EPJ Web of Conferences **302**, 05005 (2024) https://doi.org/10.1051/epjconf/202430205005
SNA + MC 2024

salome_cfd: An HPC framework for thermalhydraulics and multiphysics powered by the open-source Computational Fluid Dynamics Toolbox code_saturne

Chaï KOREN[1,*]

[1]EDF R&D, Fluid Mechanics, Energy and Environment Dept., 6 quai Watier, Chatou, 78400, France

Abstract. EDF R&D lab Chatou, Fluid Mechanics Energy and Environment department (MFEE), has been developping an HPC oriented Computational Fluid Dynamics toolbox since the late 1990s. This toolbox is based on the open-source Computational Fluid Dynamics (CFD) and multiphysics solver code_saturne. It has a proprietary extension, neptune_cfd, which is a Computational Multiphase Fluid Dynamics solver dealing with multi-phase flows. After a brief overview of code_saturne and neptune_cfd architecture and HPC capabilites, the present paper will focus on the effort made at EDF R&D to harness HPC clusters for thermalhydraulics at an industrial scale. This is made available by codes built from the start with massively-parallel architectures in mind, thus leading to highly scalable solvers which are able to handle tens of billions of cells. User-friendly access and interoperability are pivotal for engineers to take advantage of these HPC capabilities. Among common applications requiring HPC resources, a focus is first made on frontier scale simulations, then on parametric studies and uncertainty quantifications experiments, where a two-level parallelism is needed : a parallel Design of Experiment evaluation drives computations themselves (massively) parallel.

1 Introduction

EDF possesses one of the world's largest fleets of nuclear power-plants, but also hydropower plants, which has led the company to develop a strong expertise in fluid mechanics since the creation of EDF R&D division in 1946 on the island of Chatou.
Moreover, open-science and transparency are a key feature of the nuclear industry, which has always been a driving mechanism for EDF R&D teams developing simulation software such as code_saturne for computational fluid mechanics, code_aster for computational structural mechanics, Tlemac-Mascaret for free-surface flows or the platform for scientific computing Salome, to name the most well-known.
This paper will be centered around the open-source computational fluid dynamics toolbox code_saturne, which is developed at EDF R&D since 1997 and distributed under an open-source license since 2007 [1]. code_saturne is a key pillar of EDF's strategy in terms of numerical simulation tools, as it is the reference Computational Fluid Dynamics solver for single-phase flows and multiphysics for nuclear reactor thermal-hydraulics safety studies at the local scale. In addition to the core solver of the Navier-Stokes equations, code_saturne

*e-mail: chai.koren@edf.fr

also possesses a number of modules for the resolution of specific physics: thermal radiation, free surface flows, turbomachinery, fire, electric arcs, atmospheric flows, ...
code_saturne's open-source status allowed the creation of an ecosystem around it, which benefits both EDF and the hundreds of users worldwide. An example of such a success is the use of code_saturne as one of PRACE's restricted Unified European Benchmark Application Suite of software [2], as well as the participation in most grand challenges in France's national supercomputing centers supervised by GENCI (CCRT/TGCC, IDRIS, CINES) [3]. Furthermore, neptune_cfd ¸ the multiphase flow solver developed within the NEPTUNE four-party project (CEA, EDF, Framatome, IRSN) is an extension of code_saturne, sharing with the latter many of its building blocks and features, thus benefiting directly from all of the contributions to code_saturne. This is a great example of a way of leveraging an open-source code by an industrial actor.
Moreover, code_saturne can also be integrated into the Salome platform, co-developed by EDF and CEA, with an enriched Graphical User Interface (GUI), to provide a fully open-source workbench for Computational Fluid Dynamics engineers. This workbench contains many useful modules: GEOM/Shaper for CAD modeling, SMESH for meshing purposes and which integrates several algorithms, Paraview with specific extensions as well as Persalys which is the GUI for OpenTURNS and ADAO, respectively developed for uncertainty quantification studies and data-assimilation. To our knowledge, no such other workbench is available in the open-source community with equivalent capacities.
A use case application of the workbench will be presented, based on a real-life problem treated at EDF R&D: Uncertainty-Quantification applied to an industrial thermalhydraulic application where the salome_cfd platform is used, and calling upon different building blocks of salome and HPC functionalities of code_saturne to combine a parametric geometry, automatic mesh generation, simulation on HPC clusters all the while the DoE evaluation program is running on a user's laptop.

2 The salome_cfd platform

The core component of the salome_cfd platform is code_saturne [1] which is a multiphysics Computational Fluid Dynamics toolbox developped at EDF R&D since 1997, whose first industrial version, code_saturne v1.0, was released in 2001 before the code was made available under an open-source license (GPL) in 2007.

2.1 Physical modeling capacities of salome_cfd

The salome_cfd platform builds on the generic Salome platform by adding code_saturne, neptune_cfd and Syrthes (modeling of heat transfer in solid media) with the goal of providing users with a tailored workbench for, but not only, Nuclear Power Plant (NPP) related applications. Using code_saturne and neptune_cfd multiphysics capacities, users have access to single and multi-phase flows modeling options, and can also handle solid particles using Lagrangian or Eulerian approaches, free surface flows (Volume of Fluid (VoF), Arbitrary Lagrangian-Eulerian (ALE), Eulerian-Eulerian approaches), reactive flows, atmospheric flows, boiling flows, droplet-laden flows, conjugate heat transfer as well as fluid-structure interaction (monolithic internal solver or code coupling). An illustration of several NPP oriented industrial applications using either code_saturne or neptune_cfd is provided in Fig. 1.
In addition to these physical modules, nuclear industry specific modules are also available: a cooling towers module, groundwater flows (for long term storage), solidification (for ingot casting) and fusion (severe accidents mitigation). Except for the former, all of these new modules are built using the new CDO (Compatible Differential Operators) kernel of

code_saturne [4–8]. The integration within the Salome platform also provides a predefined workflow combining code_saturne and neptune_cfd with **OpenTURNS** and its GUI, **Persalys**, paving the way for parametric studies. An illustration of such an application is provided in Sec. 2.4.

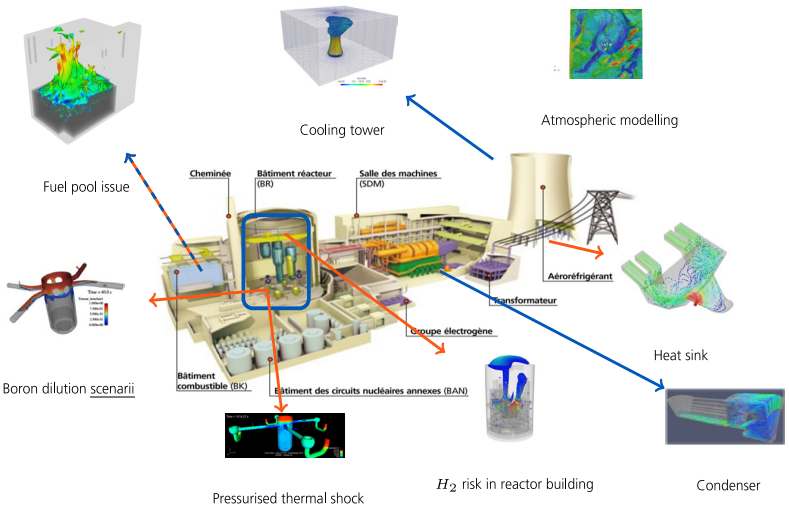

Figure 1. Nuclear Power Plant oriented industrial applications of the salome_cfd platform

2.2 code_saturne HPC framework

The code is based upon a co-located finite volume approach that can handle three-dimensional unstructured meshes containing any kind of cell type, including general polyhedra. The main parallelism paradigm of the code is based on the distributed-memory model and using domain-partionning, where communications are done using MPI with a Bulk Synchronous Parallel (BSP) model. An illustration of the Halo system (ghost cells) used by code_saturne is provided in Fig. 2.

Moreover, to partition the initial mesh code_saturne I/O functionality allows for both Graph-based approaches using external libraries such as **ParMETIS** or **PT-Scotch** and Space-filling curve based (Morton, Hilbert) using internal algorithms. The former approach generally has the advantages of providing with a more optimized partitioning for MPI based parallelism, see Fig. 3 for such an example, while the latter can handle larger meshes and has shown its capability of dealing with meshes of up-to tens of billions of cells. This approach has long been extended in the code_saturne framework to allow for hybrid parallelism mixing MPI with OpenMP instructions, thus reducing both the memory usage due to MPI and the cost of an MPI communication. An illustration of the benefit of such a hybrid approach will be shown in the last example in this section (Fig. 6).

This approach has proved its value and shown its capacity to deal with billion cells, or more, meshes for industrial configurations [9, 10]. Weak scaling results obtained on the HECToR and ARCHER supercomputers are provided in Fig. 4 and show satisfying results going up to 72 000 cores for the legacy kernel, and 98 300 cores for the CDO kernel on a 1.7B cells mesh. A more recent frontier computation was conducted on the TGCC Joliot-Curie Supercomputer, over the IRENE-AMD partition during the *Grand challenges* period of the

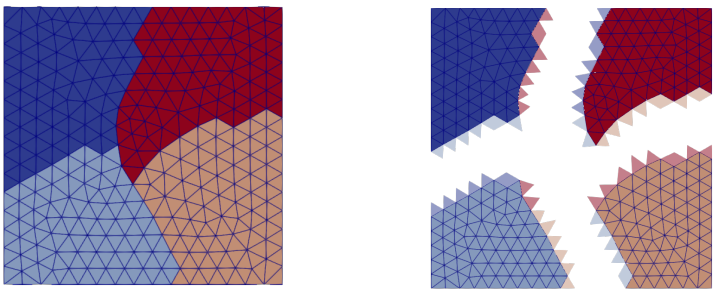

Figure 2. Ghost cell based parallelism. Left: triangular mesh with cells colored by rank id. Right: Shrinked version of left side mesh, with ghost cells (without edges) colored by owner rank id.

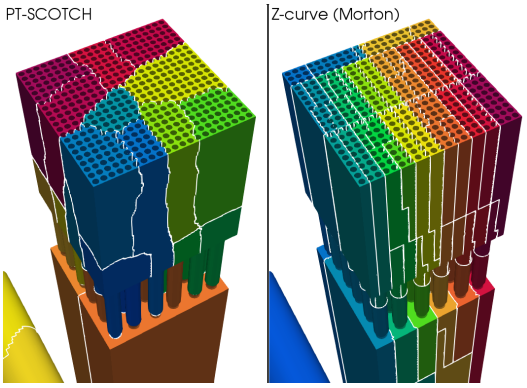

Figure 3. Example of graph-based partitioning vs. space-filling curve partitioning

supercomputer. Computations were run for a non nuclear application, a reactive fluidized bed reactor, using the neptune_cfd solver and a Particles laden gas flow using Eulerian-Eulerian modeling [3]. It was possible to run the computation over a 64 billion cells mesh over 144 000 MPI tasks. For this work, the size of the MPI buffers exceeded 1GB of ram per task, hence limiting the number of cores usable using a pure MPI approach and exhibiting the interest of leveraging a hybrid MPI/OpenMP parallelism as explained beforehand. A finer run, with 512 billion cells was unable to complete due to limitations when writing the initial state data files for entire computational domain. Preparing for the next generation of supercomputers is also an active field of work and research in the code_saturne and neptune_cfd development team at EDF R&D. The HPC community is in the middle of a transition from a fully x86/CPU oriented field to a GPU dominated scene (see the November 2023 TOP500 list [13]). While x86 CPUs are still heavily represented, more and more ARM based, as well as RISC-V based ([14]), processors are being deployed, mainly thanks to their energy efficiency when compared to x86 units.

For all of these reasons, porting and optimizations were done or are ongoing for these different platforms. Examples of such work are available in [15, 16].

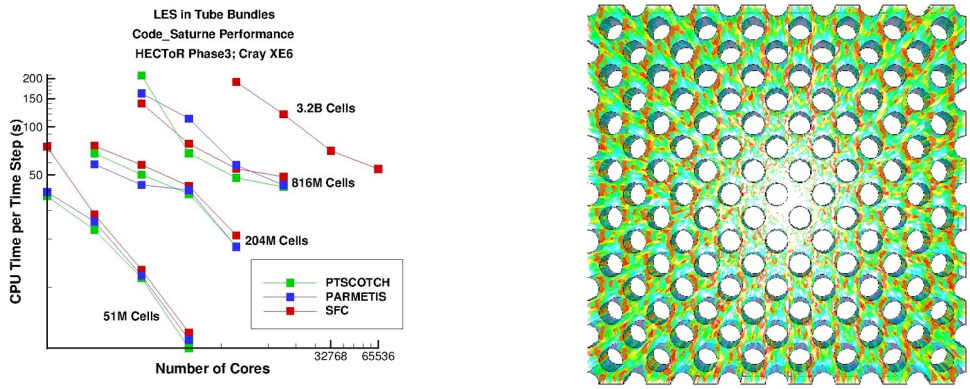

Figure 4. Weak scaling curves obtained with code_saturne on the HECToR and ARCHER supercomputing facilities on the tube bundle experiment of [11].

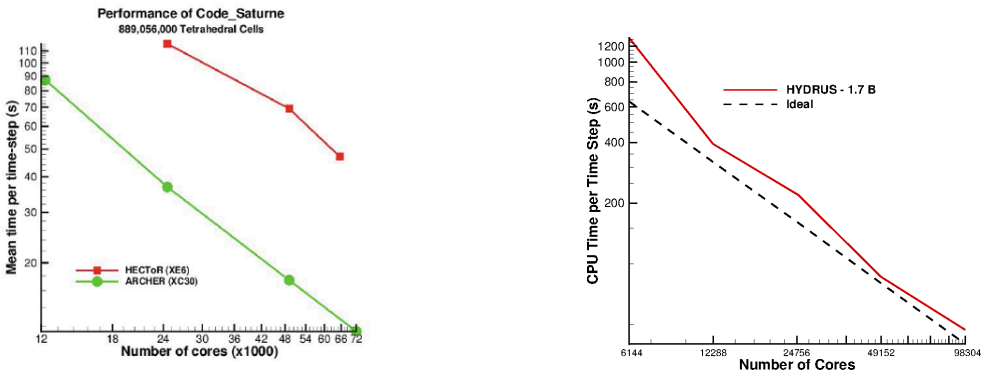

Figure 5. Weak scaling curves obtained with code_saturne on tetrahedral meshes. Left: code_saturne legacy kernel on experiment from [11], Right: code_saturne CDO kernel applied to the HYDRUS3D test case (Groundwater flows).

2.3 Making HPC accessible to engineers

The code_saturne framework aims at providing both research scientists and engineers with a Computational Fluid Dynamics code capable of dealing with any kind of flow or fluid related application in the nuclear industry. In addition to developing and maintaining a highly scalable and efficient CFD solver, as part of the R&D division of an industrial company, the code_saturne and neptune_cfd development team also needs to ensure that thermalhydraulics or reactor physics engineers are capable of using the code and harnessing the power of HPC infrastructures in order to address their day-to-day work.

A first step is providing an easy way of interacting with the code and the HPC infrastructure. For code_saturne and neptune_cfd, it is done thanks to the built-in Graphical User Interface (GUI) provided for both solvers. The GUI allows to handle a dataset of a computation, but also the submission of a new computation on a supercomputer by using a workload manager (SLURM, PBS/Torque, CCC, LSF, OAR and SGE are supported) without the user having to manually edit batch scripts. All one has to do is provide the needed resources (number of

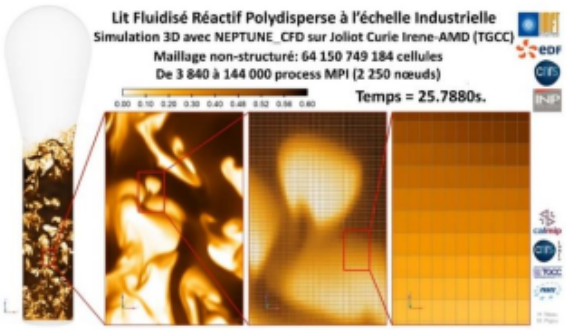

Figure 6. Fluidized bed reactive simulation. Description of the physical modeling is available in [12]

nodes, tasks per node) and wished wall-clock time: the code will handle the rest (see Fig. 7 for an example of the submission widget available in the code_saturne GUI). Moreover, submission parameters are stored in the case data files with the host machine and installation metadata, hence giving the user the possibility of storing these parameters for different hosts, for example a workstation on which the case was set and a cluster on which the real size run is done. To avoid loss of data due to insufficient allocated wall-clock time, the solver automatically checks for the specified wall-clock time and stops the computation all the while performing the checkpoint saving actions allowing for a restart if there is not enough time to finish the run. If this occurs, the code creates a stamp file ("run_status.exceeded_time_limit") to inform the user that it was unable to fully run the computation and that a resubmission is necessary for it to finish. A functionality introduced in 2023 (code_saturne v8.0) allows for an automatic resubmission in case of an attained time limit.

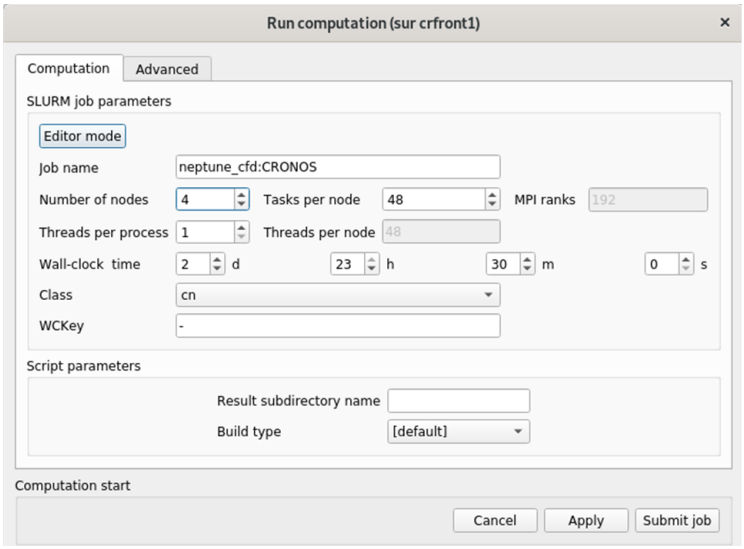

Figure 7. View of the submission widget available in the code_saturne and neptune_cfd GUI

Another important aspect for HPC usage is the interoperability capabilities of the code.

Sooner or later, a code will be coupled with other software packages or simulation tools. User-friendly access to this coupling should be provided while not harming the HPC performance by inducing an important overhead due the coupling technology used. code_saturne and neptune_cfd are provided with an in-house coupling library PLE [17], but also contain built-in functionality using the MEDCoupling library [18]. External works, outside of EDF R&D, have also successfully coupled code_saturne using the MUI [19] and CWIPI [20] libraries. In addition to these HPC oriented coupling libraries, an FMU export using the FMI standard [21] is also possible. For predefined couplings such as Conjugate Heat Transfer (with Syrthes), System-scale thermalhydraulic / CFD (with CATHARE [22]) or a generic python script, coupling parameters can be entirely handled using the code_saturne GUI that provides a dedicated widget for these applications, thus rendering code coupling application more easily usable.

2.4 Parametric studies

A final HPC oriented use case is that of parametric studies. In today's industry, the capability of exploring a DoE (Design of Experiment) using a simulation tool is necessary for Reduced Order Modeling approaches, Uncertainty Quantification (UQ) or Optimization process. To achieve such a goal, code_saturne and neptune_cfd run command possesses a *parametric* option, which allows modifying an element of predefined model on runtime. For example, to modify the mesh to use, a user can simply use the following command line command:

```
code_saturne run --parametric-args mesh=new_mesh_file.mesh_format
```

This functionality allows one to modify the physical time of simulation, the type of boundary conditions, the notebook parameters or to set a previous computation to restart from. The code_saturne GUI provides users with a notebook page, where it is possible to define parameters which are then visible both in the GUI and in User defined advanced functions, and usable as initial conditions, boundary conditions, source terms, reference values for physical properties equations, etc.
Thanks to these options, user can easily handle simple DoE or couple code_saturne and neptune_cfd with dedicated libraries using python scripts. In salome_cfd, a predefined workflow with OpenTURNS and its GUI (Persalys) is available. Simply put, in code_saturne GUI within salome_cfd, an export button is available which exports a data model which was parametrized using the notebook of the GUI to Persalys. The exported model is then treated as function **F** such as **Y = F(X)** where **X** is an input and **Y** is the output. The user can then define the type of DoE to use (Optimization, UQ, Reduced Order Model generation, ...) and then proceed to run the DoE.
An example of a shape optimization process is provided in Fig. 8 where everything is done using salome_cfd. The user defines a geometry in the CAD module (step a), then defines mesh generation options (step b) based on the CAD, and finally defines a CFD model based on the mesh from step b (step c). This "combined" function is then provided to Persalys (d). Thanks to salome_cfd a two-level parallelism is used: the evaluation of the DoE is done in a parallel manner where the workload is divided between different threads (first level) and each evaluation of a set of parameters is a parallel computation using code_saturne or neptune_cfd (second level). To render all of this completely transparent to the user, the definition of the DoE and its execution can be done on the user's workstation or laptop, while each CFD computation is run on a predefined distant host (HPC cluster, in Fig. 8, EDF cluster CRONOS is visible). To do so, several low-level building blocks of Salome are used. A built-in client first transfer all the large files which are needed for the CFD computations to the distant host including the reference CFD data model. Then, when the DoE is evaluated,

each thread is submitting a job on the distant host using Salome **JobManager** tool, and checks on the jobs status regularly. Once a computation is complete, the thread syncs the output files, which are lightweight ascii files, and provides Persalys with the results of the evaluation. For long running jobs, the DoE can be paused and restarted later if needed.

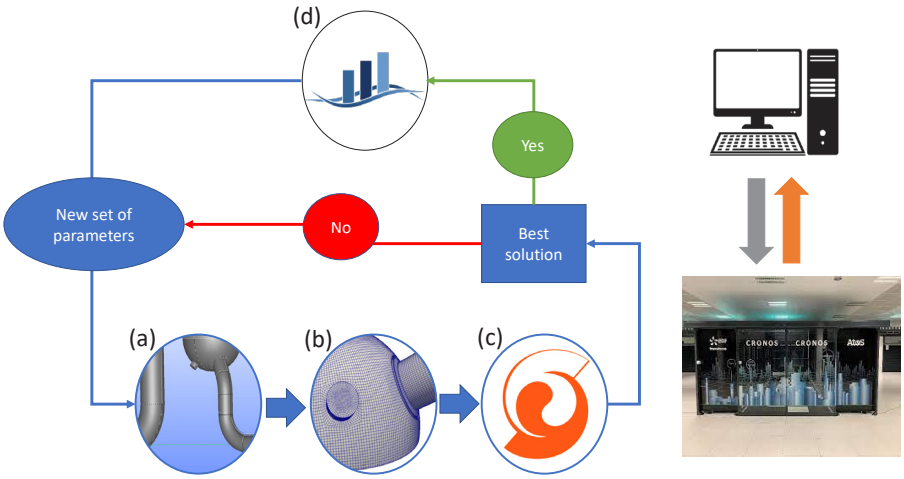

Figure 8. View of the submission widget available in the code_saturne and neptune_cfd GUI

An example of application of this workflow to a nuclear industry use case is available in [23]. In the work of Wald et al., the code_saturne & OpenTURNS workflow was applied to a situation where a fuel assembly of a pressurized water reactor is blocked inside the transfer tube between the reactor building and the spent fuel pool. The authors studied the influence of several parameters, both physical (pool temperatures, residual power, flux peak) or geometrical (locked position of the fuel assembly, sizes of the pools), as well as the sensitivity to mesh refinement (3 levels, going from 8M to 256M cells) for a total of several hundreds of computations. This kind of work, combining CFD, Uncertainty Quantification, as well as meta-modeling to propagate uncertainties while taking into account the influence of the mesh refinement or the turbulence model, could not have been done without a suitable workflow combining state of the art simulation and data science tools.
An additional feature available in salome_cfd for parametric studies is the combination of code_saturne with the Melissa library as showcased in [24]. This coupling adds the possibility to conduct in-situ ensemble operations over large meshes and/or number of simulations and thus drastically reducing the amount of data which needs to be stored to the hard-drive during a single calculation.

3 Conclusion

HPC infrastructure is the backbone of modern science and engineering applications by unleashing an unprecedented amount of computational power. For an industrial company such as EDF, harnessing this kind of computational resources is a challenge to handle larger computational models, higher fidelity models than those used in the past and an increasing number

of parametric studies applied to optimization, reduced order modeling, uncertainty quantification, etc. Beyond this scientific challenge, there is also an economical challenge of being able to contain the costs induced by numerical simulation due to both hardware and energy consumption, while not limiting the activities cited here-above.

This paper showcased the strategy which is applied for thermalhydraulics C(M)FD at EDF R&D for dealing with these challenges: first of all, highly optimized solvers are developed and deployed, Sec. 2.2, to ensure that a massively-parallel computation is not wasting allotted resources. Secondly, efforts are made to facilitate the usage of HPC clusters by users, Sec. 2.3, mainly with the help of dedicated GUI widgets and host specific configuration to reduce errors and facilitate transitions between workstations and supercomputers. Finally, an application combining the different tools is given in Sec. 2.4, which exhibits what can now be achieved when provided with a large amount of computational hours, and how HPC can be harnessed to deal with industrial scale applications thanks to the salome_cfd platform powered by code_saturne.

Near future milestones of code_saturne and neptune_cfd include further optimization of the hybrid parallelism to increase efficiency on ARM based systems, continue the effort of porting the code's kernels on GPU hardware all the while keeping a maintainable code and avoiding an increase of the technical debt. Another topic not presented in this paper, but which is also studied is the usage of energy consumption management tools (see [25] for an example) to modulate hardware energy usage during a computation to adapt a CPU frequency at specific steps (I/O for example), thus decreasing the total energy consumption of a single computation.

References

[1] Archambeau, F., Mechitoua, N., Sakiz, M., code_saturne: A finite volume code for the computation of turbulent incompressible flows-Industrial applications, International Journal on Finite Volumes **1** (2004)

[2] PRACE website: http://www.prace-project.eu

[3] TGCC Joliot-Curie supercomputer Grand Challenges report, page 26, https://www-hpc.cea.fr/docs/2021/Grands-Challenges-Joliot-Curie-2021.pdf, 10/03/2024

[4] Bonelle, J. (2014). Compatible Discrete Operator schemes on polyhedral meshes for elliptic and Stokes equations (Doctoral dissertation, Université Paris-Est).

[5] Bonelle, J., Fournier, Y., Moulinec, C. (2018). New polyhedral discretisation methods applied to the Richards equation: CDO schemes in code_saturne. Computers & Fluids, 173, 93-102.

[6] Bonelle, J., Ern, A., Milani, R. (2020). Compatible Discrete Operator schemes for the steady incompressible Stokes and Navier–Stokes equations. In Finite Volumes for Complex Applications IX-Methods, Theoretical Aspects, Examples: FVCA 9, Bergen, Norway, June 2020 IX (pp. 93-101). Springer International Publishing.

[7] Bonelle, J., Fonty, T. (2023, October). Compatible Discrete Operator schemes for solidification and segregation phenomena. In International Conference on Finite Volumes for Complex Applications (pp. 187-194). Cham: Springer Nature Switzerland.

[8] Jang, Y., Bonelle, J., Kruse, C., Hülsemann, F., Ruede, U. Fast Linear Solvers for Incompressible CFD Simulations with Compatible Discrete Operator Schemes. 2023. hal-04087358

[9] Vezolle, P., Heyman, J., D'Amora, B., Braudaway, G., Magerlein, K., Magerlein, J., Fournier, Y. (2010, October). Accelerating computational fluid dynamics on the IBM Blue

Gene/P supercomputer. In 2010 22nd International Symposium on Computer Architecture and High Performance Computing (pp. 159-166). IEEE.

[10] Fournier, Y., Bonelle, J., Le Coupanec, E., Ribes, A., Lorendeau, B., Moulinec, C. (2015). Recent and upcoming changes in code saturne: computational fluid dynamics HPC tools oriented features. In Fourth International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering, Dubrovnik, Croatia

[11] Simonin, O., Barcouda, M., 4th Int. Symp. on Apps of Laser Anemometry to Fluid Mechanics. (1988).

[12] Neau, H., Pigou, M., Fede, P., Ansart, R., Baudry, C., Mérigoux, N., Laviéville, J., Fournier, Y., Renon, N., Simonin, O. Massively parallel numerical simulation using up to 36,000 CPU cores of an industrial-scale polydispersed reactive pressurized fluidized bed with a mesh of one billion cells, Powder technology, 366, 906–924 (2020)

[13] HPC TOP500 website https://www.top500.org/lists/top500/list/2023/11/

[14] European processor initiative https://www.european-processor-initiative.eu/

[15] ANTAO, Samuel, MOULINEC, Charles, FOURNIER, Yvan, et al. GPU Acceleration at Scale with OpenPower platforms in Code Saturne. SuperComputing 2018

[16] Teratec 2022 Graviton HPC Hackathon https://teratec.eu/library/pdf/evenements/2022/HPC-Hackathon-TERATEC-Information-planning-en.pdf

[17] Fournier, Y. (2020). Massively parallel location and exchange tools for unstructured meshes. International Journal of Computational Fluid Dynamics, 34(7-8), 549-568.

[18] MEDCoupling on-line documentation https://docs.salome-platform.org/latest/dev/MEDCoupling/developer/index.html

[19] MUI library website https://mxui.github.io/

[20] CWIPI library website https://w3.onera.fr/cwipi/fr

[21] FMI website https://fmi-standard.org/

[22] CATHARE code website https://cathare.cea.fr/

[23] Wald, J.F., Iooss, B., Uncertainty quantification and metamodeling of multi-fidelity CFD computation of a heated fuel assembly, Submitted to Best Estimate Plus Uncertainty International Conference (BEPU 2024), Real Collegio, Lucca, Tuscany, Italy, May 2024.

[24] Terraz, T., Ribes, A., Fournier, Y., Iooss, B., Raffin, B., Melissa: Large Scale in Transit Sensitivity Analysis Avoiding Intermediate Files, Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, Denver, Colorado, United-States (2017)

[25] BSC's EAR website https://www.bsc.es/research-and-development/software-and-apps/software-list/ear-energy-management-framework-hpc