, *139*, 111101.
5. Serdeczny, M.P.; Comminal, R.; Mollah, M.T.; Pedersen, D.B.; Spangenberg, J. Numerical modeling of the polymer flow through the hot-end in filament-based material extrusion additive manufacturing. *Addit. Manuf.* **2020**, *36*, 101454. [CrossRef]
6. Nieves-Remacha, M.J.; Kulkarni, A.A.; Jensen, K.F. OpenFOAM Computational Fluid Dynamic Simulations of Single-Phase Flows in an Advanced-Flow Reactor. *Ind. Eng. Chem. Res.* **2015**, *54*, 7543–7553. [CrossRef]
7. exaFOAM Project Website. Available online: <https://exafoam.eu/> (accessed on 28 February 2024).
8. OpenFOAM HPC Committee Code Repository. Available online: <https://develop.openfoam.com/committees/hpc/-/tree/develop/> (accessed on 28 February 2024).

9. AbdelMigid, T.A.; Saqr, K.M.; Kotb, M.A.; Aboelfarag, A.A. Revisiting the lid-driven cavity flow problem: Review and new steady state benchmarking results using GPU accelerated code. *Alex. Eng. J.* **2017**, *56*, 123–135. [[CrossRef](#)]
10. Ghia, U.; Ghia, K.N.; Shin, C.T. High-Re solutions for incompressible flow using the Navier–Stokes equations and a multigrid method. *J. Comput. Phys.* **1982**, *48*, 387–411. [[CrossRef](#)]
11. Koseff, J.R.; Street, R.L. Erratum: The Lid-Driven Cavity Flow: A Synthesis of Qualitative and Quantitative Observations. *J. Fluids Eng.* **1984**, *106*, 390–398. [[CrossRef](#)]
12. Bna, S.; Spisso, I.; Olesen, M.; Rossi, G. PETSc4FOAM: A Library to Plug-in PETSc into the OpenFOAM Framework. Zenodo. Available online: <https://zenodo.org/records/3923780> (accessed on 28 February 2024).
13. Brogi, F.; Bnà, S.; Boga, G.; Amati, G.; Esposti Ongaro, T.; Cerminara, M. On floating point precision in computational fluid dynamics using OpenFOAM. *Future Gener. Comput. Syst.* **2024**, *152*, 1–16. [[CrossRef](#)]
14. Fiolitakis, A.; Arndt, C.M. Transported PDF simulation of auto-ignition of a turbulent methane jet in a hot, vitiated coflow. *Combust. Theory Model.* **2020**, *24*, 326–361. [[CrossRef](#)]
15. ERCOFTAC. Swirling Diffuser Flow. Available online: https://www.kbwiki.ercofac.org/w/index.php?title=UFR_4-06_Description (accessed on 28 February 2024).
16. Classic Collection Database. Available online: [http://cfd.mace.manchester.ac.uk/ercofac/doku.php?id=cases:case060&s\[\]=conical](http://cfd.mace.manchester.ac.uk/ercofac/doku.php?id=cases:case060&s[]=conical) (accessed on 28 February 2024).
17. Dehkordi, B.; Moghaddam, H.; Jafari, H. Numerical Simulation of Flow Over Two Circular Cylinders in Tandem Arrangement. *J. Hydrodyn. Ser. B* **2011**, *23*, 114–126. [[CrossRef](#)]
18. Mittal, S.; Kumar, V.; Raghuvanshi, A. Unsteady incompressible flows past two cylinders in tandem and staggered arrangements. *Int. J. Numer. Methods Fluids* **1997**, *25*, 1315–1344. [[CrossRef](#)]
19. Griewank, A.; Walther, A. Algorithm 799: Revolve: An implementation of checkpointing for the reverse or adjoint mode of computational differentiation. *ACM Trans. Math. Softw.* **2000**, *26*, 19–45. [[CrossRef](#)]
20. Hupertz, B.; Chalupa, K.; Krueger, L.; Howard, K.; Glueck, H.-D.; Lewington, N.; Chang, J.-I.; Shin, Y.-S. On the Aerodynamics of the Notchback Open Cooling DrivAer: A Detailed Investigation of Wind Tunnel Data for Improved Correlation and Reference. *SAE Int. J. Adv. Curr. Prac. Mobil.* **2021**, *3*, 1726–1747. [[CrossRef](#)]
21. 4th Automotive CFD Prediction Workshop. Available online: <https://autocfd.eng.ox.ac.uk/#test-cases> (accessed on 28 February 2024).
22. Lacy, D.; Clark, A. Definition of Initial Landing and Takeoff Reference Configurations for the High Lift Common Research Model (CRM-HL). In Proceedings of the AIAA Aviation 2020 Forum, Virtual, 15–19 June 2020; AIAA Paper 2020–2771. [[CrossRef](#)]
23. Shur, M.; Spalart, P.; Strelets, M.; Travin, A. A hybrid RANS-LES approach with delayed-DES and wall-modelled LES capabilities. *Int. J. Heat Fluid Flow* **2008**, *29*, 1638–1649. [[CrossRef](#)]
24. Chin, V.; Peters, D.; Spaid, F.; McGhee, R. Flowfield measurements about a multi-element airfoil at high Reynolds numbers. In Proceedings of the 23rd Fluid Dynamics, Plasmadynamics, and Lasers Conference, Orlando, FL, USA, 6–9 July 1993; AIAA Paper 1993–3137. [[CrossRef](#)]
25. Pitz, R.W.; Daily, J.W. Combustion in a turbulent mixing layer formed at a rearward-facing step. *AIAA J.* **1983**, *21*, 1565–1570. [[CrossRef](#)]
26. Chamorro, L.P.; Porté-Agel, F. Effects of thermal stability and incoming boundary-layer flow characteristics on wind turbine wakes: A wind-tunnel study. *Bound. Layer Meteorol.* **2010**, *136*, 515. [[CrossRef](#)]
27. Wu, Y.T.; Porté-Agel, F. Large-Eddy Simulation of Wind-Turbine Wakes: Evaluation of Turbine Parametrisations. *Bound. Layer Meteorol.* **2011**, *138*, 345–366. [[CrossRef](#)]
28. Chamorro, L.P.; Porté-Agel, F. Turbulent flow inside and above a wind farm: A wind-tunnel study. *Energies* **2011**, *4*, 1916. [[CrossRef](#)]
29. Wu, Y.-T.; Porté-Agel, F. Simulation of Turbulent Flow Inside and Above Wind Farms: Model Validation and Layout Effects. *Bound. Layer Meteorol.* **2013**, *146*, 181–205. [[CrossRef](#)]
30. Galeazzo, F.C.C.; Weiß, R.G.; Lesnik, S.; Rusche, H.; Ruopp, A. Understanding Superlinear Speedup in Current HPC Architectures. *Preprints* **2024**, 2024040219. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.