European Data Infrastructure
H2020-INFRAEDI-2018-2020

INFRAEDI-02-2018: Centres of Excellence on HPC

EoCoE-II

Energy Oriented Center of Excellence:
toward exascale for energy

Grant Agreement Number: 824158

D7.2
First report of the exascale co-design group

Project and Deliverable Information Sheet

EoCoE-II	Project Ref:	EINFRA-824158
	Project Title:	Energy Oriented Center of Excellence: toward exascale for energy
	Project Website:	http://www.eocoe.eu
	Deliverable ID:	D7.2
	Deliverable Nature	Report
	Dissemination Level:	PU *
	Contractual Date of delivery:	M18 30/06/2020
	Actual Date of delivery	M21 15/09/2020
	EC Project Officer	Evangelia Markidou

* The dissemination levels are indicated as follows: PU – Public, CO – Confidential, only for members of the consortium (including the Commission Services) CL – Classified, as referred to in Commission Decision 2991/844/EC.

Document Control Sheet

Document	Title:	First report of the exascale co-design group
	ID:	D7.2
	Available at:	http://www.eocoe.eu
	Software Tool:	Microsoft Word
Authorship	Written by:	Paul Gibbon (FZJ), Edouard Audit (CEA)
	Contributors:	Pasqua D'Ambra (CNR), Johanna Bruckmann (RWTH), Franceso Buonocore (ENEA), Fabio Durastante (CNR), Phillip Franke (FZJ), Mathieu Lobet (CEA), Yen-Sen Lu (FZJ), Sebastian Lührs (FZJ), Edoardo Di Napoli (FZJ), Alessandro Pecchia (CNR) Bruno Raffin (INRIA), Herbert Owen (BSC), Alison Walker (UBAH)
	Reviewed by:	Project Executive Committee (PEC), Scientific Challenge Leaders

Document Keywords:
Dissemination; communication

Index

1. Executive Summary

The ECG is the co-design core of the project and was set up to facilitate interaction between the technically oriented WPs and Scientific Challenges comprising WP1, initiating discussions and cooperation between WPs when mission-critical application design decisions are needed. In effect, the ECG is broadly composed of technical experts and application lead developers from all WPs, providing a direct communication channel between application developers, HPC experts, and mathematical algorithm specialists, and to encourage a strong co-design by all project partners. This body is deliberately kept outside the formal WP structure and is also mandated to monitor and recommend cross-WP actions which were not foreseen in the DoW, for example in response to new hardware innovations or external developments, or to recommend remedial action in response to show-stopping events such as a lead developer leaving the project. The ECG is animated and fostered by the ECG-leader in close consultation with the project coordinator and manager. The meetings of the ECG take place monthly, generally via videoconference, but relevant issues are often addressed at meetings of the management board or PEC.

The purpose of this document is to report on the ECG aspects of Task 7.1 of the original proposal (strategic and technical coordination), in particular:

- Progress on Flagship codes towards Exascale
- Identification of bottlenecks and possible mitigation measures
- Progress on numerical libraries and tools critical to meeting the Scientific Challenge goals

These three aspects were central to the original conceptual design of the EoCoE-II project illustrated in Figure 1. In the following Sections 2 and 3 of this document we will describe how these activities have been implemented in practice, reporting on progress in performance enhancements, but also remedial action which has been taken on several flagship codes to keep these tasks on course. For the most part, the contents of this deliverable draw on the material available in the M18 Deliverables, but also includes activity involving the flagship codes and libraries which occurred outside the original DoW scope.

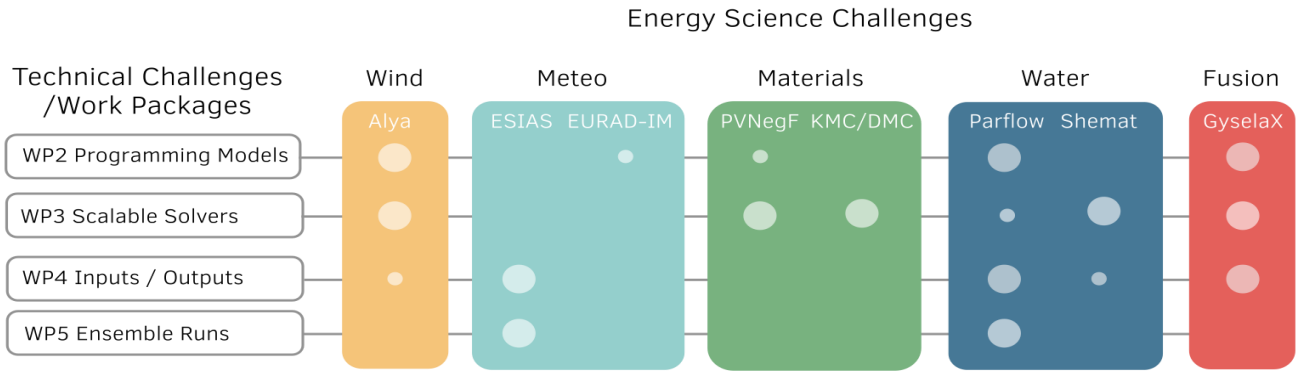

Figure 1: Original co-design concept of EoCoE-II, identifying interaction points between Technical and Energy Science (WP1) challenges where project resources are concentrated within work-packages WP2-WP5.

2. Scientific Challenges and Flagship Codes

As indicated above in the Summary, this deliverable aims to summarize the efforts which have been undertaken in the EoCoE-II project to bring a selected number of codes to Exascale. The reward for achieving this will be to enable predictive modelling with unprecedented accuracy and/or scope in the five constituent energy science domains Wind, Meteo, Materials, Water and Fusion of WP1. As the structural outline in Figure 1 makes clear, these 'exascaling' tasks are effectively distributed among the five main work packages WP1-WP5, and accordingly documented in deliverables D1.2, D2.2, D3.2, D4.1 and D5.1. To help the reader locate details particular to each flagship code, Table 1 below provides a progress summary and references to the relevant sections in each of these deliverables and the original task in the DoW. Not included here are the transversal activities in WP2-WP5 necessary to develop exascale tools and libraries – these are described later in Section 3.

Code	Lead Partner	Current implementation	Hardware	Current scalability (#cores)	Exascale-relevant activity	Deliverable Reference	Task in DoW
Alya	BSC	MPI for multi-node	CPU and GPU	100 000, 24-hour production run on marenostrum	Demonstrator	D1.2, Section 3.3	1.1
					Performance opt	D2.2, Section 2.2	2.2
					LA solver	D3.2, Section 6	3.4
					FTI checkpointing	D4.1, Section 6.3	4.3
ESIAS	FZJ	MPI for multi-node	CPU	262 144 (4096 ensemble member on JUQUEEN)	Demonstrator	D1.2, Section 4.4	1.2
					I/O refactoring	D4.1, Section 5.2	4.2
					Melissa DA	D5.1, Section 6	5.2
EURAD-IM	FZJ	MPI for multi-node	CPU and GPU	8192 (ESIAS-chem JUQUEEN), 960 (JURECA)	Demonstrator	D1.2, Section 4.4	1.2
					Performance opt	D2.2, Section 6.2	2.3
					I/O refactoring	D4.1, Section 5.2	4.2
					Melissa DA	D5.1, Section 6	5.2
LibNEGF	CNR	MPI	CPU and GPU	36 000 cores on JUWELS (4 MPI x 12 threads per node)	Demonstrator	D1.2, Section 5.4	1.3
					Performance opt	D2.2, Section 7	2.4
					LA solver	D3.2, Section 3	3.1
KMC/DMC	UBAH	MPI	CPU	8192 cores on Isambard Marvell ThunderX2 (ARM)	Performance opt	D1.2, Section 5.3	1.3
					FMM solver	D1.2, Section 5.3	1.3
ParFlow	FZJ	C/ MPI	CPU and GPU	35 x (4CPU + 4GPU nodes) on JUWELS	Demonstrator	D1.2, Section 6.5	1.4
					Performance opt	D2.2, Section 8	2.5
					LA solver	D3.2, Section 4	3.2
					I/O refactoring	D4.1, Sections 4.2 & 4.3	4.2
					Melissa DA	D5.1, Section 5	5.3
SHEMAT-Suite	RWTH	Fortran / OpenMP	CPU and GPU		Demonstrator	D1.2, Section 6.5	1.4
					LA solver	D2.2, Section 8	3.2
					I/O refactoring	D4.1, Sections 4.2 & 4.3	2.5, 4.2
Gysela	CEA	Fortran / MPI / OpenMP	CPU (x86 and ARM)	98 304 with 59% efficiency; 49 152 (80%) on IRENE-AMD at TGCC	Demonstrator	D1.2, Section 7.4	1.5
					Performance opt	D2.2, Section 9	2.6
					LA solver	D3.2, Section 5	3.3
					I/O refactoring	D4.1, Section 4.2	4.2

Table 1: Summary of flagship codes and location of optimisation tasks within M18 deliverables and original Description of Work

It is worth noting that the overall PM effort in WP2-WP5 amounts to nearly 60% of the project total. Another 25% is dedicated to the energy science payload in WP1, roughly 1/3 of which goes towards 'demonstrator' development (eg verification and benchmarking against other satellite codes). Around 2/3 of the project PM are therefore committed to EoCoE-II exascale software development. So far it has not proved necessary to significantly reallocate these resources, apart from a few cases in which personnel changes forced some slight adjustments. The most prominent example here is the replacement of the PVnegf code by libNEGF due to the PVnegf lead developer leaving FZJ. In this case, some resources were rechannelled from FZJ to CNR, where the libNEGF lead is hosted and the corresponding reprioritisation made in the affected WPs.

Code Demonstrator repository

In the deliverable D1.2, the description within the code demonstrator section refers to a snapshot of the current status of the flagship codes (and satellite codes, whenever possible) at M18. This description includes an example of scientific simulation and relative results. For each code, we have provided the current software within a protected Gitlab repository accessible to 3rd parties by request.
https://gitlab.maisondelasimulation.fr/eocoe-ii/code-demonstrators.git
The goal by the end of the project (M36), is to provide up-to-date versions of EoCoE-II codes containing all the optimization and refactoring work that has been invested over the project lifetime.

2.1. Scientific Challenges and Flagship Codes

Although the optimization and refactoring tasks undertaken in the four technical work packages are largely self-contained, they do require close cooperation between the application developer and HPC/tool experts, and in some cases additional cooperation between WPs. For example, the Parallel Data Interface package is developed in WP4, but its interfacing and implementation are performed within WP2 where most of the expertise on the algorithmic side of the application lies. Likewise, highly optimized linear algebra packages are developed in WP3 but in practice, designed, tuned and implemented via exchanges between WP1, WP2 and WP3.

The interrelationships between the various exascale preparation subtasks for each flagship code are depicted in the Gantt chart of Figure 2. This chart is not exhaustive but serves to show how the final Demonstrator 2 milestone relies on the collective completion of various strands of optimization from single-node optimization, I/O refactoring, adoption of scalable, GPU-capable linear algebra kernels. In the case of the meteo and hydrology challenges, the final workflow foresees the adoption of fully load-balanced ensemble runs including data assimilation. For more details on each subtask, the reader may refer to the references given in Table 1 to find the relevant sections in the M18 deliverables.

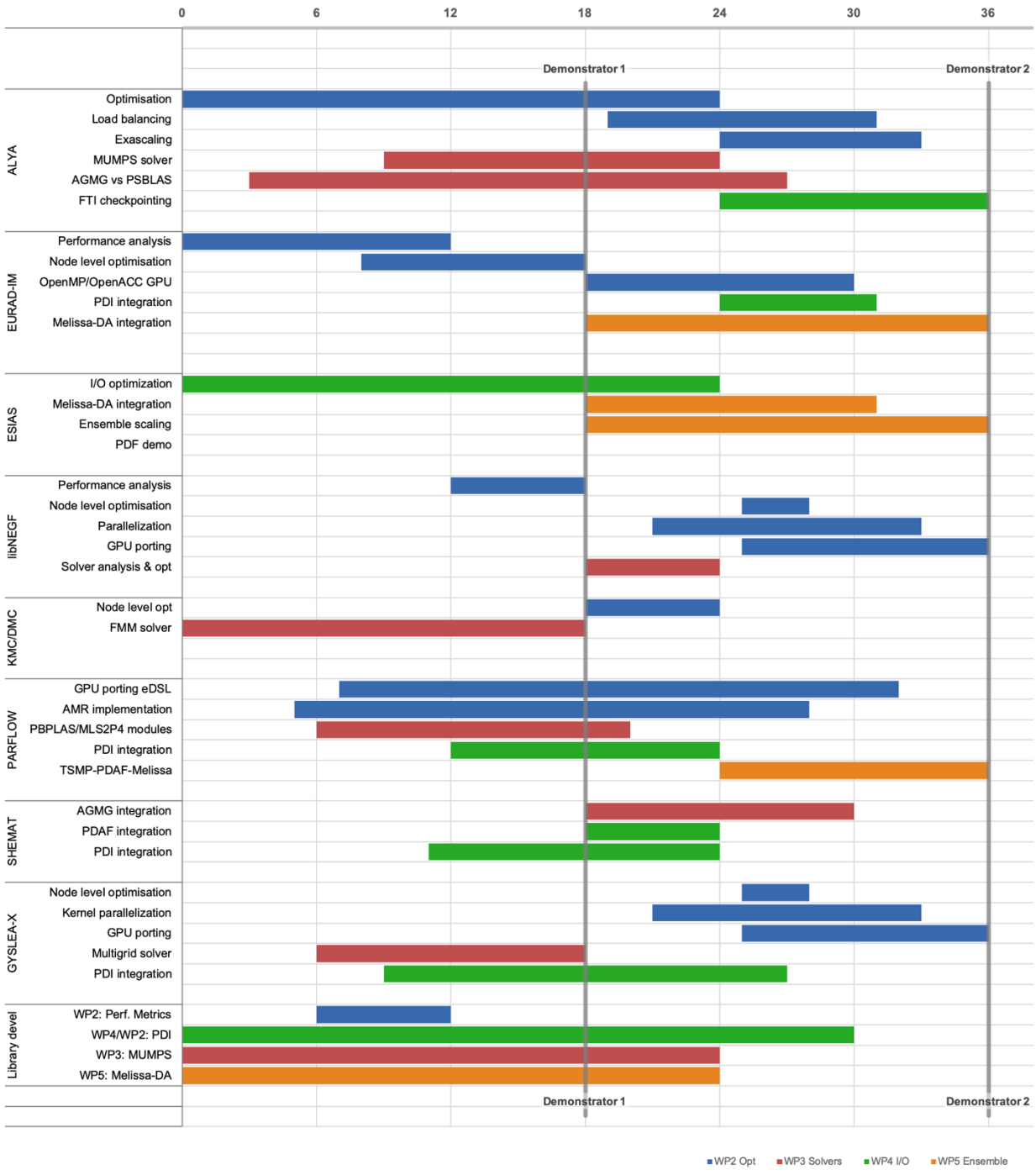

Figure 2: Global Gantt chart of Exascale activity invested in EoCoE-II Flagship codes over the project lifetime. The colour coding here refers to the technical WP in which the task is anchored (see legend). Each bar represents a subtask or part of a subtask either in the original DoW, or which may have arisen because of a strategic change. Milestones at M18 and M36 refer to the Code Demonstrators described in WP1 (D1.2)

In the following, we highlight some of the measures which have been successfully implemented in each code at the time of writing and which in some cases are already available in the M18 demonstrator version.

2.2. ALYA

The high-performance computational mechanics code Alya is designed to solve complex coupled multi-physics/ multi-scale engineering problems, in this case, to model wind power from the rotating turbine blade level up to an entire wind farm including complex terrain. Performance enhancements which have been undertaken in the first half of the project include:

- Node level optimization and vectorization in cooperation with FAU
- Refactoring of the parallel core of Alya. Possibility of working with empty subdomains. Treatment of periodic boundary condition. Preliminary tests on a pipelined CG run on up to 32K cores.
- Significant progress with co-execution on heterogeneous clusters (CPU + GPU) using OpenACC, including tests on PizDaint and MareNostrum IV
- Fast and scalable geometric mesh partitioning based on a space-filling curve
- Extensive testing of optimized linear algebra packages for the most compute-intensive part of the code. These include Maphys, AGMG, and PSBlas/MLD2P4 iterative solvers; and Pastix and MUMPS direct solvers. Details of these comparisons can be found in D3.2, an example of which is shown in Figure 3 below.

Cores	Total Million Unknowns	AGMG - CPU time [s]	PSBLAS - CPU time [s]
48	5.6	0.419	0.368
96	5.6	0.231	0.192
192	5.6	0.130	0.099
384	44.8	0.743	0.606
768	44.8	0.430	0.316
1536	44.8	0.293	0.169
3072	358.4	0.524	0.523
6144	358.4	0.543	0.294
12288	358.4	0.843	0.205

Figure 3: Scaling comparison between the AGMG and PSPLAS algebraic solvers

As yet there have been no show-stoppers with the roadmap and indeed some tasks have been started earlier than intended. However, as the developers point out in D1.2, resources for testing at scale are difficult to come by both locally and within the available PRACE network of Tier-0 machines. This issue needs to be addressed both for this and other flagship codes capable of scaling beyond 10^5 cores.

2.3. ESIAS/EURAD-IM

In the meteorology challenge, solar and wind power prediction is performed using a multi-code framework working together. These codes, WRF (Weather Research Forecasting model) for meteorological analyses, and EURAD-IM for air quality assessments (with an aerosol focus for EoCoE) are coupled together within the ESIAS framework to allow large ensemble simulations above 1000 members. Broadly speaking, optimization work was performed on EURAD-IM and I/O refactoring in tandem with data assimilation handling was made with ESIAS. Particular improvements of EURAD-IM include:

- Extensive code refactoring (with FAU) including change of data structure for vectorization and dynamic memory management, thus eliminating major load imbalances between the different MPI processes.
- First steps towards hybrid parallelization (MPI+OpenMP) have been made and show good performance for selected code fragments.

Work on ESIAS itself – particularly the improvement on I/O in WP4 and the integration of Melissa-DA package developed in WP5 – were pushed back somewhat because of a hiring delay at FZJ, but these tasks are now getting back on track and are scheduled after M18.

2.4. libNEGF

Plans for the original materials code PVnegf were dealt a major blow just before the EoCoE-II project got underway when its lead developer left FZJ to take up a new position in industry. At the F2F meeting in Brussels (M9) a strategic decision was made to replace this code by libNEGF, whose lead developer Alessandro Pecchia (CNR) was invited to join the project. The revised roadmap for this code is detailed in Section 5.5 of D1.2. Despite this almost fatal setback, rapid progress has already been made on several open issues with the new library:

- Scalability up to at least 36000 cores (see Figure 4 and D2.2, Section 7.2)
- Refactoring underway for multi-GPU operation which will soon be tested on the newly commissioned (70 PFlops) JUWELS Booster at FZJ
- New collaboration with MaX-II CoE on optimization and parallelization of the Wannier90 kernels to enable full quantum transport simulations
- Revised goal to deliver a pre-exascale code neXGf by M36

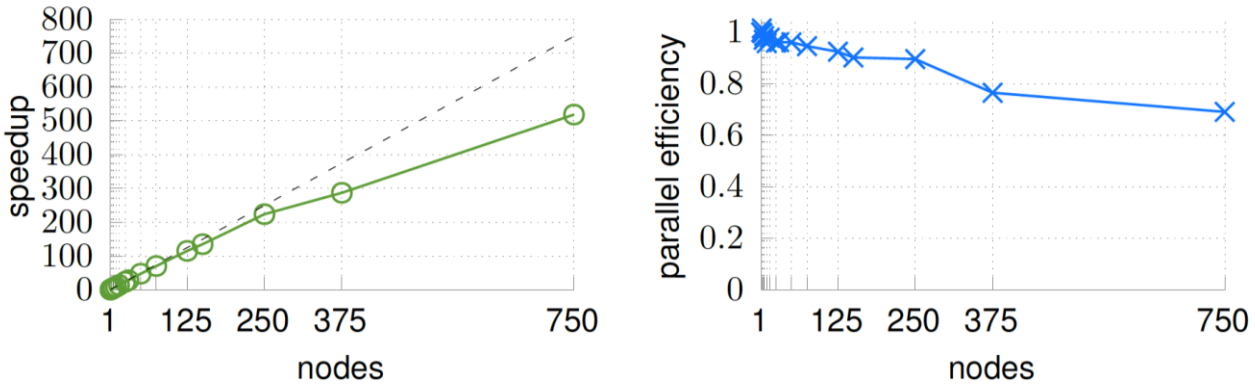

Figure 4: Scaling of the 6x6 silicon supercell test inputs of the optimized version of LIBNEGF on up to 750 JUWELS node with 4 MPI ranks per node and 12 OpenMP threads per rank

2.5. KMC/DMC

The pair of Monte-Carlo codes developed by UBAH for modelling of perovskite photovoltaic devices and organic solar cells are less advanced than the DFT library libNEGF in terms of exascale readiness but represent important showcases in this energy domain. Algorithmic improvements and other optimization work on the Bath KMC code include the implementation of a new FMM-based electrostatic solver for KMC, enabling access to physical systems that contain millions of charges with good parallel scaling. We demonstrated good strong and weak scaling for 128 million charges on up to 4096 cores of an Intel Skylake Gold platform and 8192 cores of the Isambard Marvell ThunderX2 ARM machine.

2.6. Parflow

ParFlow (v3.2) is a massively parallel, physics-based integrated watershed model incorporating fully coupled dynamic 2D/3D hydrological, groundwater and land-surface processes for large scale problems. Its credentials as a flagship code for EoCoE-II were already signalled in the previous EoCoE funding period, where scalability of over 260k cores of the JUQUEEN machine was shown. Despite this advanced starting point, the

ambition to push the code to genuine exascale capability is hard-wired into the EoCoE-II work programme, with several noteworthy early successes:

- a working multi-GPU (CUDA) version showing 20x speedup over the earlier pure CPU ParFlow - see Figure 5 below
- Selection for the 70 PFlop JUWELS GPU Booster Early Access Programme (Sept-Oct 2020)
- Preparation of full integration of optimized PSBLAS/MLD2P4 linear algebra solvers for the compute-intensive Richard's equation for groundwater flow. This will be achieved via the implementation of an interface for the PSBLAS/MLD2P4 solvers to the Kinsol library.
- Integration of the PDI interface allowing seamless switching between HDF5 and netCDF parallel I/O formats
- Preparatory work for full integration of Melissa Data Assimilation (WP5)

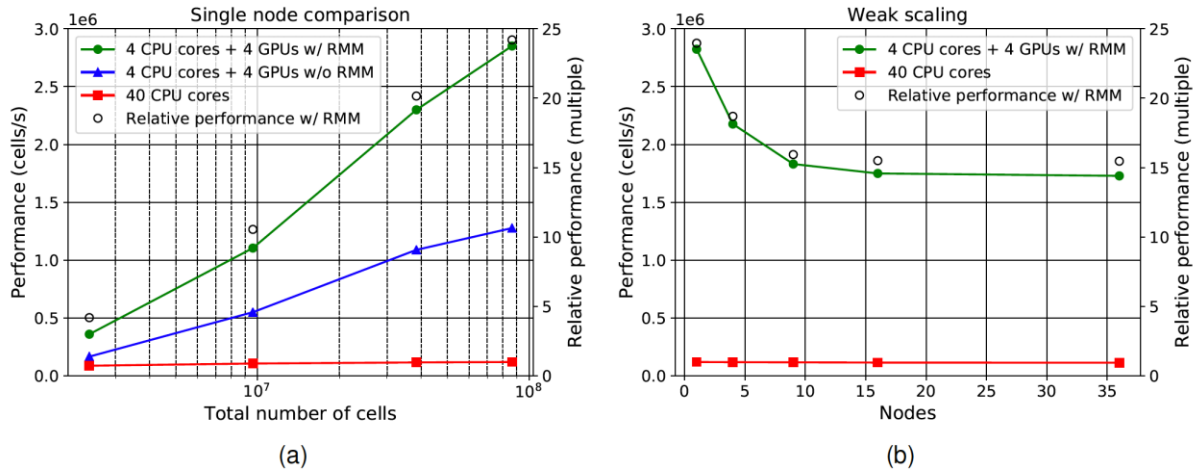

Figure 5: CPU vs GPU performance of ParFlow with a) single-node comparison and b) weak scaling comparison

2.7. SHEMAT Suite

For the SHEMAT-Suite, a software framework for geothermal reservoir modelling, intensive I/O processes represent one of the main bottlenecks, both for data initialization and post-processing. A major task has been to integrate PDI into SHEMAT-Suite in order to decouple the backend I/O library from the simulation. Preparation for this step has been completed and implementation is underway. A further improvement has been the integration of the AGMG linear algebra solver via the PETSc library.

2.8. GyselaX

The Gysela code is one of the leading established tools in magnetic fusion research and also starts from a fairly advanced position in terms of HPC readiness. Despite an initial setback after one of the lead developers left the project, a number of achievements can be highlighted at this midterm point:

- Successful porting and scaling tests on the TGCC Irene-AMD machine, demonstrating strong scaling up to 98 304 cores with 59% efficiency - Figure 6
- Quantitative benchmarking to compare performance on Haswell and ARM architectures
- Cooperation with RIKEN to port code to Post-K Fujistu machine
- Development of scalable geometric multigrid solver on a stretched polar grid

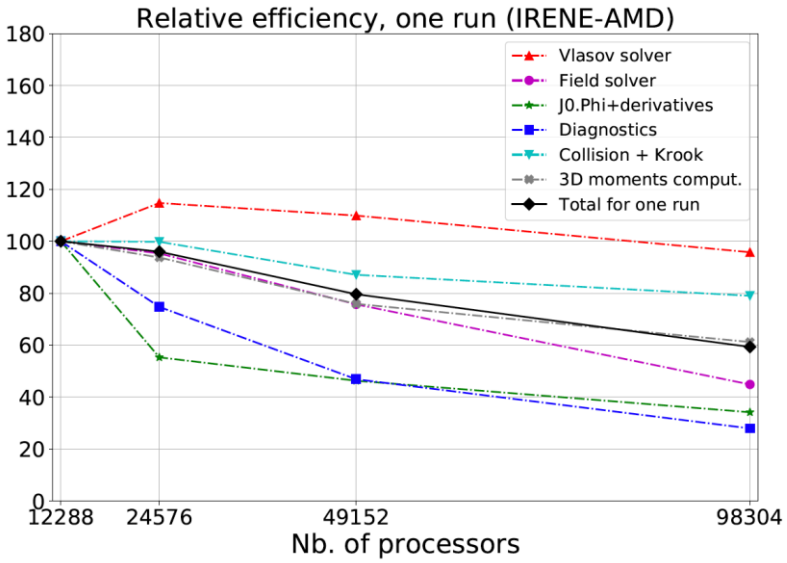

Figure 6: Strong scaling of GyselaX on the Irene-AMD machine (TGCC)

3. Technical Challenges: HPC libraries and tools for exascale

The technical challenges posed by bringing the flagship codes to exascale readiness levels represent the backbone of EoCoE-II and are of course embedded within WP2-WP5. While much of the effort displayed in Figure 2 is dedicated to library design for or implementation in one of the flagship/satellite codes, some of the TC work is reserved for stand-alone transversal efforts to further develop the HPC libraries and tools in response to the latest architectural developments and supercomputer availability. These efforts are briefly summarized in the following: detailed descriptions can again be found in the relevant deliverable D2.2, D3.2, D4.1 and D5.1

3.1. Metrics definition and performance tools

In EoCoE-II careful attention has been paid to covering all important aspects of code optimization and providing state-of-the-art analysis tools to allow application developers to identify bottlenecks and assist the refactoring process. To this end the FAU group has provided a valuable addition to the EoCoE consortium, making their LIKWID analysis tool available. Together with the Score-P and Paraver performance suites provided through the close cooperation with the POP CoE, the project has been well equipped to address the tasks foreseen in WP2. Code performance is measured with the help of quantitative metrics developed within the EoCoE-I project, the definition of which is given in Sections 7.3 and 7.4 of deliverable D2.2, along with the tools used to extract them via the automated JUBE system.

3.2. Linear Algebra packages: Efficient AMG preconditioners for multi-GPGPU clusters

Considerable efforts at CNR are devoted to designing the efficient implementation of an Algebraic MultiGrid (AMG) preconditioner tailored to recent generations of Nvidia Graphics Processing Units (GPUs) already available in the sequential open-source package BootCMatch (Bootstrap algebraic multigrid based on Compatible weighted Matching). The inherent parallelism of modern GPUs in all the kernels involving sparse matrix computations is exploited both for the setup of the preconditioner and its application in a Krylov solver, outperforming preconditioners available in the well known hypre library as well as in the Nvidia AmgX library. A recent example of this work drawing on tests made with EoCoE-II applications with up to 10^{10} unknowns can be found in Ref.[1], one result of which is shown in Figure 7. The hybrid CPU/GPU approach

permits savings in the solve time and, equally importantly, large savings in the energy consumption, since to deal with the same number of dofs at comparable execution time we need to use fewer nodes

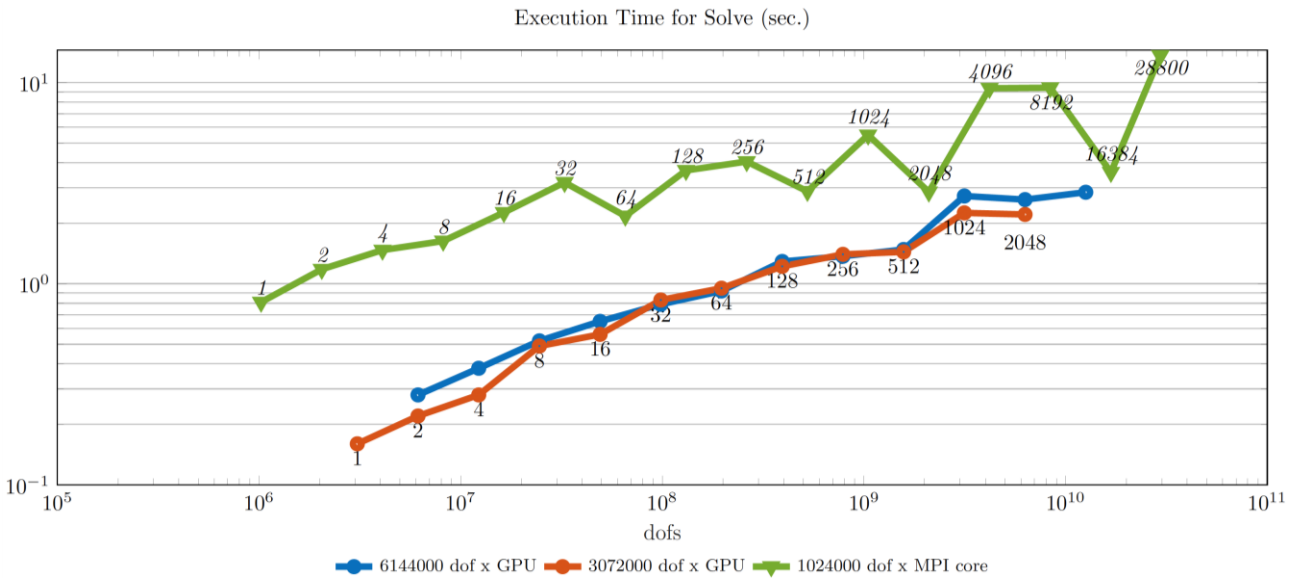

Figure 7: Time-to-solution time (in seconds) for a 3D Poisson problem computed via the Conjugate Gradient method, preconditioned by one of the Algebraic Multigrid preconditioners from AMG4PSBLAS in a weak-scalability setting on the Piz Daint supercomputer. The x-axis represents the degrees of freedom (dofs) while the labels denote the number of MPI cores (green line) or number of GPUs (blue, red).

[1] P. D'Ambra, Durastante and Filippone, 'AMG Preconditioners for Linear Solvers Towards Extreme Scale', https://arxiv.org/abs/2006.16147

3.3. I/O strategies for exascale

In the context of the data transport issues tackled in WP4, there are three generic topics relevant to Exascale which also receive attention both inside EoCoE and in-kind efforts of partners outside the project:

- Leveraging of I/O cache device infrastructure capabilities: All future Exascale systems have to deal with a massive amount of I/O, which does not scale as fast as the computational capabilities when utilizing classical parallel filesystems. Therefore, intermediate local or global cache devices have to be used. Within EoCoE we test the IME cache capabilities for the EoCoE codes (as a representative type of this architecture) and try to allow better cache utilization by introducing SIONlib as a cache-aware API for IME.

- In-transit data compression: Having more data and more compute capabilities also mean, that compression algorithms can move into the computational part of the simulation (instead of having it as a pure post-processing step). This allows reducing data size on the fly to lower the I/O-Compute-Scalability gab. These capabilities have been enabled in ParFlow, for example.

- Fault-Tolerance handling: Running on Exascale level also increases the chance for system failures due to the number of involved computational elements. By utilizing the FTI library for EoCoE codes (e.g. for Alya so far), the impact of a system fault can be mitigated on the fly.

3.4. Melissa-DA framework

The Melissa-DA framework for large scale data assimilation developed in WP5 has been tested on two supercomputers, JUWELS (Germany) and Jean-Zay (France). The current prototype supports the ParFlow hydrology code and the EnKF assimilation method, with experiments scaled up to 1024

ensemble members so far. It is anticipated that the coupled ESIAS/EURAM-IM application will push the framework to even larger ensembles in the 2nd half of the project.

4. Summary

As pointed out in deliverable D2.2, at the halfway point of the project, no code is entirely ready to run with all the planned optimizations and developments foreseen in Figure 2 of Section 2. Nevertheless, a number of applications are already very advanced (ALYA, ParFlow, GyselaX) and already exhibit excellent performance. In many cases, applications have been delayed in their development schedule due to late recruitment (EURAD-IM, Gysela, SHEMAT-Suite, ParFlow), loss of key developer staff (libNEGF, Gysela), the impact of the Covid-19 crisis (libNEGF), and problem of access to computing resources (Meso-NH).

Regarding some of the KPIs which were conceived for the original DoW (HPC libraries integrated into EoCoE-II codes; applications capable of running on up to 80% of future PRACE machines; participation in EuroHPC demonstrator facilities), we are confident that these goals will be comfortably met within the next 12 months. For example, optimized linear algebra packages have been integrated into ALYA, ParFlow, SHEMAT-Suite and GyselaX; Melissa-DA is in the process of being incorporated into ParFlow, ESIAS/EURAD-IM and the PDI library is or will become a standard part of at least 4 EoCoE-II applications. The flagships ALYA, ParFlow, ESIAS, libNEGF and GyselaX all exhibit excellent scalability, with ALYA, ParFlow and libNEGF also capable of multi-GPU operation.

These applications are therefore ideal candidates to evaluate Exascale Demonstrator/Pre-exascale machines and are already used for PRACE projects. On the other hand, benchmarking on current PRACE machines has not been easy for the EoCoE-II code developers, particularly since rigorous testing at scale (e.g. beyond 10^5 cores) can easily consume millions of core-hours and EoCoE-II test allocations are limited. Despite this obstacle, most of the applications are on track and should be ready for the first pre-exascale machines due to be commissioned before the end of 2021.