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

D7.5
Final 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:	D7.5
	Deliverable Nature	Report
	Dissemination Level:	PU *
	Contractual Date of delivery:	M30 30/06/2021
	Actual Date of delivery	M42 30/09/2022
	EC Project Officer	Matteo Mascagni

* 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:	D7.5
	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

As explained in the mid-term edition of this deliverable, the ECG oversees the co-design spirit 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. The purpose of this document is to provide a final assessment 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, highlighting 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 exhaustive reports to be found in the M36/42 Deliverables, but may also include activity involving the flagship codes and libraries which have taken place outside the original DoW scope.

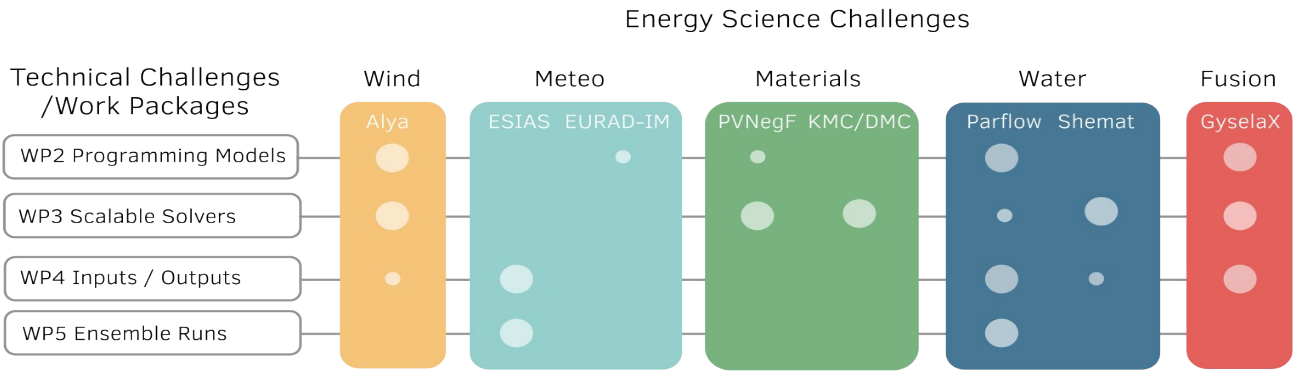

Figure 1: Original co-design concept of EoCoE-II, identifying interaction points between Technical and Energy Science (WP1) challenges where project resources are concentrated within work-packages WP2-WP5.

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 four technical work packages WP2-WP5, and are documented in detail in the M36/M42 deliverables D2.3, D3.4, D4.4 and D5.3. Where appropriate, performance enhancements enabling new scientific advances with the so-called 'demonstrator' codes are also highlighted in D1.3. Additional transversal activities in WP2-WP5 necessary to further develop exascale tools and libraries are described later in Section 3.

Code	Lead Partner	Current implementation	Hardware	Exascale-relevant activity	Deliverable Reference	Task in DoW
Alya	BSC	MPI for multi-node	CPU and GPU (NVIDIA)	Demonstrator Performance opt LA solver In-situ visualization	D1.3, Section 3.3 D2.3, Section 5.2 D3.4, Section 6 D4.4, Section 7.2	1.1 2.2 3.4 4.4
EURAD-IM	FZJ	MPI for multi-node	CPU and GPU	Demonstrator Performance opt I/O refactoring Melissa DA	D1.3, Section 4.4 D2.3, Section 6.3 D4.4, Section 5.2 D5.3, Section 3	1.2 2.3 4.2 5.2
ESIAS	FZJ	MPI for multi-node	CPU	Demonstrator I/O refactoring FTI checkpointing Melissa DA	D1.3, Section 4.4 D4.4, Section 5.2 D4.4, Section 6 D5.3, Section 3	1.2 4.2 4.3 5.2
LibNEGF	CNR	MPI	CPU and GPU (NVIDIA)	Demonstrator Performance opt	D1.3, Section 5.4 D2.3, Sections 7.1-7.4	1.3 2.4
KMC/DMC	UBAH	MPI	CPU	Performance opt FMM solver	D1.3, Section 5.4 D1.3, Section 5.3	1.3 1.3
ParFlow	FZJ	C/ MPI	CPU and GPU (NVIDIA + AMD)	Demonstrator Performance opt LA solver In-situ visualization Melissa DA	D1.3, Section 6.6 D2.3, Section 8.2 D3.4, Section 4.1 D4.4, Section 7.2 D5.3, Section 3	1.4 2.5 3.2 4.4 5.3
SHEMAT-Suite	RWTH	Fortran / OpenMP	CPU	Demonstrator Performance opt LA solver	D1.3, Section 6.6 D2.3, Section 8.3 D3.4, Section 4.2	1.4 3.2 4.4
Gysela	CEA	Fortran / MPI / OpenMP	CPU (x86 and ARM)	Demonstrator Performance opt LA solver I/O refactoring	D1.3, Section 7.4 D2.3, Section 9.4 D3.4, Section 5.1 D4.4, Section 5.1	1.5 2.6 3.3 4.2

Table 1: Summary of flagship codes and location of optimisation tasks within M42 deliverables and original Description of Work. See also Table 4 in D2.3 for more details on the porting to various hardware platforms.

As previously noted, 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. Reallocation of resources was only necessary in a few cases where personnel changes forced some adjustments. A notable and ultimately successful priority shift was the replacement of the PVnegf code by libNEGF due to the PVnegf lead developer leaving FZJ. In this case, resources were rechannelled from FZJ to CNR, where the libNEGF lead is hosted and the corresponding reprioritisation made in the affected WPs. This switch has proved highly fortuitous since libNEGF has become one of the EoCoE Exascale demonstrators. Staff fluctuations at RWTH and FZJ meant that the envisaged common platform (ExaTerr) for ParFlow and SHEMAT Suite also had to be cancelled (see section 8.4 of D2.3)

Code Demonstrator repository

In deliverable D1.3, the description within the code demonstrator section refers to the status of the flagship codes (and satellite codes, whenever possible) at M42. This description includes an example of scientific simulation and related results. For each code, the current software has been provided within a protected Gitlab repository, accessible to 3rd parties by request.
https://gitlab.maisondelasimulation.fr/eocoe-ii/code-demonstrators.git

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 have almost always required close cooperation between the application developer and HPC/tool experts, and in some cases additional cooperation between WPs. For example, the Parallel Data Interface (PDI) 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 relied 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 foresaw the adoption of fully load-balanced ensemble runs including data assimilation, but this could not be completed. For more details on each subtask, the reader may refer to the references given in Table 1 to find the relevant sections in the M36/42 deliverables.

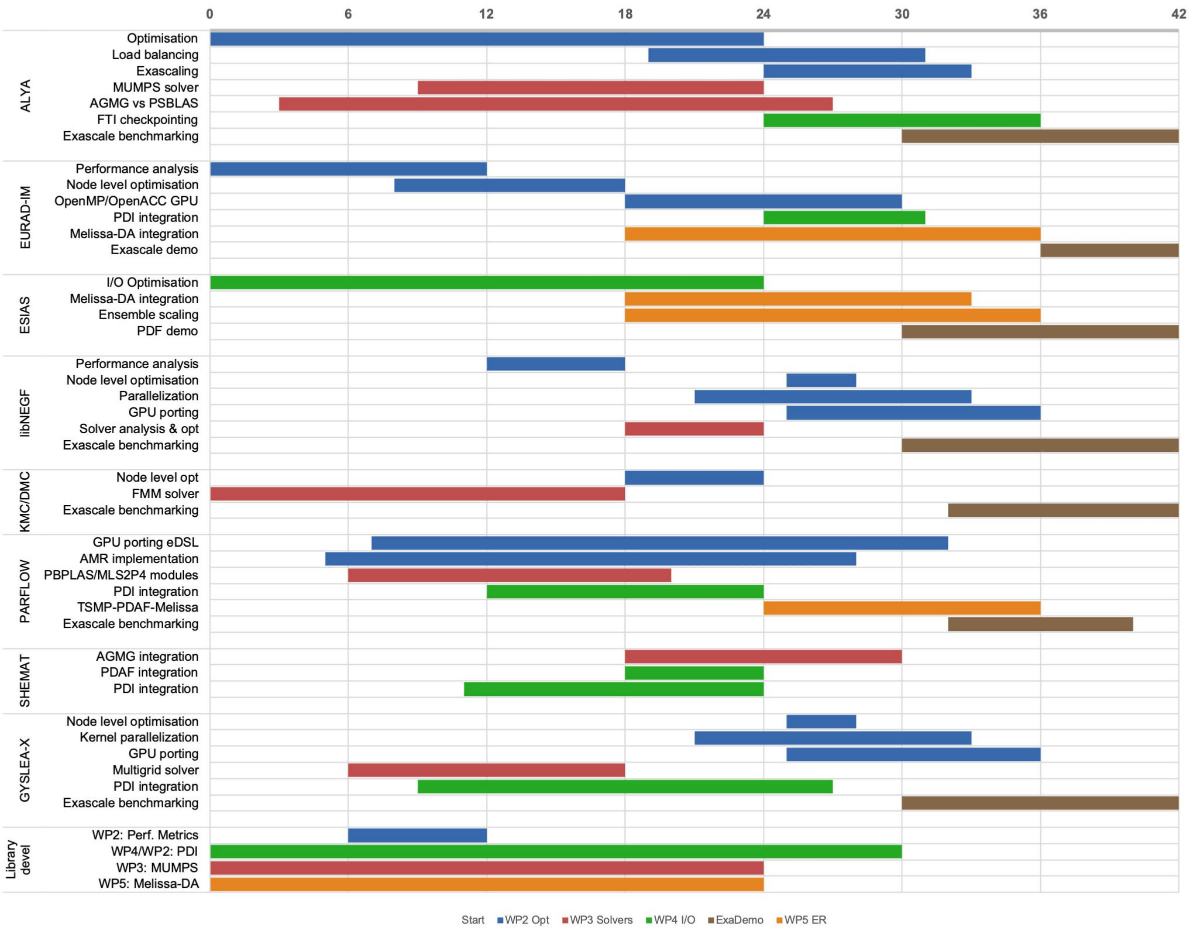

Figure 2: Global Gantt chart of Exascale activity invested in EoCoE-II Flagship codes over the project lifetime. The colour coding here refers to the technical WP in which the task is anchored (see legend). Each bar represents a subtask or part of a subtask either in the original DoW, or which may have 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 and benchmarked in each code at the time of writing and which in some cases are already available in the demonstrator versions.

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 second half of the project include:

- 80X speedup of the volumetric element assembly on GPU, the most time-consuming kernel of the ALYA code. On an A100 NVIDIA GPU, this routine was benchmarked at 50% of the peak performance.
- Benchmarking on most European supercomputers: ALYA is now part of the Unified European Applications Benchmark Suite for CPU and GPU architectures.
- Taking advantage of the improvements for the GPU, the volumetric element assembly on the CPU is now almost 4× faster.
- The energy efficiency of Alya's assembly on the GPU is now more than five times better than on the CPU, as one should expect from the energy efficiency of the supercomputers reported in the

Top500 list. Instead, at the beginning of the project, the CPU implementation was more energy efficient than the GPU one. Some people within BSC thought low-order Finite Element assembly was not well suited for GPUs, and little could be done.

- One of Alya's main limitations in the path towards exascale was its lack of algorithmic scalability. Until recently, most of Alya's iterative linear algebra solvers were classical solvers like GMRES or Conjugate Gradient (CG) – with or without Deflation. For such solvers, it is well known that the number of iterations increases as the problem size increases. Within EoCoE, we have interfaced Alya with the multigrid preconditioner library AMG4PSBLAS. This has allowed us to solve ABL flows over complex terrain with no significant increase in the number of iterations from 32 million elements to 16 thousand million elements. This is the biggest problem (in the number of elements) that has been solved with Alya. Moreover, thanks to the excellent algorithmic scalability of the linear solver, it is the first time that the entire code obtains correct weak scalability for cases where an implicit time treatment is used
- Alya has been coupled with PDI to endow it with in-situ visualization and analysis capabilities.

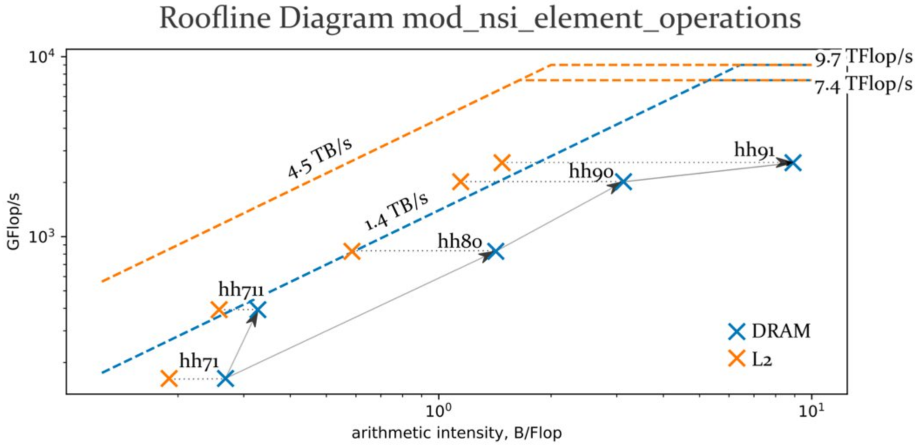

Figure 3: Roofline diagram of the optimized versions created during EoCoE-II for both the DRAM and the L2 cache bandwidth

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.

To demonstrate the parallel performance of the code, the scaling behaviour is investigated for a single ensemble member and then extrapolated to 128 members. The integration time is 24 hours. The scaling behaviour in the scaling figure shows the associated speedup as a function of the number of cores used. The scaling shows that using 16 cores per ensemble can still have 3 times of speed-up that cost around 15 minutes to produce a 24 hours simulation. In practice, we choose 12 cores per ensemble member that requires a total of 128 nodes for the ensemble simulation to keep the full use of a node.

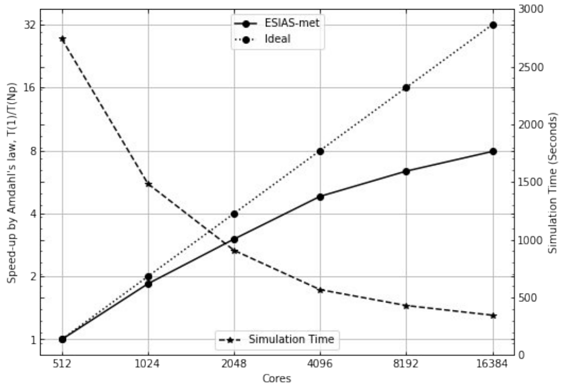

Figure 3: Scaling of the ESIAS-met for ensemble simulation with 128 members by strong scaling

2.4. libNEGF

Significant progress on refactoring and scaling of libNEGF has been made on several open issues:

- Scalability up to at least 36k cores with CPU version.
- Extensive refactoring of kernel routines for multi-GPU operation, benchmarked on the (70 PFlops) JUWELS Booster at FZJ

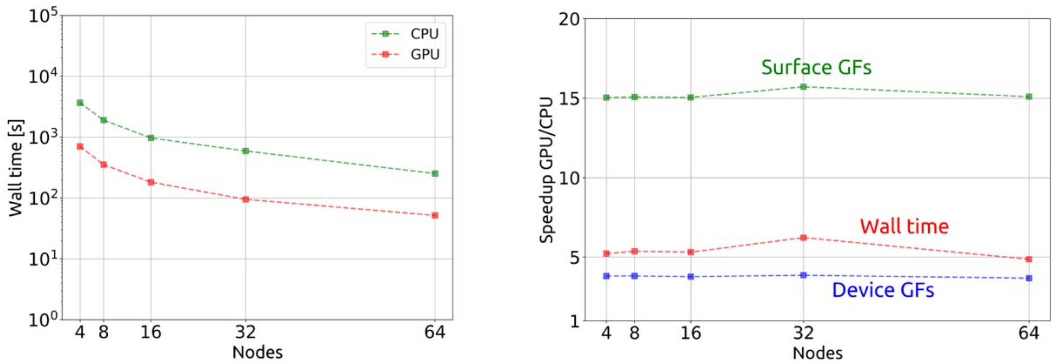

Figure 4: Speedup of GPU run compared to CPU for a 6x6 silicon supercell test with libNEGF For more details, see D2.3, section 7.2

2.5. 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. In EoCoE-II, most of the effort focused on GPU porting, where some impressive improvements could be demonstrated:

- a working multi-GPU (CUDA) version showing over 20x speedup over the earlier pure CPU ParFlow - see Figure 5 below
- Development of a prototype AMR version of ParFlow using the p4est library.

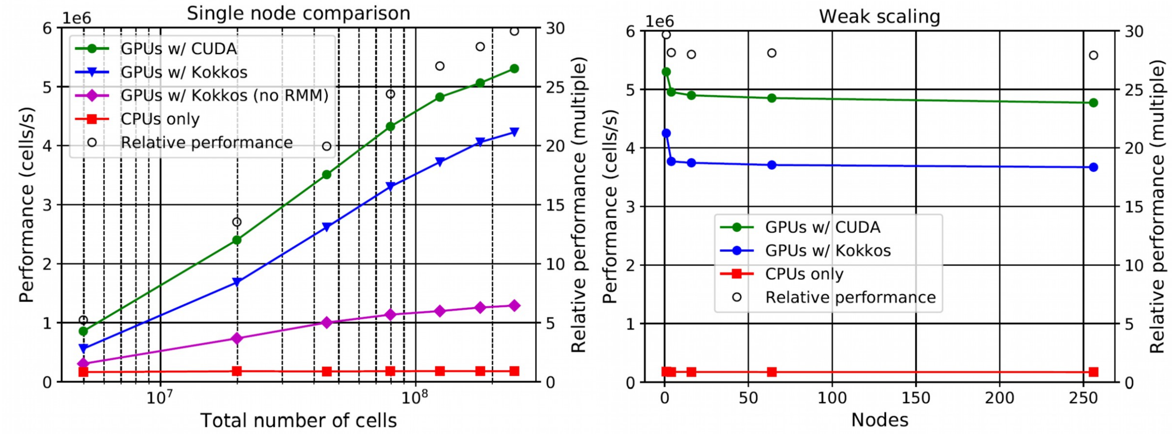

Figure 5: CPU vs GPU performance of ParFlow with a) single-node comparison and b) weak scaling comparison. Data is shown for both Kokkos and CUDA GPU implementations.

2.6. GyselaX

The Gysela code is one of the leading 5-dimensional gyrokinetic codes for plasma turbulence and transport in magnetic fusion research, aiming at understanding transport losses in current devices and at proposing routes towards improved confinement regimes. It also started in EoCoE-II from a fairly advanced position in terms of HPC readiness, highly parallelized up to hundreds of thousands of CPU. Significant highlights include:

- Over 70% improvement in runtime over a 12-month optimization effort - Figure 6.
- Successful porting and scaling tests on the pre-exascale CEA-HF machine, demonstrating weak scaling up to 810k cores with better than 60% efficiency (without I/O) – Figure 7

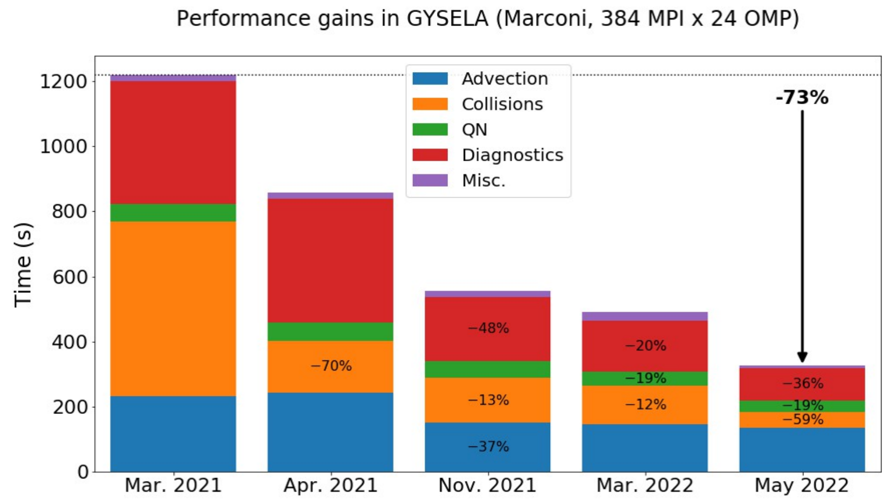

Figure 6: Optimisation performed between March 2021 and May 2022 on the main kernels of GYSELA - see D2.3, section 9.4.4

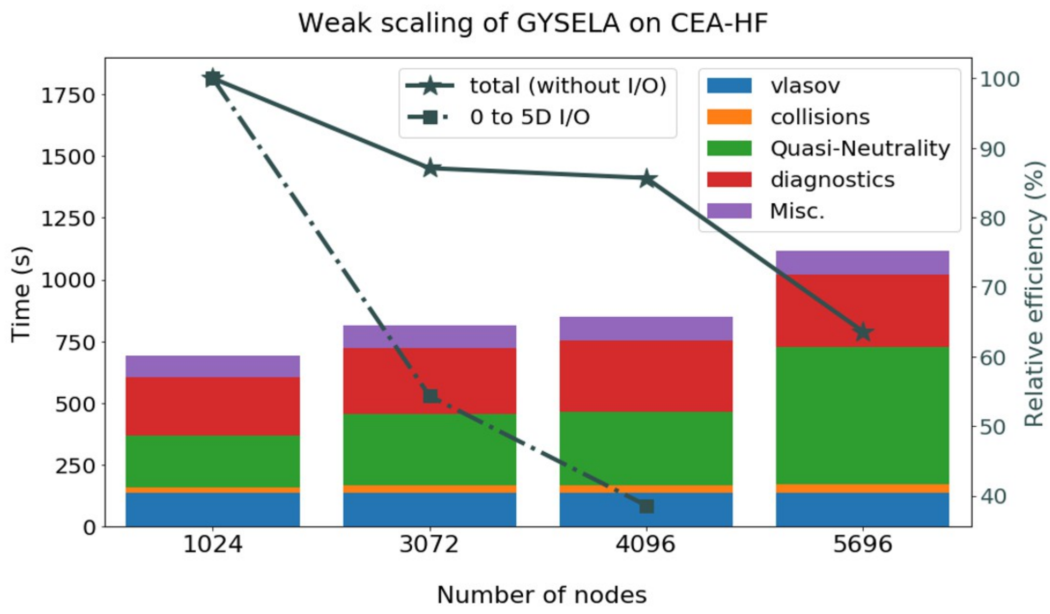

Figure 7: GYSELAX weak scaling from 1024 to 5696 nodes of the CEA-HF supercomputer (6330 AMD EPYC 7763 nodes = 810 240 cores). For details, see D2.3, section 9.4.4

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 implementation in one or more 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 D2.3, D3.4, D4.4 and D5.3.

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. Examples of these and other tools to analyze dynamic load balancing, for example, can be found in deliverable D2.3.

3.2. Linear Algebra packages

In WP3 considerable effort has been devoted to improving the generic scalability of different linear algebra solvers, independently of the scientific applications within WP1. Progress on this front will likely benefit users of other codes well beyond the EoCoE-II project. As already widely reported in Deliverable D3.4, we have revised and improved the PSBLAS linear algebra package, implementing iterative solvers, we have started a novel package AMG4PSBLAS of preconditioners, and we have defined a combined toolkit PSCToolkit containing both packages, together with some other support tools. An extensive discussion on the new software libraries and on scalability results have been included in Deliverable 3.3 and in a paper published on SIAM Journal on Scientific Computing (Vol. 43, Issue 5, 2021), where we present weak scalability results on up to 27000 CPU cores and up to 2048 GPUs of Piz Daint for linear systems with sizes up to O(1010) unknowns. Finally in Deliverable 3.4, we discuss some comparisons with the NVIDIA AMGX library (rel. 2.2.0) on up to 1024 GPUs of Marconi 100. Operator Complexity (Fig.8.a) shows the advantage in using the solvers from PSCToolkit in terms of memory footprint and computational complexity of the preconditioners. Weak scalability in time to solve systems of dimension up to 6.1 × 109 unknowns (Fig.8.b) shows that our solvers (see PSCTVA8S) generally gives a significant benefit with respect to state-of-the-art solvers for hybrid architectures.

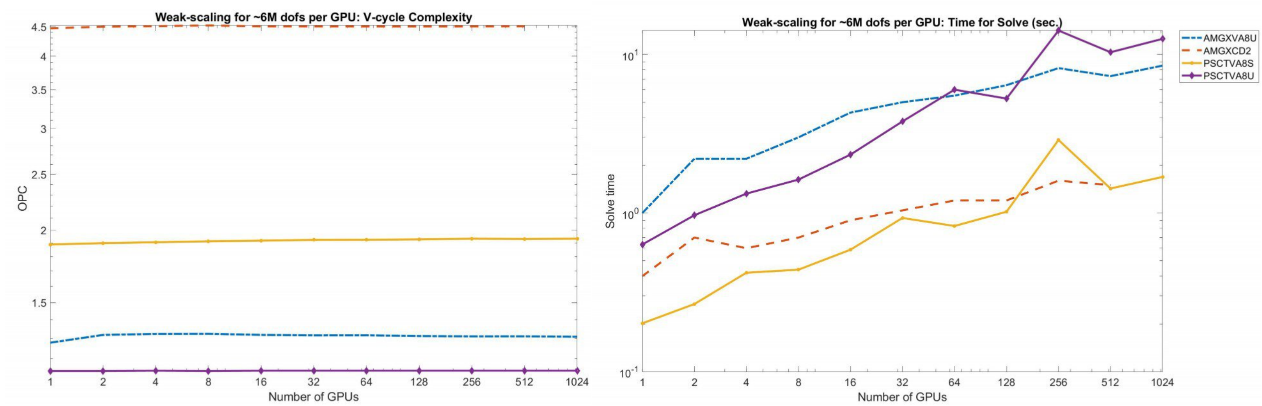

Figure 8: Weak scaling of a) complexity and b) time to solution of various solvers in the PSCtoolkit.

3.3. I/O strategies for exascale

In the context of the I/O and data transport issues tackled in WP4, there are three main 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. The actual I/O bandwidth 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 solution by DDN for the EoCoE codes (as a representative type of a global cache infrastructure) and try to allow better cache utilization by introducing SIONlib as a cache-aware API for IME. This setup was tested for Gysela checkpoint writing in particular using either the default HDF5 task-local based format but also a SIONlib based checkpointing format. In addition the cache was also tested for the regular ParFlow PBF output format. More details can be found in Deliverable D4.4.

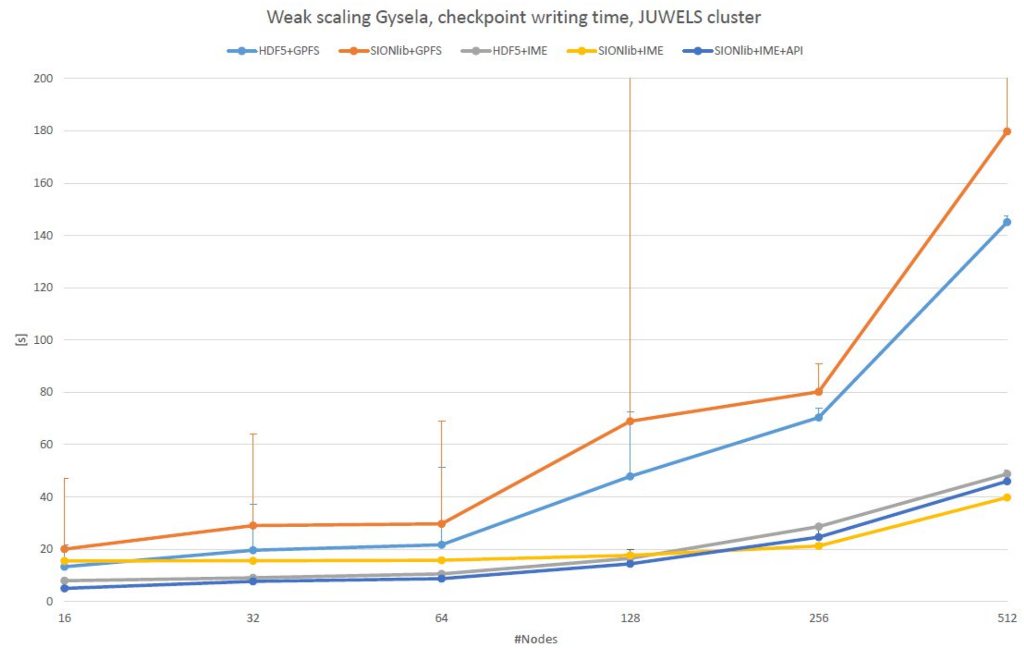

Figure 9: Checkpoint writing time (for two checkpoints in total) of Gysela (v. 32.0) on JUWELS Cluster (using 48 cores per node) using a classical parallel file system (IBM Spectrum Scale) backend and the IME based High Performance Storage Tier of the system.

- 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. Depending on the actual data and selected compression algorithm a sweet spot between the additional compression overhead and the saved I/O time can be found. Within the frame of the project, these capabilities have been implemented into ParFlow by allowing ZLIB based compression for the NetCDF4 based output files of ParFlow.

- 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), the impact of a system fault can be mitigated on the fly. In addition FTI was also implemented underneath of Melissa-DA to allow fault tolerance capabilities to be used within ensemble based calculations such as ESIAS. This general work on resiliency capabilities for ensemble based applications was also driven by the introduction of a particle filter framework, which also allows to utilize local cache infrastructure. This approach was tested, in the frame of a separate internship on the Fugaku system at Riken.

3.4. Melissa-DA framework

The Melissa-DA library for large scale data assimilation developed in WP5 made important progress in 'field-testing' with the real-world applications ParFlow and the coupled ESIAS/EURAM-IM framework. highlights are worth a special mention here:

- Proof-of-principle demonstration of the Melissa-DA library for performing ensemble runs with over 1000 members, showing up to 2X speed-up compared to the conventional PDAF library with ParFlow.
- Scalability of Melissa-DA up to nearly 50k cores on the Fugaku supercomputer.
- Extensive testing on European supercomputers.

4. Summary

At the start of the EoCoE-II project, a series of optimizations and refactoring measures were foreseen for each nominated flagship code to prepare them to run on future exascale systems. Depending on the application, these measures included single-node optimization, load balancing, replacement of linear algebra libraries, incorporation of the Melissa-DA data assimilation framework or I/O libraries and in-situ visualisation. At the project conclusion we can report that at least four of these codes, ALYA, libNEGF, ParFlow and GyselaX, have progressed sufficiently that they can be routinely on the largest European Tier-0 systems currently available and already exhibit excellent performance.

Regarding the KPIs which were conceived for the original DoW (HPC libraries integrated into EoCoE-II codes; 5 applications capable of running on up to 80% of future PRACE machines; participation in EuroHPC demonstrator/pre-exascale facilities), we can state these goals have been reached to a large extent. For example, optimized linear algebra packages have been integrated into ALYA, ParFlow, SHEMAT-Suite and GyselaX; Melissa-DA has been fully incorporated into ParFlow, ESIAS/EURAD-IM and the PDI library is an optional standard part of at least 4 EoCoE-II applications. The flagships ALYA, ParFlow, ESIAS, libNEGF and GyselaX all exhibit excellent scalability beyond 10k-100k cores, with ALYA, ParFlow and libNEGF also fully 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 – see tables below. ALYA is one of two CFD codes taken up in the EuroHPC hardware benchmarking suite.

System		PizDaint	JUWELS Cluster	JUWELS Booster	Marconi 100	Mare Nostrum4	SuperMUC NG	Joliot-Curie KNL	Joliot-Curie Rome	Hawk
Node Architecture [a]	CPU	INT Xeon	INT Xeon	AMD EPYC	IBM Power9	INT Xeon E5	INT Xeon Skylake	INT KNL	AMD EPYC	AMD EPYC
	GPU	NVD P100	NVD V100	NVD A100	NVD V100	N/A	N/A	N/A	N/A	NVD A100
Location		Lugano, Switzerland	Jülich, Germany	Jülich, Germany	Bologna, Italy	Barcelona, Spain	Garching, Germany	Paris, France	Paris, France	Stuttgart, Germany
Peak performance (PF)		27	12	71	32	11	27	2	12	26
EoCoE Code/Library										
ALYA		++	++	+	+	++	++	++	++	++
ESIAS			++	-		+	+	+	+	++
EURAD-IM			++							
libNEGF				++						
KMC/DMC				++						
ParFlow			++	++						

SHEMAT		+							
Gysela				++		+	++	++	
AMG4PLBAS	++								
…									
PDI	+	++	+	++	+	+	+	+	+
FTI	+	+	+	+	++	+	+	+	+
Melissa-DA		++			++				

Table 1: Compatibility of EoCoE Applications with PRACE Tier-0 systems (++ actual benchmarks obtained, + architecture supported, 0 untested architecture, - architecture unsupported)

[a]Node Architecture: AMD - AMD, INT - Intel, NVD - NVIDIA, N/A - not applicable. Overview of PRACE Tier-0 systems: [15]

		EuroHPC Pre-exascale systems			EuroHPC Peta-scale systems					Other Peta/pre-Exascale systems		
System		LUMI	Leonardo	MareNostrum5	Deucalion	Vega	Discoverer	Karolina	MeluXina	FUGAKU	SUMMIT	CEA-HF
Node Architecture[b]	CPU	AMD EPYC	INT Xeon Ice Lake	NVD Grace	Fujitsu A64FX	AMD EPYC	AMD EPYC	AMD EPYC	AMD EPYC	ARM A64FX	IBM Power9	AMD EPYC Milan
	GPU	AMD Instinct	NVD A100	NVD	N/A	NVD A100	N/A	NVD A100	NVD A100	N/A	NVD GV100	NVD A100
Location		Kajaani, Finland	Bologna, Italy	Barcelona, Spain	Minho, Portugal	Maribor, Slovenia	Sofia, Bulgaria	Ostrava, Slovakia		Riken (Kobe), Japan	ORNL, USA	Bruyères-le-Châtel, France
Peak performance (PFlops)		550	250	314	10	10	6	15		488	200	8.8
EoCoE Codes/libaries												
ALYA				+						+	+	
ParFLOW												
Gysela										++		++
WalBerla												++
Melissa-DA										++		

Table 2: Compatibility of EoCoE Applications and Libraries on EuroHPC and other pre-Exascale systems (++ actual results obtained, + architecture supported, 0 untested architecture, - architecture unsupported)