ACROSS

HPC Big DAta ArtifiCial intelligence cross Stack PlatfoRm TOwardS ExaScale

D2.2 – Description of key technologies and platform design

Deliverable ID	D2.2
Deliverable Title	Description of key technologies and platform design
Work Package	WP2
Dissemination Level	PUBLIC
Version	0.8
Date	2021 - 11 - 30
Status	Submitted
Deliverable Leader	ATOS
Main Contributors	IT4I, LINKS, AVIO, ECMWF, SINTEF, CINECA, CINI, MORFO

Disclaimer: All information provided reflects the status of the ACROSS project at the time of writing and may be subject to change. This document reflects only the ACROSS partners' view and the European Commission is not responsible for any use that may be made of the information it contains.

The ACROSS project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955648. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Italy, France, Czech Republic, United Kingdom, Greece, Netherlands, Germany, Norway.

Published by the ACROSS Consortium

Document History

Version	Date	Author(s)	Description
0.1	2021-10-07	ATOS	Initial Draft
0.2	2021-11-08	LINKS	Contribution to the Sections 1.4 and 5
0.3	2021-11-23	ATOS & all	Include and complete missing parts of the document (pilots requirements section2, orchestration platform section 6, HW technologies section 4, conclusion), reorganize sections
0.4	2021-11-26	LINKS	In review
0.5	2021-11-26	LINKS	Updated in-review version
0.6	2021-11-29	CINECA	Review submitted
0.7	2021-11-29	LINKS/ATOS	Pre-submission version
0.8	2021-11-30	LINKS/ATOS	Submission

Table of Contents

Glossary

Acronym	Explanation
AI	Artificial Intelligence
ANN	Artificial Neural Network
BPMN	Business Model Model and Notation
CFD	Computational Fluid Dynamics
CLI	Command Line Interface
DAG	Directed Acyclic Graph
DL	Deep Learning
DRAM	Dynamic Random Access Memory
ERT	Ensemble based Reservoir Tool
FDB	Fields Database
FMLE	Fast Machine Learning Engine
FPGA	Field-Programmable Gate Array
GPU	Graphics Processing Unit
GUI	Graphical User Interface
HEAppE	High-End Application Execution Middleware
HPC	High Performance Computing
HPDA	High Performance Data Analytic
HW	Hardware
NVM	Non-Volatile Memory
PBS	Portable Batch System
SLURM	Slurm workload manager
SNN	Spiking Neural Network
UDF	User Defined Functions
LES	Large Eddy Simulation
ML	Machine Learning
MPI	Message Passing Interface
NNP	Neural Network Processor
SW	Software
TOSCA	Topology and Orchestration Specification for Cloud Applications

UDF	User Defined Function
VM	Virtual Machine
WARP	Workflow-aware Advanced Resource Planner
WMS	Workflow Management System
YORC	YSTIA Orchestrator

List of figures

List of tables

Executive Summary

This deliverable presents the ACROSS key technologies and first definition of the ACROSS platform that will be built around the infrastructural capabilities offered by supercomputing centers (i.e., IT4I, CINECA, and ATOS experimental research infrastructure). The co-design activities (foreseen in WP2) will ensure the ACROSS platform to be ready for executing exascale level workflows, by being able to optimally orchestrate high-level tasks, job scheduling, and workload distributing among the most appropriate computing elements.
To enhance orchestration and scheduling policies so that infrastructural resource heterogeneity will be effectively exploited by cross-stack workflows, ACROSS will make effective and efficient use of the platform by developing and integrating advanced resource management policies and by leveraging innovative monitoring mechanisms of the platform behavior, aimed at improving overall energy efficiency. Moreover, ACROSS will exploit the availability of specialized hardware ecosystem, with focus on reconfigurable devices.

Objectives of the deliverable:

Deliverable 2.2 is related to Milestone 2 (ACROSS Key technologies and platforms specifications, M9), with the objective to define the overall ACROSS architecture co-design with technologies, platform and software choices. It directly derives from deliverable D2.1 “Summary of pilots co-design requirements”, which provides technical requirements and constraints expressed by pilots. This deliverable will serve as a baseline for the Milestone 3 (ALPHA version: ACROSS platform and technologies) and Milestone 4 (The FIRST pilots use cases integration).
The main objectives of the deliverable D2.2 are:

- Identify the technologies and frameworks that will be part of the ACROSS assets.
  - Mapping matrix.
  - Identify the technology choices for each pilot.
- Define the platform architecture that involve these technologies.
- Describe the identified key hardware technologies.
- Describe the identified key software technologies.
- Describe the orchestrator architecture

Position of the deliverable in the whole project context:

Deliverable D2.2 is linked to WP2 “Cross stack convergence & co-design for HPC and Data driven HPDA software environment” dedicated to the co-design, key technology and platform identification, infrastructure set-up and integration and lesson learned. The D2.2 cover the activities related to key technologies and platform identification.

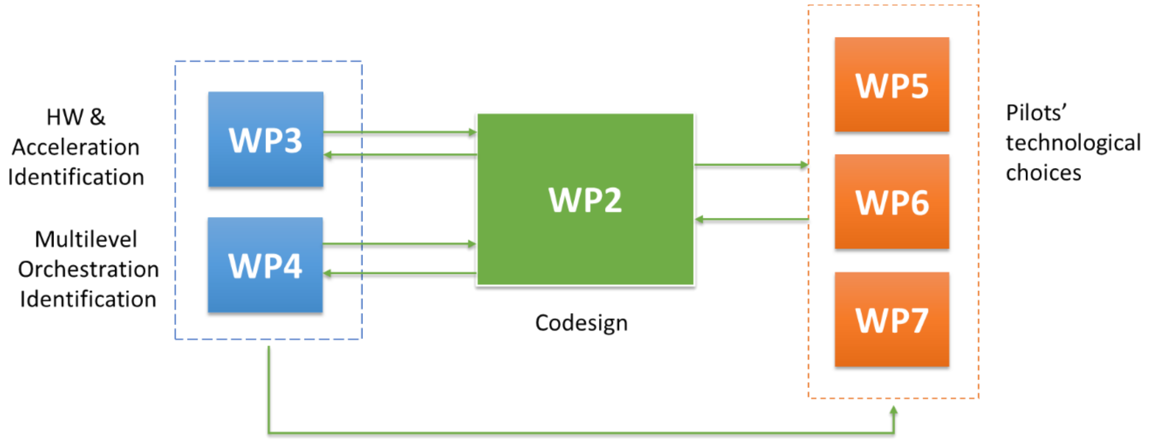


Figure 1 WP2 position in ACROSS project

Figure 1 represents how WP2 activities related to the Key Technologies and ACROSS platform identification are related to the other WPs. It collects the main hardware and software technology choices (based on the baseline and improved requirements) from Pilot WPs (WP5, WP6, and WP7), multilevel orchestration key software technologies from WP4, and key hardware and acceleration technologies from WP3.

Description of the deliverable

This deliverable describes the hardware and software technologies that will compose the ACROSS platform. These technologies have been identified according to the pilots use cases requirements, as specified in D2.1 and improvement plan documents, in order to enable the optimization of their related workflows. Such identified technologies are categorized in hardware and software platforms.

Firstly is described the methodology of work used to achieve the deliverable goals, then is provided a mapping of pilots requirements to identified technologies (section 2), i.e., for each pilots use cases, the way workflows will be optimized and the required technologies are described.

The global ACROSS platform architecture is described in section 3. Then, the document establishes a list of hardware (section 4) and software (section 5) technologies that have been identified. Finally, the ACROSS software stack related to workflows orchestration is depicted in section 6. An appendix section is also added to the end of the document to provide a common set of terms used in the context of application workflows description and their orchestration within the ACROSS platform.

1 Introduction

The identification of the ACROSS key technologies and platform definition is related to pilots requirements and the cross-stack convergence and co-design.

1.1 Scope

The scope of D2.2 is to define the initial ACROSS platform software and hardware architecture, based on software and technologies choices, which have been made according to pilots co-design requirements established in D2.1.
A set of software and hardware technologies has been identified at the beginning of the project. These technologies have been selected according to the pilots' requirements and constraints, and mapped to the different pilots use cases.
These technologies have then been assembled to define the ACROSS platform, resulting in both a software architecture (as defined in WP4) and a hardware/software architecture (as defined in WP3).

1.2 Related documents

ID	Title	Reference	Version	Date
D2.1	Summary of pilots co-design requirements		3.0	2021-08-31
D7.1	Stage 1 requirements for HW/SW integration		2.0	2021-08-31

1.3 Methodology of work

In the scope of the Milestone 2, the WP2 activities were focused on the definition of key technologies related to platform's design and to ensure that the pilot's requirements acquired and consolidated in MS 1 are covered by the identified technologies and that the ACROSS platform fulfils the overall technical goals of the project and helps pilots to achieve their objectives and related KPIs.
The same co-design approach that was defined for the Milestone 1 was followed during the Milestone 2 activities. It includes the regular WP2 telcos with key project partners to discuss, present and plan WP2 tasks and the creation of the documents related to the co-design activities: 'Mapping matrix' document and 'Improvement plan and requirements' document.

Mapping matrix

Mapping matrix focuses on the clear overview of the identified SW, HW, Tools, and technologies and maps them to the actual pilot's use-case requirements. This document is a result of a numerous convergence interactions between technical WPs and pilot WPs in relation to the updated pilot's workflows and overall pilot's use-cases improvements. For details, see Section 2.

Improvement plan and requirements

Improvement plans document contains all the envisioned/foreseen improvements with regards to the baseline pilot workflows definition as reported in the questionnaires, along with their improved technological requirements, towards the full achievement of the objectives and KPIs defined in the DoA of the proposal. To this end, the document aims at selecting pilot workflow elements that can move beyond the baseline, in terms of expected objectives and KPIs, by refining and improving requirements at the hardware and software level.

Document template:
Pilot improvement ID #1

- Baseline
    - Reference to the baseline workflow item to be improved (e.g., LES CFD, workflow execution, accelerating a ML/DL model on NNPs, etc.)
- Technological domain
    - Improvement field of competence (e.g. data movement, code acceleration, workflow orchestration, etc.) and related tech WPs
- Improvement Rationale
    - Why is the improvement needed with respect to the baseline workflow?
- Description
    - List of functional description of the proposed improvements
- Expected results & KPIs
    - Expected (quantitative where possible) improvement over baseline and definition of relative KPI (e.g. Time to Solution speedup in percentage, etc.)
- Improved workflow BPMN
    - Description of the new workflow's scheme, if the improvement will determine a modification of the baseline scheme. In this case, please highlight which parts are changed or which parts imply a change in the non-functional HW/SW requirements.
- Technical requirements: Hardware
    - List of improvement's hardware requirements (e.g. FPGA, NNP, etc.) and description (if needed)
- Technical requirements: Software & Orchestration
    - List of improvement's software requirements (e.g. HyperTools, TensorFlow, etc.) and description (if needed)
- Feasibility degree
    - How much effort and time will the improvement's implementation require by the partners?
    - Please provides a feasibility category for the improvement (committed, hard, easy) followed by some additional observations/notes.
- Quality of end product
    - Standardization/Portability
    - Competitivity (vs SoA)
    - Evolution towards new computing technologies

1.4 Mapping with ACROSS overall objectives

The key technologies identification and platform design linked to Milestone 2 (M9) was focused to cover in part the ACROSS Overall Platform Objectives O1.1, (see Figure 2), the pilot-specific objectives O2.1, O2.2, and O2.3, and the ACROSS technology-specific objectives O3.1, O3.2.

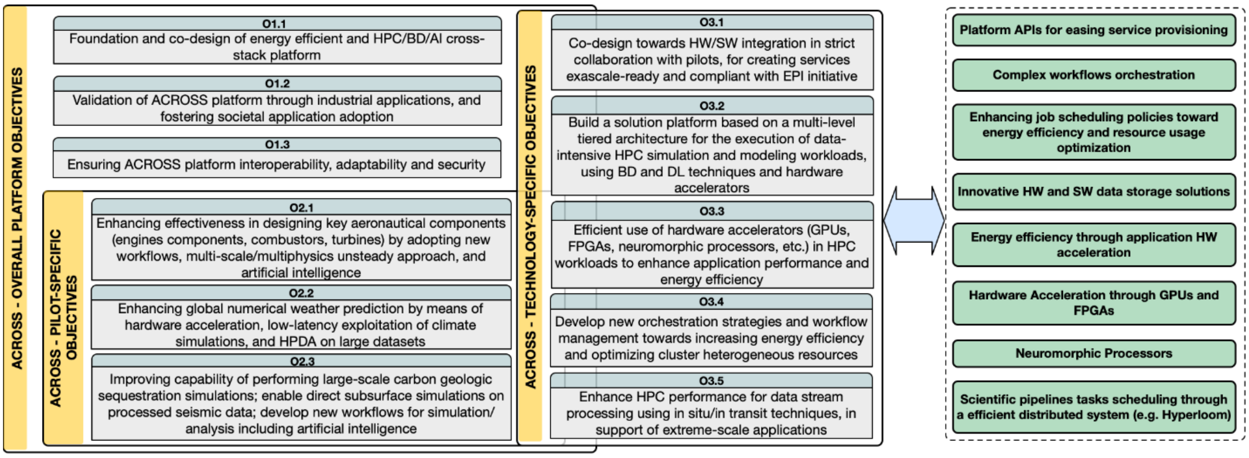
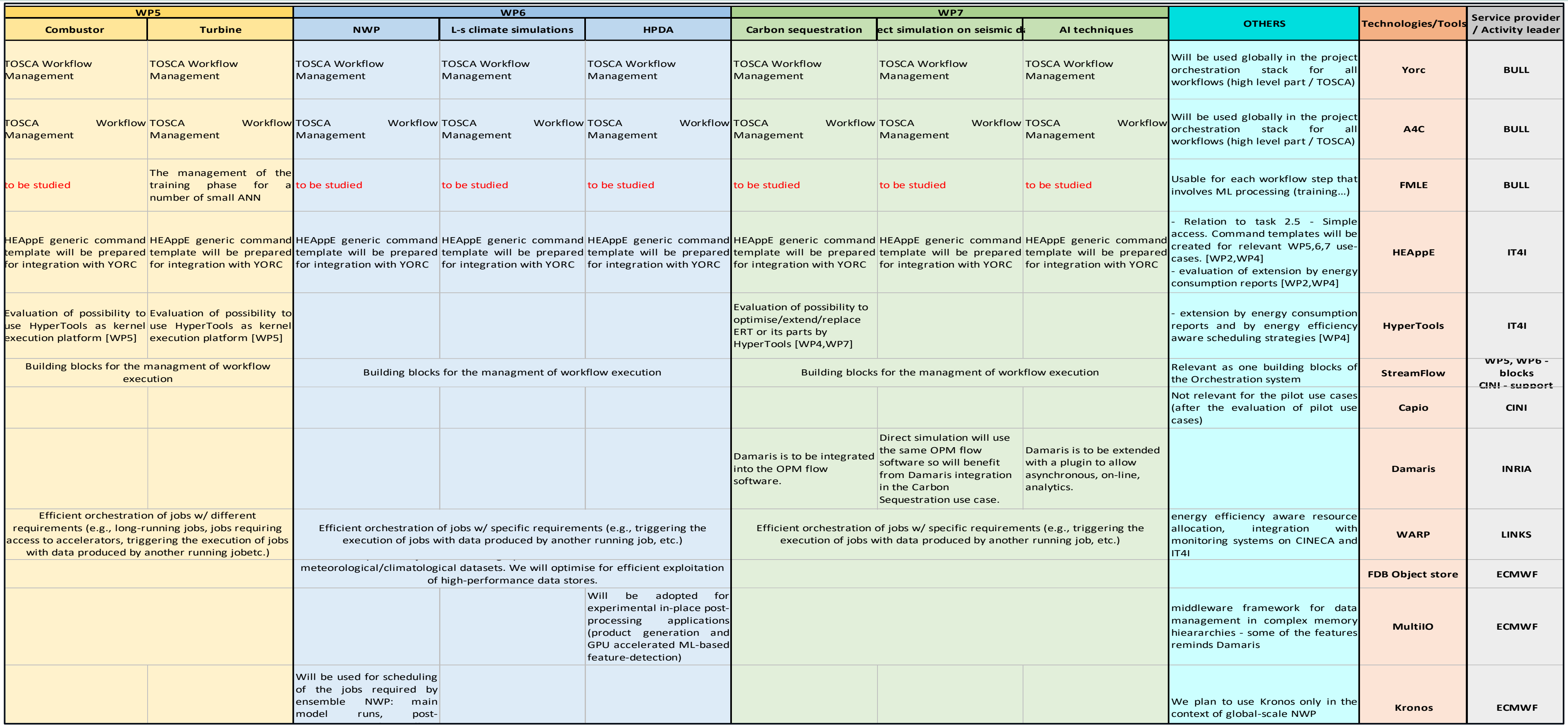

ACROSS - OVERALL PLATFORM OBJECTIVES
O1.1
Foundation and co-design of energy efficient and HPC/BD/AI cross-stack platform
O1.2
Validation of ACROSS platform through industrial applications, and fostering societal application adoption
O1.3
Ensuring ACROSS platform interoperability, adaptability and security
ACROSS - PILOT-SPECIFIC OBJECTIVES
O2.1
Enhancing effectiveness in designing key aeronautical components (engines components, combustors, turbines) by adopting new workflows, multi-scale/multiphysics unsteady approach, and artificial intelligence
O2.2
Enhancing global numerical weather prediction by means of hardware acceleration, low-latency exploitation of climate simulations, and HPDA on large datasets
O2.3
Improving capability of performing large-scale carbon geologic sequestration simulations; enable direct subsurface simulations on processed seismic data; develop new workflows for simulation/ analysis including artificial intelligence
ACROSS - TECHNOLOGY-SPECIFIC OBJECTIVES
O3.1
Co-design towards HW/SW integration in strict collaboration with pilots, for creating services exascale-ready and compliant with EPI initiative
O3.2
Build a solution platform based on a multi-level tiered architecture for the execution of data-intensive HPC simulation and modeling workloads, using BD and DL techniques and hardware accelerators
O3.3
Efficient use of hardware accelerators (GPUs, FPGAs, neuromorphic processors, etc.) in HPC workloads to enhance application performance and energy efficiency
O3.4
Develop new orchestration strategies and workflow management towards increasing energy efficiency and optimizing cluster heterogeneous resources
O3.5
Enhance HPC performance for data stream processing using in situ/in transit techniques, in support of extreme-scale applications
Platform APIs for easing service provisioning
Complex workflows orchestration
Enhancing job scheduling policies toward energy efficiency and resource usage optimization
Innovative HW and SW data storage solutions
Energy efficiency through application HW acceleration
Hardware Acceleration through GPUs and FPGAs
Neuromorphic Processors
Scientific pipelines tasks scheduling through a efficient distributed system (e.g. Hyperloom)

Figure 2 ACROSS Objectives

2 Mapping matrix



WP5		WP6			WP7			OTHERS	Technologies/Tools	Service provider / Activity leader
Combustor	Turbine	NWP	L-s climate simulations	HPDA	Carbon sequestration	ect simulation on seismic d	AI techniques			
TOSCA Workflow Management	TOSCA Workflow Management	TOSCA Workflow Management	TOSCA Workflow Management	TOSCA Workflow Management	TOSCA Workflow Management	TOSCA Workflow Management	TOSCA Workflow Management	Will be used globally in the project orchestration stack for all workflows (high level part / TOSCA)	Yorc	BULL
TOSCA Workflow Management	TOSCA Workflow Management	TOSCA Workflow Management	TOSCA Workflow Management	TOSCA Workflow Management	TOSCA Workflow Management	TOSCA Workflow Management	TOSCA Workflow Management	Will be used globally in the project orchestration stack for all workflows (high level part / TOSCA)	A4C	BULL
to be studied	The management of the training phase for a number of small ANN	to be studied	to be studied	to be studied	to be studied	to be studied	to be studied	Usable for each workflow step that involves ML processing (training...)	FMLE	BULL
HEAppE generic command template will be prepared for integration with YORC	HEAppE generic command template will be prepared for integration with YORC	HEAppE generic command template will be prepared for integration with YORC	HEAppE generic command template will be prepared for integration with YORC	HEAppE generic command template will be prepared for integration with YORC	HEAppE generic command template will be prepared for integration with YORC	HEAppE generic command template will be prepared for integration with YORC	HEAppE generic command template will be prepared for integration with YORC	- Relation to task 2.5 - Simple access. Command templates will be created for relevant WP5,6,7 use-cases. [WP2,WP4] - evaluation of extension by energy consumption reports [WP2,WP4]	HEAppE	IT4I
Evaluation of possibility to use HyperTools as kernel execution platform [WP5]	Evaluation of possibility to use HyperTools as kernel execution platform [WP5]				Evaluation of possibility to optimise/extend/replace ERT or its parts by HyperTools [WP4,WP7]			- extension by energy consumption reports and by energy efficiency aware scheduling strategies [WP4]	HyperTools	IT4I
Building blocks for the managment of workflow execution		Building blocks for the managment of workflow execution			Building blocks for the managment of workflow execution			Relevant as one building blocks of the Orchestration system	StreamFlow	WP5, WP6 - blocks CINI - support
								Not relevant for the pilot use cases (after the evaluation of pilot use cases)	Capio	CINI
					Damaris is to be integrated into the OPM flow software.	Direct simulation will use the same OPM flow software so will benefit from Damaris integration in the Carbon Sequestration use case.	Damaris is to be extended with a plugin to allow asynchronous, on-line, analytics.		Damaris	INRIA
Efficient orchestration of jobs w/ different requirements (e.g., long-running jobs, jobs requiring access to accelerators, triggering the execution of jobs with data produced by another running jobetc.)		Efficient orchestration of jobs w/ specific requirements (e.g., triggering the execution of jobs with data produced by another running job, etc.)			Efficient orchestration of jobs w/ specific requirements (e.g., triggering the execution of jobs with data produced by another running job, etc.)			energy efficiency aware resource allocation, integration with monitoring systems on CINECA and IT4I	WARP	LINKS
		meteorological/climatological datasets. We will optimise for efficient exploitation of high-performance data stores.							FDB Object store	ECMWF
				Will be adopted for experimental in-place post-processing applications (product generation and GPU accelerated ML-based feature-detection)				middleware framework for data management in complex memory hieararchies - some of the features reminds Damaris	MultiIO	ECMWF
		Will be used for scheduling of the jobs required by ensemble NWP: main model runs, post-						We plan to use Kronos only in the context of global-scale NWP	Kronos	ECMWF

Figure 3 Mapping matrix part 1

				We will adopt containers for WFLOW deployment					Containers (Cloud)	WP6 - containers IT4I, CINECA - support
				We will adopt containers for WFLOW deployment and we will consider them for WRF ancillary tasks	Singularity containers are an option for distribution to be tested for efficiency/scalability	Singularity containers are an option for distribution to be tested for efficiency/scalability	Singularity containers are an option for distribution to be tested for efficiency/scalability	Containers are seen as a convenient way of managing the deployment of workflow execution environments (HPC side) in the context of the Orchestration system	Containers (HPC)	WP6, WP7 - containers IT4I, CINECA - support
								VMs are seen as a convenient way of managing the deployment of workflow execution environments (Cloud side) in the context of the Orchestration system	VMs for orchestrator deployment	LINKS
	Development of ANNs for turbine design						Development of ANN Models	Support of DL-based Applications	AI Framework	BULL
	Support of deployment and optimisation for AI			We plan to use GPUs for HPDA tasks (feature detections on global-scale NWP outputs)	Optimization of GPU usage in linear solver of OPM Flow, exploring CUDA implementation [W3,WP7]		Damaris AI/Analytics plugin could take advantage of available GPUs		Porting code to GPU	IT4I
										BULL
Under evaluation	Under evaluation	Under evaluation	Under evaluation	Under evaluation	Possible use of FPGAs will be re-evaluated before mid-project.	Under evaluation	Damaris AI/Analytics plugin could take advantage of available GPUs		Porting code to FPGA	LINKS
								Provide FPGA Hw/Sw Infrastructure. Emulation of NN Processing	FPGA	BULL
	Acceleration of ANN Training/Inference						Acceleration of ANN Training/Inference	Support of DL Acceleration	AI-Acceleration	BULL
		We plan to use, whenever possible, a specific FDB engine optimized for storage-class memory (i.e. Intel Optane DC available on Galileo100)							Storage Class Memory	Cineca
		NVM technologies adoption							NVM	Cineca
					Can be used for high performance data migration	Can be used for high performance data migration			Burst Buffer	BULL
								- providing energy efficiency monitoring for partners [WP2,WP4] - extension HyperTools & HEAppE by worfklow energy consumption monitoring [WP2,WP4]	HDEEM + Meric	IT4I
								This tool provides several metrics monitoring HPC Clusters status (power consumption, nodes availability, network availability....)	ExaMon	CINECA
								Task 2.3 Compliance to EPI guideline and exascale perspective. EPI seminar/workshop will be held on November 10th.	EPI	BULL, LINKS

Figure 4 Mapping matrix part 2

Based on the inputs provided into the mapping matrix document the following SW & HW technologies and tools were identified. The following list is divided into four categories. Global category contains the technologies that will be adopted globally i.e., usable across all pilot's use-cases. The remaining three categories are separated into respective pilot WPs. The following list serves as a high-level overview of the identified key technologies and respective activity leader/technology provider. A more detailed description of these technologies and their envisioned utilization is part of the respective chapters describing the SW (Sec. 3), HW (Sec. 4), Orchestration (Sec. 5) and pilot's improved workflows (Sections 2.1, 2.2, 2.3).

- Global
  - Yorc, A4C [ATOS]
  - HEAppE [IT4I]
  - VMs for orchestrator deployment [LINKS]
  - HDEEM+Meric [IT4I]
  - ExaMon [CINECA]
  - StreamFlow [CINI]
  - WARP [LINKS]
  -

- WP5
  - FMLE [ATOS]
  - HyperTools [IT4I]
  - AI Framework & AI-Acceleration [ATOS]

- WP6
  - FDB Object Store [ECMWF]
  - MultiIO [ECMWF]
  - Kronos [ECMWF]
  - HPC & Cloud containers [WP6, IT4I, CINECA]
  - GPU [IT4I, ATOS]
  - Storage Class Memory & NVMs [CINECA]

- WP7
  - HyperTools [IT4I]
  - Damaris [INRIA]
  - HPC containers [WP7, IT4I, CINECA]
  - AI Framework & AI-Acceleration [ATOS]
  - GPU [IT4I, ATOS]
  - FPGA [LINKS]
  - Burst Buffer [ATOS]

2.1 Greener aero-engine modules optimization Pilot

2.1.1 Combustor use case

The UTHERM3D tool allows to accurately predicting the temperatures of the metallic components inside the aeronautical combustors solving the heat transfer problem by means of loosely coupled, multi-physics and multi-scale approaches, which allow an optimal modeling for each physical phenomenon affecting the heat transfer. The TECFLAM swirl burner jointly developed by the Darmstadt, Heidelberg, Karlsruhe and the DLR institutes was identified as the first test case for the first part of the project.

During the M1-M9 period these months, the computational infrastructure has been set up to perform the simulation with UTHERM3D procedure on IT4I clusters. In addition, scalability tests have been carried out on ANSYS Fluent code, with which the UTHERM3D tool is developed. The tests have been carried out directly with the reactive simulation of the TECFLAM combustor, discretized with a polyhedral mesh of 42 million elements. Figure 5 summarizes the results of the scalability test carried out on the Barbora cluster[1] of IT4I in terms of time per time step.

[1] https://docs.it4i.cz/barbora/introduction/

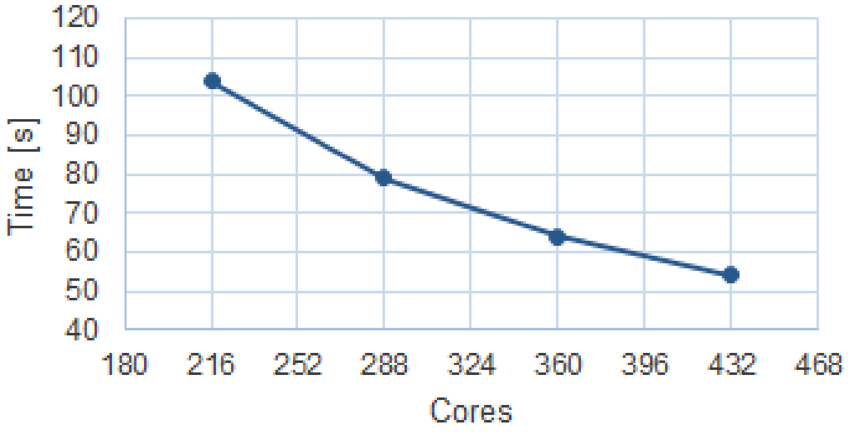


Figure 5 Scalability test on IT4I Barbora cluster

The UTHERM3D simulation that will define the timing and capabilities of the baseline version of the tool is currently underway. This will supply the base to compare the performance of the new procedure that will be developed in the next months (with respect to the time of writing this document) that will bring to the reduction of the times tied up to the simulation.

The reduction in computational time will be achieved by revisiting the management of data exchange between the various Fluent sessions. In particular, the coupling strategy will ideally remain the same but all reading and writing data operations will be replaced with a more efficient approach.

The baseline (application) workflow and its improved version are reported in the Figure 6.

The new data management will be carried out by a dedicated application-programming interface, developed by ANSYS, the System Coupling software. This new approach will simplify the setup procedure of the different simulations within the UTHERM3D procedure (see Figure 7) and, at the same time, will reduce the calculation time because it will eliminate all the operations of writing and reading of the data to be exchanged.
The data constituting the boundary conditions of the simulations will be allocated in special partitions of the memory and updated by the System Coupling software through UDF (User Defined Functions) that constitute the native language of ANSYS Fluent. In order to fully exploit the capabilities of the System Coupling software, the ANSYS version will be updated from 19.3 to 21.2.

The first part of the activity was conducted on the IT4I Barbora cluster; however, to better exploit the performance in terms of scalability of ANSYS Fluent code and to have a further saving in terms of computation time, tests on an increasing number of computing resources will be conducted on the new IT4I Karolina cluster. If the tests will lead to tangible benefits (i.e., better scalability, shorter computational time), this new computing cluster will be used over the remainder of the ACROSS project. The technologies, computational resources used and upcoming activities are summarized in Table 1.

Regarding the software requirements, as already described in D2.1, the ANSYS Fluent solver for highly nonlinear problems does not have great advantages in working with GPUs, for this reason during the whole project CPUs will be used. For the enhanced UTHERM3D procedure, the only requirement is version 21.2 of the ANSYS suite, which is being tested in this period on IT4Innovations Karolina[2] cluster. This release will allow us to be able to use the most up-to-date version of the ANSYS System Coupling software.

[2] https://www.it4i.cz/en/infrastructure/karolina

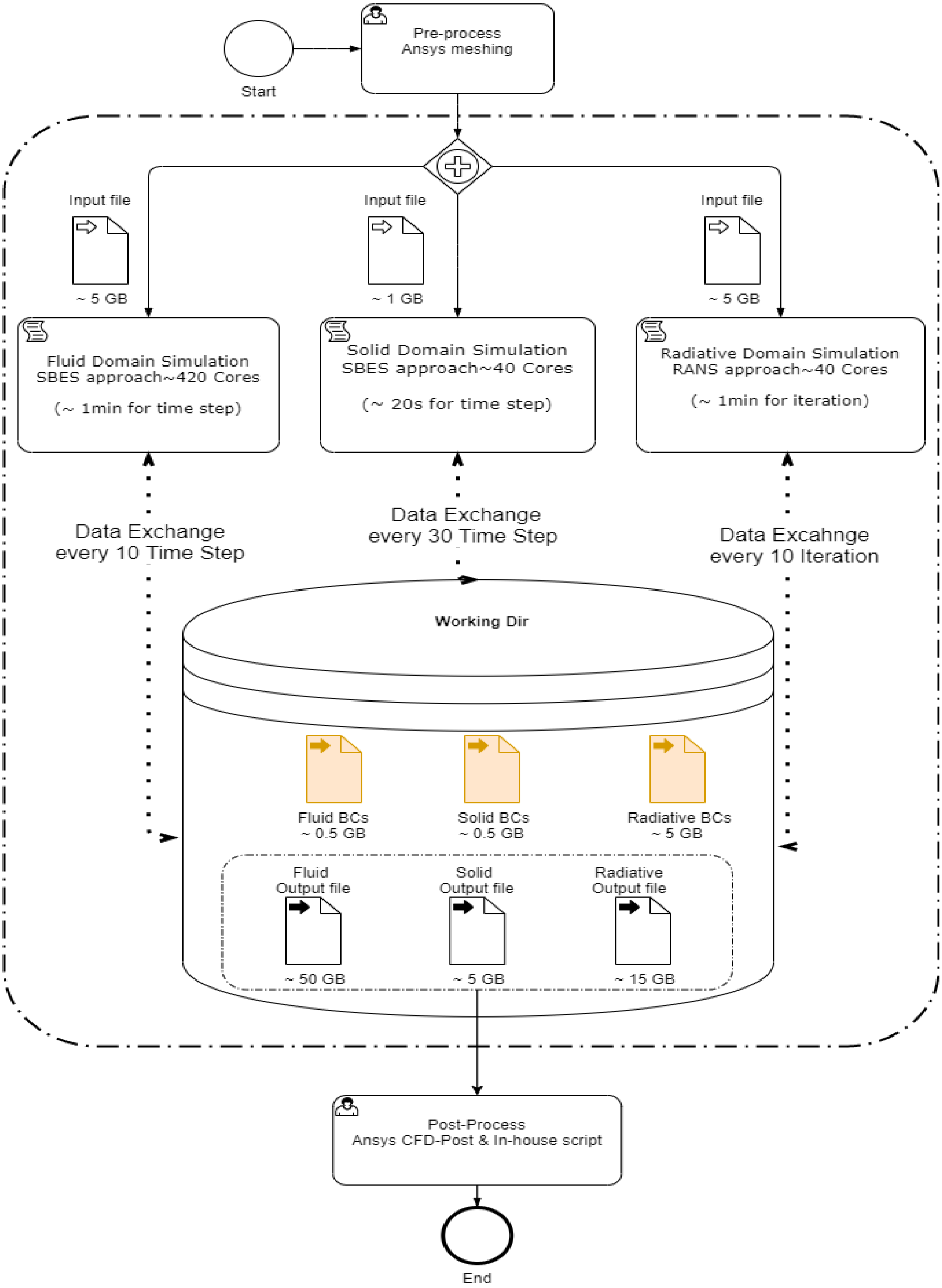


Figure 6 UTHERM3D baseline workflow

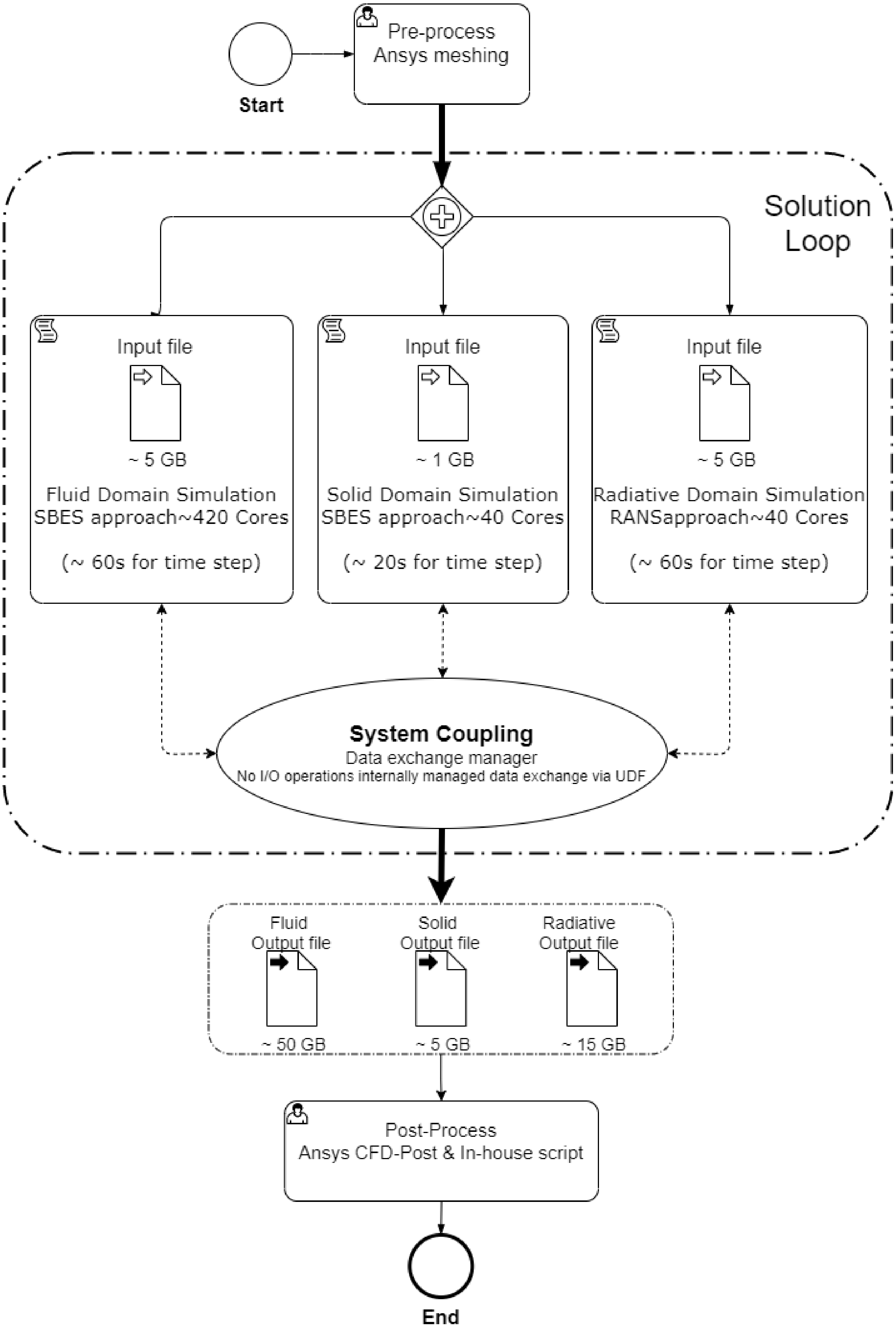


Figure 7 UTHERM3D improved workflow

Performance improvement	Technologies and computational resources used, actions
time to solution	*CFD (CPUs only):* • Scalability test on Barbora cluster for the ANSYS Fluent code performance • Set up the environment (use of external licenses) to use Karolina cluster • New scalability tests will be performed on Karolina cluster

Table 1 Technologies/computational resources and performance improvement for the WP5 combustor pilot

Considering the first outcomes covering the M1-M9 period of the ACROSS project, Table 2 shows the updated KPIs. The main objectives set by the industrial team GE Avio are the improvement of the productivity target (i.e., the reduction of U-THERM3D simulation time with respect to the state-of-the-art by at least 30%) acting on both modeling aspects of the physical problem and the improvement brought by (new) hardware resources. The other aspect is to improve the quality of the numerical results, getting as close as possible to the experimental field results; in particular, the objective is to center the experimental temperature measurement with a margin of ±30K.This can be done by acting on the modeling aspects used within the simulation. As a final statement, expected results will be aimed at improving the quality of the numerical prediction with a reduction of the calculation time. From this point of view, the objective is to achieve the same computing times as simulating only the fluid domain with adiabatic walls (theoretical limit that can be achieved by eliminating the waiting time for boundary conditions updating) but with the quality generated by a multi-scale/multi-physics approach as U-THERM3D.

WP5 Combustor revised KPIs	
KPI - 2.1	Productivity target (time-to-design reduction with regards to current situation) for both aeronautical test cases. At least **30%**, acting on both the modeling aspects and the improvement brought by (new) hardware resources
KPI - 2.2	Combustor metal temperature prediction with regards to experiments. Target reduction of uncertainty margin to ±30K by acting on the modeling aspects used within the simulation.

Table 2 Revised version of combustor KPIs

2.1.2 Turbine use case

A detailed description of the application workflow for the turbine pilot use case has been already reported in the Deliverable D2.1. After introducing the main objectives of the pilot use case, the main KPIs have been reported in the same deliverable, concerning the time reduction for the preliminary design phase of a LPT module, and an expected benefit in terms of improved performance for the module itself.

As previously underlined in the Deliverable D2.1, a “baseline” workflow has been implemented in the first phase of the project (M1-M9), while pointing out the main area of improvement, mostly concerning the HPDA/LES workflow part and the AI tools.

Focusing on one side on the baseline and on the other side on the improved workflows and tools, the proper technologies and platforms were identified. Looking at the overall workflow (see Figure 8) three main areas can be identified requiring HPC resources: CFD calculations, HPDA analysis, AI tools. Different tools and approaches are adopted in all these parts, thus leading to specific requirements in terms of platforms and technologies.

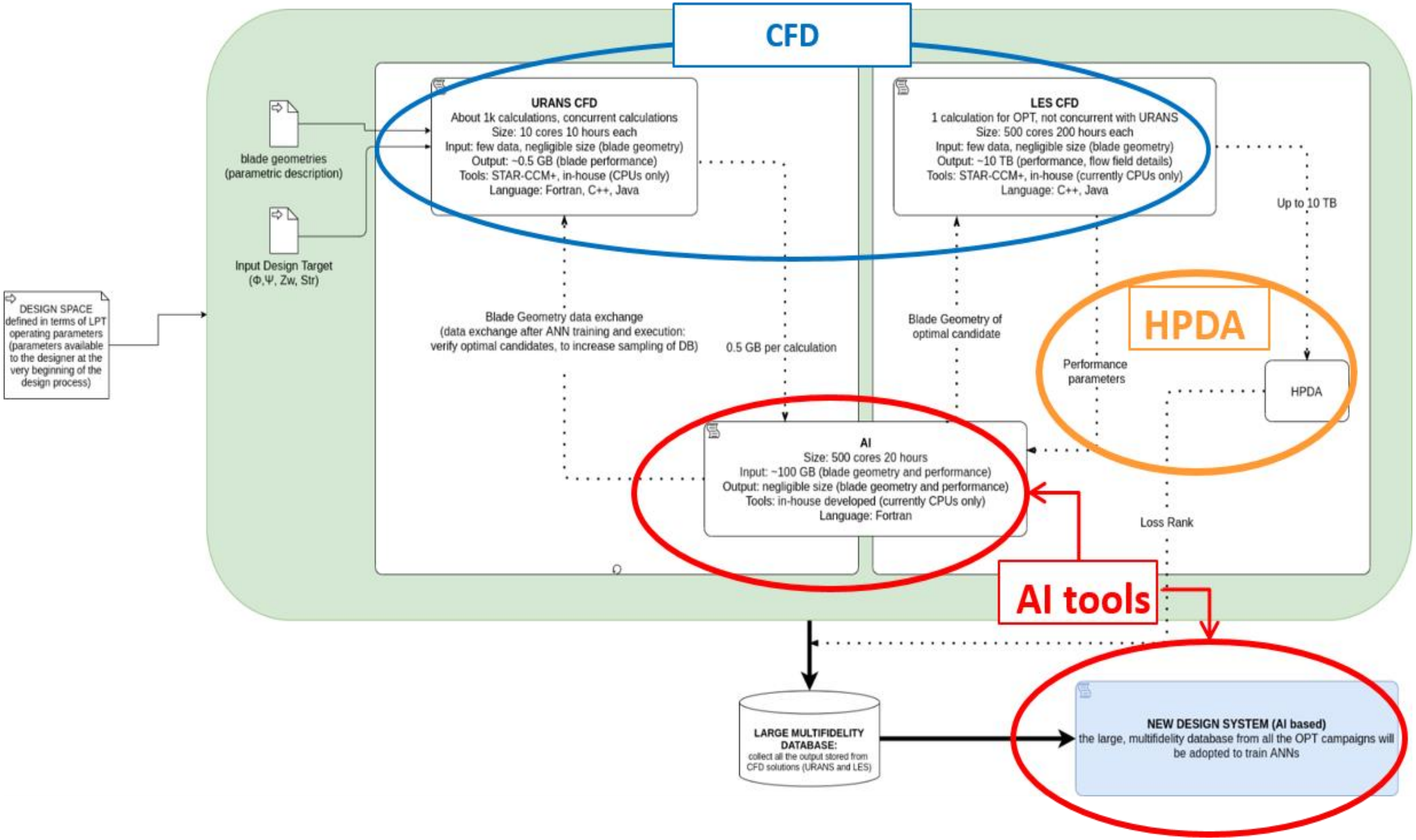

CFD
URANS CFD
About 1k calculations, concurrent calculations
Size: 10 cores 10 hours each
Input: few data, negligible size (blade geometry)
Output: ~0.5 GB (blade performance)
Tools: STAR-CCM+, in-house (CPUs only)
Language: Fortran, C++, Java
LES CFD
1 calculation for OPT, not concurrent with URANS
Size: 500 cores 200 hours each
Input: few data, negligible size (blade geometry)
Output: ~10 TB (performance, flow field details)
Tools: STAR-CCM+, in-house (currently CPUs only)
Language: C++, Java
blade geometries
(parametric description)
Input Design Target
(Φ,Ψ, Zw, Str)
DESIGN SPACE
defined in terms of LPT operating parameters (parameters available to the designer at the very beginning of the design process)
Up to 10 TB
Blade Geometry data exchange
(data exchange after ANN training and execution:
verify optimal candidates, to increase sampling of DB)
0.5 GB per calculation
Blade Geometry of
optimal candidate
HPDA
Performance
parameters
HPDA
AI
Size: 500 cores 20 hours
Input: ~100 GB (blade geometry and performance)
Output: negligible size (blade geometry and performance)
Tools: in-house developed (currently CPUs only)
Language: Fortran
Loss Rank
AI tools
LARGE MULTIFIDELITY
DATABASE:
collect all the output stored from
CFD solutions (URANS and LES)
NEW DESIGN SYSTEM (AI based)
the large, multifidelity database from all the OPT campaigns will be adopted to train ANNs

Figure 8 Turbine workflow and main field of interest for platform choice

CFD calculations are carried out with the commercial software STAR-CCM+, and, currently, it runs on CPUs only. As it is a commercial code, there is no possibility to access the source code looking for improvements. On the contrary, it is possible to take care of the guidelines provided by the software house in order to optimize the performance on HPC infrastructure, looking, for example to the tuning of memory bandwidth, MPI libraries, etc.

The new Galileo100[3] (G100) cluster at CINECA has been selected for this task and computational resources (2.4Mhours, 10TB storage) have been allocated from October 2021 to October 2022 through the 6th EU-ICEI call[4]. In order to derive some preliminary information about the code scaling and overall performance, a LES calculation has been adopted to test the performance. This calculation is representative of the ones that will be carried out during the project, with a mesh size of about 2M polyhedral cells (~344 M verts), a memory requirement of 68 GB RAM, and a solution size of ~35 GB.

All the tests are carried out asking for a "full node" reservation, and CPU time is estimated using STARCCM+ monitor: "Elapsed Time per Time Step (s)". Ten time steps are monitored for each setup, and the average time is adopted for performance evaluation.

The main results of this campaign are summarized in Figure 9 where the computational time is reported as a function of the number of CPU cores adopted for the calculation. The time obtained on the CINECA G100 cluster with 280 CPU cores was adopted as a reference, and all the results are shown in terms of relative time ($time/time_{reference}$). These results allow comparing the performance of the previous CINECA Galileo infrastructure and the new G100 one. Moreover, the impact of different MPI environment is also highlighted comparing OpenMPI and IntelMPI. The outcome of the test campaign is as follows:

- Good scalability of STARCCM+ up to more than 500 CPU cores
- A wall clock time reduction of ~30% with the new Galileo100 cluster

Concerning scalability, these results seem to be aligned with the performance claimed by Siemens in the code guidelines and coherent with the fact that the code scales well until a sufficient amount of cells are computed by each CPU core (~70k cells/CPU core in the current test when using 550 CPU cores).

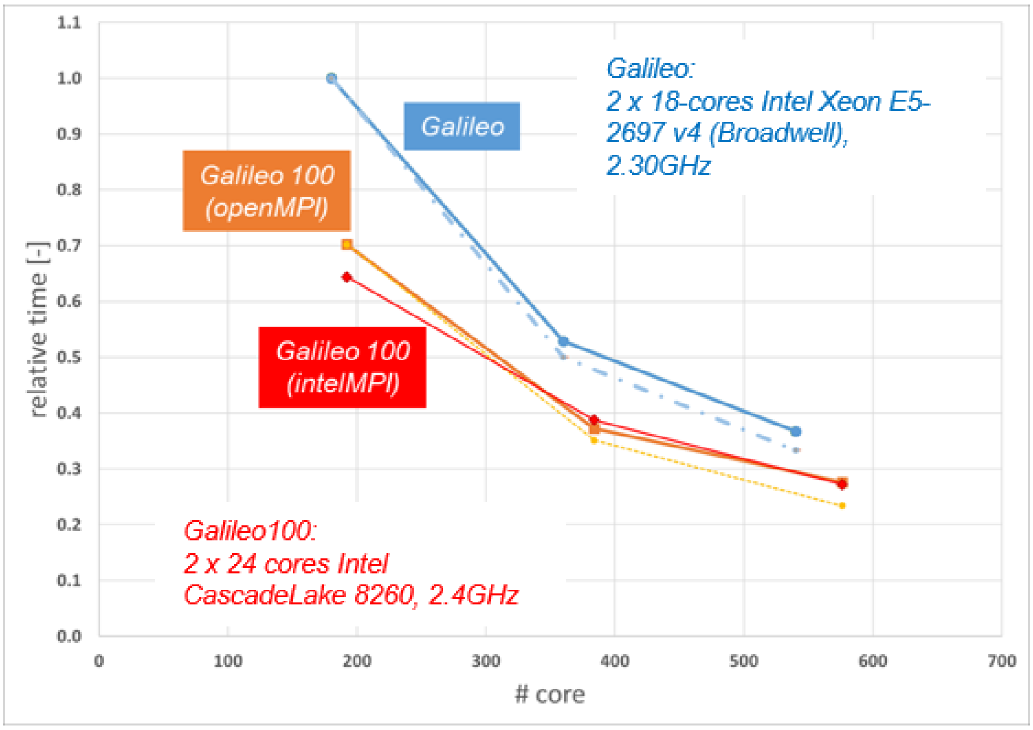


Figure 9 STAR-CCM+ performance

Further tests were also carried out on the G100 cluster by varying the number of nodes. These results are reported in terms of calculation speedup in Figure 10, and they confirm the high-level of parallelism of the code, thus suggesting effective exploitation of the HPC resources.

[3] https://www.hpc.cineca.it/hardware/galileo100

[4] https://prace-ri.eu/hpc-access/collaborative-calls/prace-icei-calls-for-proposals-call-6/

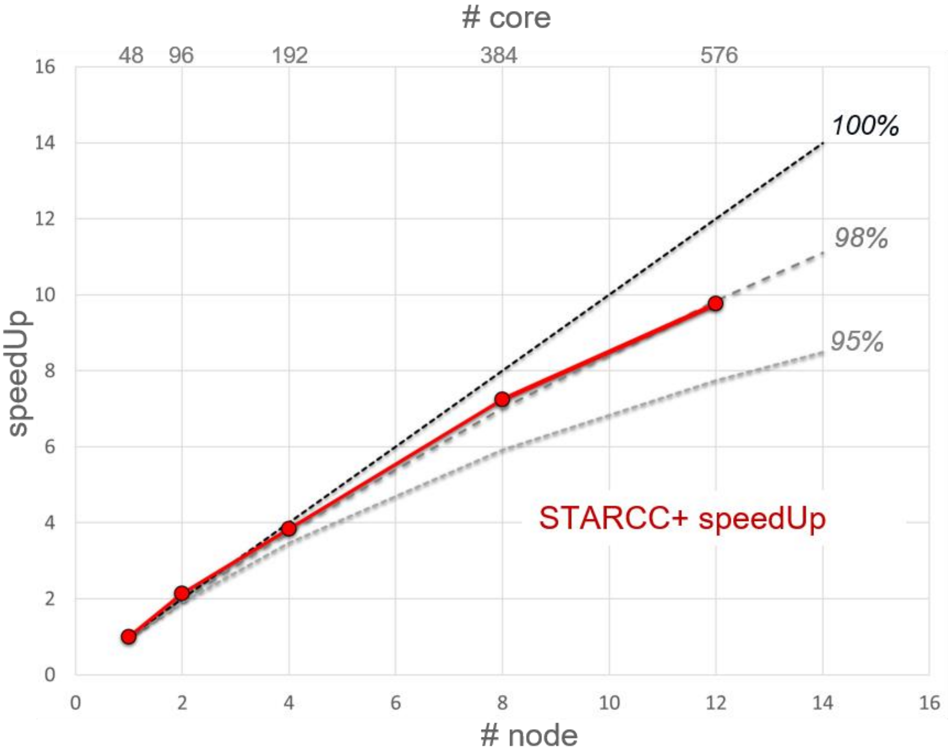


Figure 10 STAR-CCM+ speedup on G100 cluster

The second area of interest concerns AI models. In this case, it is important to distinguish between the current models and the improved ones that will be developed during the project.
Current tools, mainly consist of single objective ANNs: feedforward networks with two hidden layers non-recurrent and fully connected. These tools are Fortran and bash scripts in-house developed by Morfo and can be executed on CPUs only. Only a serial-execution version of the ANN(s) is available and the only form of parallelism consists of a simultaneous running of different ANNs (for different objective functions, or ANNs architectures for hybridization, etc.).
In the improved version, instead, the new ANN models will be able to manage a very large database, multi-fidelity data, and multi-dimensional output. These new tools will be developed in python leveraging Keras/TensorFlow tools.
Concerning computational infrastructures and technologies, the new models will be able to exploit accelerators for reducing both the training and the inference time. More in detail, the new models will be ported on GPUs, which will be mostly used during all the project duration for supporting the training and inference phases of the models. At the same time, the WP3 team will test the performance also adopting FPGA technology for both training and inference. Finally, a reference case will be selected by the WP5, and the trained model will be provided to ATOS and LINKS that will port and test it for testing and benchmarking the inference phase on a neuromorphic system.

In addition, different platforms and technologies will be adopted between the baseline and the improved version of the HPDA tools. Baseline tools have been developed by UNIGE adopting both Matlab and Fortran. In any case, they can be executed on CPUs only and are usually performed a-posteriori on the LES calculation. In ACROSS they will be ported to Python in order to ease their execution on GPUs. The adoption of GPUs acceleration, indeed, is one of the planned improvements for reducing the computational effort of this part of the workflow. As already reported in the deliverable D2.1, the other main improvement concerns the development of a new workflow in which HPDA is performed while LES calculation is still running. HPDA therefore will feature more as a co-processing step rather than a post-processing step, allowing interesting possibilities in terms of time reduction. Focusing on the platforms/technologies choices, an effective orchestration strategy for managing the LES calculation running on several CPUs, and the HPDA analysis running on both CPUs and GPUs, and using filesystems to exchange information.

From a broader perspective, therefore, two main directions of improvements have been identified for this WP5 use case: on the one hand, the introduction of new technologies and/or more effective frameworks/workflow;

on the other hand, the performance improvement of all the adopted tools, by a proper tuning on the HW or by the adoptions of accelerators. After all, the co-design phase has been able to underline the main targets for the models development in accordance with the proper hardware selection and orchestration strategy.
The previous considerations are summarized in the following tables, where technologies and platforms choices have been associated with the corresponding improvement in the workflow (see Table 3 or with an improvement in the performance of the models (see Table 4).

Workflow Improvement	**Technologies and computational resources used, actions**
Orchestration strategy	*CFD/AI* • CFD-URANS calculations driven by AI models. CFD and AI software will leverage different HW architectures (CPUs only for CFD, CPUs/GPUs for AI) but are strictly interconnected in the workflow. *LES/HPDA* • The new workflow described in the Deliverable D2.1 highlight the importance of an effective orchestration strategy between LES and HPDA, that may runs in different HW while exchanging a very large amount of data.
Data handling	LES/HPDA • A large amount of data is provided by the LES calculations to the HPDA tools.
Improved framework	*AI:* • The adoption of a more effective framework (TensorFlow/Keras) will enable the development of improved models able to manage the very large database built during the project, to handle multi-fidelity data, and multi-dimensional output. It will also make easier and effective the adoption of HW accelerators. *HPDA:* • The porting to Python of the baseline models developed by UNIGE adopting Matlab will help in the introduction of GPUs, while increasing the model portability.

Table 3 Technologies/computational resources and workflow improvement for the WP5 turbine pilot

Performance improvement	**Technologies and computational resources used, actions**
time to solution	*CFD (CPUs only):* • Exploitation of G100 cluster tuning (cache, vectorization, NUMA policies, interconnection, etc.) for maximize commercial code (STAR-CCM+) performance
accelerators	*AI:* • Porting on GPUs and adoption of accelerated models for the training/inference phases during the project. • Test on performance with FPGA for both training and inference. • Test on inference performance for a reference case on a neuromorphic system starting from an already trained model. *HPDA:* • Porting on GPUs

Table 4 Technologies/computational resources and performance improvement for the WP5 turbine pilot

WP5 Turbine revised KPIs	
KPI - 2.3	The overall KPI of the WP5 pilot is an expected time reduction (~50%) of the design procedure related to the introduction of the new DS as a whole into the Avio Aero industrial design procedure
KPI - 2.4	A speed-up of at least 20% is expected for the introduction of GPU's for the AI models. Even more marked improvements (speed up > 50%) appear necessary looking to more advanced HW (FPGA, neuromorphic systems, etc.), to justify their introduction in daily usage of the DS
KPI - 2.5	For the improved workflow on LES/HPDA, a time reduction at least of 30% is expected. Moreover, it is important to underline that it will also ensure a more homogeneous database in terms of LES convergence.

Table 5 Revised version of turbine KPIs

2.2 Weather, Climate, Hydrological and Farming Pilot

WP6 application workflows can be seen as complex workflows that can be decomposed in 4 main sub-workflows:

1. Global-scale Numerical Weather Prediction (NWP) and product generation
2. Global-scale climatological simulation
3. Large-scale Hydrological simulation
4. Meso-scale Numerical Weather Prediction (regional down-scaling) and data post-processing.

Improvements planned for WP6 will affect all sub-workflows and will involve a number of different technologies. In this subsection, we provide a schematic representation of the baseline version of each sub-workflow as well as the improved version.

2.2.1 Global-scale Numerical Weather Predictions

In the baseline version of the global-scale numerical weather prediction sub-workflow, coherent with the operational execution of the IFS model at ECMWF, data generated by the numerical model are encoded in GRIB format and stored on FDB high-performance object store. FDB data and metadata are archived on Lustre parallel file system.

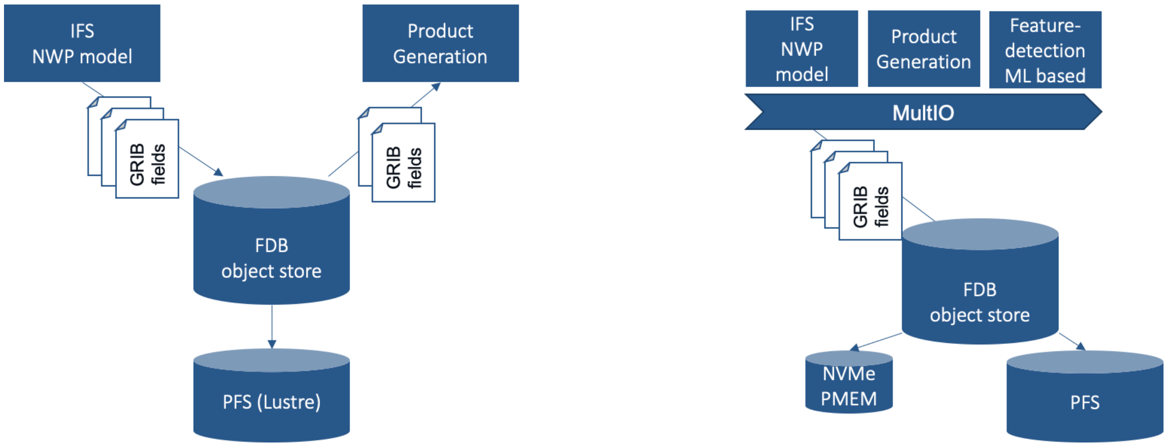


Figure 11 Baseline version of Global-scale Numerical Weather Prediction sub-workflow (left) and improved version (right)

We perform a high-resolution forecast (9km resolution) and 51 ensemble members (18km resolution) 4 times a day. Each run generates ~70TB in 1h critical window.

During the 1h time-critical window nearly 70% of the data are read-back to perform product generation (i.e., re-gridding, interpolation, cropping).

The improved version of the sub-workflow introduces two major improvements: data post-processing is performed in-situ, thanks to the adoption of the MultIO software stack that manages both product generation and ML-based feature detection (i.e., hurricane tracking)

The second improvement is related to efficient exploitation in FDB of high-performance data store. In the context of ACROSS we will develop support for heterogeneous data store, such as the adoption of different data stores for metadata (i.e., local or distributed NVMe, Storage-Class Memory) and GRIB data (distributed NVMe or HDD-based parallel file systems)

With the planned improvements, we expect to be able to nearly double the model resolution and demonstrate 5km forecasts at global-scale.

2.2.2 Climatological simulations

In the baseline version of the climatological sub-workflow, we adopt the ICON model generating GRIB and NetCDF data. The ICON model, as well as all numerical weather prediction models, generates a full description of the atmosphere at each time-step. On the contrary, data analysts require assessing the evolution of a physical variable over time, so they need time-series, thus climatological simulations needs to be post-processed to generate time-series by transposing the produced data.

Storing 1 daily output for 1 month simulation at 5km resolution requires approximately 10TB when data are encoded in GRIB2 compressed data format, thus a 30 year run requires nearly 3.6 PB.

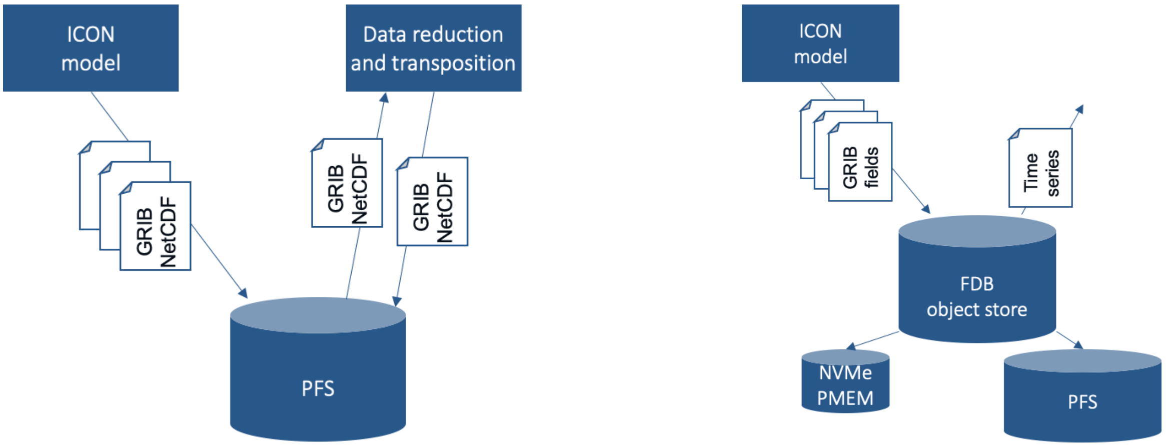


Figure 12 Baseline version of Climatological sub-workflow (left) and improved version (right)

The planned improvements includes the adoption of FDB object store for handling ICON I/O and the definition of a FDB index structure optimized for time-series data access, to completely avoid the data transposition step. Moreover, MPI will assess the behavior of the ICON model implementation for heterogeneous computing resources (CPU + GPU acceleration).

We expect to be able to demonstrate Climatological simulations at cloud-resolving resolution (i.e., 5km or better) and Grand-Ensemble climatological simulations at a coarser resolution.

2.2.3 Hydrological simulations

We are targeting full-ensemble hydrological simulation at 1km resolution over Meuse and Rhine river basins performed by WFLOW hydrological model developed by Deltares. The baseline implementation, done in Python, requires 20 minutes for each ensemble member, for a total execution time of nearly 17 hours.

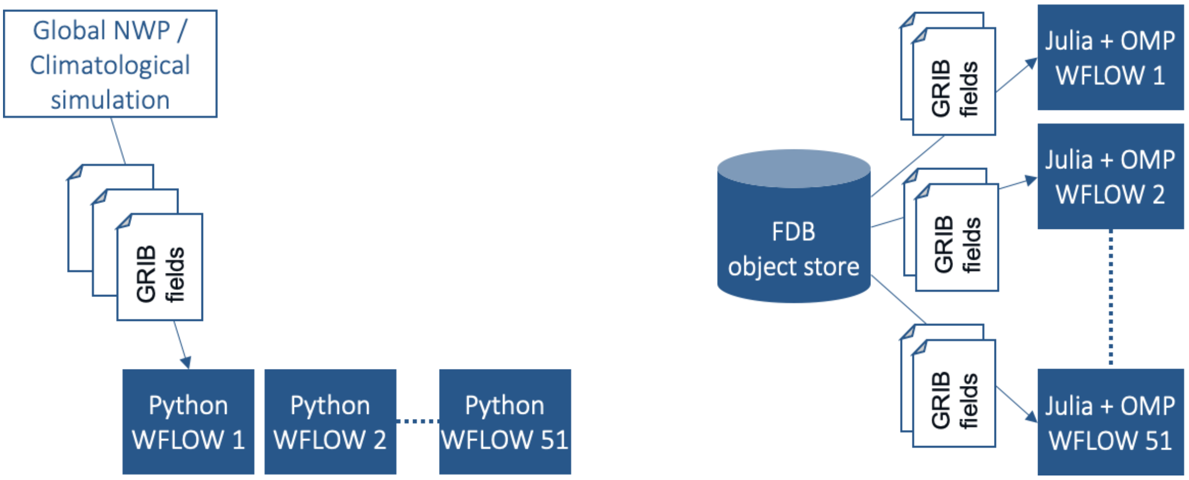


Figure 13 Baseline version of Hydrological sub-workflow (left) and improved version (right)

In the context of ACROSS project, Deltares is committed to port WFLOW in Julia language and exploit multi-threading to reduce the execution time of each member of the ensemble. Moreover, by exploiting StreamFlow and/or HyperQueue as well as the ACROSS orchestration services we are looking forward to execute in parallel the WFLOW ensemble members on cloud resources (either VMs or containers) to benefit from horizontal scalability.

We also plan to fetch forcing data directly from FDB object store, thus co-location with global-scale NWP may be beneficial in reducing overall latency.

We expect to be able to execute the whole hydrological sub-workflow (all ensemble members) in less than 1h with an overall speed-up of 20x.

2.2.4 Meso-scale NWP forecasts and Farming Advisory Services

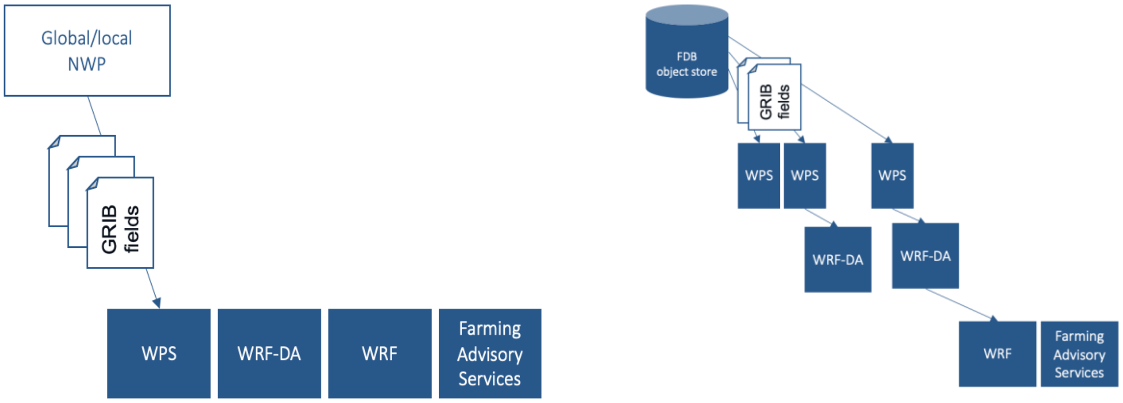


Figure 14 Baseline version of Meso-scale NWP sub-workflow (left) and improved version (right)

Meso-scale Numerical Weather Predictions are initialized by global-scale NWP at coarser resolution.

The general idea is to fetch initial and boundary condition from a meteorological or climatological global-scale model and refine the simulation on a finer spatial grid. Moreover, regional NWP models usually allow definition of sub-domains to be computed at even finer resolution.

In the baseline approach, a regional down-scaling workflow requires several input data coming form global-scale NWP and the simplest approach is to wait for completion of global forecasts and then start data preparation for the regional model.

Moreover, in the ACROSS workflow, we will perform 2 steps of data assimilation, to further improve forecast accuracy.

The baseline workflow executes a sequential chain of steps: data preprocessing, performed by WRF Pre-processing System (WPS), data assimilation, performed by WRF-DA, and eventually the regional-scale numerical weather prediction.

In the improved workflow, we aim at implementing a stream-like post-processing of global-scale NWP forecasts. In particular, we will decompose WPS and WRF-DA tasks in smaller tasks at finer granularity; their execution will be triggered by a notification system that will check the availability of the global-scale initial and boundary condition. These changes in the pre-processing and data assimilation tasks will considerably improve the starting time of the main WRF task. This stream-like processing poses several requirements on the orchestration and resource allocation services: we need a notification system able to create a link between the global-scale NWP sub-workflow and this regional-downscaling sub-workflow, then we have to handle an increased number of smaller pre-processing jobs that can be executed on cloud as well as HPC resources. Timely execution of such small jobs should be properly handled minimizing the overhead due to HPC resource allocation systems. Thus, we are considering the adoption of fine-grained scheduling techniques such as those offered by StreamFlow and/or HyperQueue.

2.2.5 Technology choices

The following table summarizes the improvement planned for WP6 and the technologies/solutions adopted in each improvement.

Improvement	**Technologies and computational resources used, actions**
In-situ data processing	*MultIO* Global-scale NWP will adopt MultIO software stack for in-situ/in-transit data processing.
Heterogeneous computing platform	*GPUs, CUDA* In the context of Climatological simulations, we will assess the benefits of heterogeneous computational resources (CPU+GPU) for the atmospheric model in ICON
Data handling	*FDB* ICON model will adopt FDB object store as main data management system. FDB, adopted by IFS and ICON models, will be extended to efficiently exploit heterogeneous data stores (local and distributed NVMe-based parallel file systems, and optionally Storage-Class Memory devices) Hydrological and regional NWP will fetch from FDB by using C++ or Python APIs
Improved analysis	*MultIO, TensorFlow* Global-scale NWP will be coupled with ML-based feature detection system (i.e., hurricane tracking). Data analysis will benefit from MultIO for in-situ data processing.
Stream processing	*ACROSS orchestration system* (StreamFlow) Regional-scale NWP will benefit from stream-like processing of input data, to reduce the latency between global-scale NWP, WRF pre-processing and data assimilation.
Better orchestration	*Kronos* Global-scale NWP will adopt Kronos for fine-grained scheduling of NWP model and all ancillary and post-processing tasks *ACROSS orchestration system (HyperQueue)* In the Hydrological simulation ensemble we will exploit cloud-based horizontal scalability by adopting virtualization (VMs or Containers) and efficient orchestration services
Code optimization	*Code refactoring, OMP* We will refactor the original WFLOW Python implementation to benefit from Julia language speed-up and we will adopt multi-threading to further improve WFLOW performances

Table 6 Summary of WP6 planned improvements and the technologies/solutions to be adopted.

2.3 Energy and Carbon Sequestration Pilot

The key technologies chosen for the WP7 pilot use cases follow from the HW/SW requirements process, as documented in D2.1, and the integration requirements process, as documented in D7.1.

We provide here an overview over the improved workflows, performance improvements expected, the technologies to be used, and some benchmarks for KPIs. We do not repeat the description of the pilot and use cases that can be found in D2.1 section 2.3.

2.3.1 Improved workflows and technology choices.

There are two main workflows to be considered. The first is a high-level workflow for ensemble studies, with applications such as history matching/inverse problems, optimization, or uncertainty quantification. A complete run of this workflow entails simulating an entire ensemble of cases, typically 50 or 100 cases, then repeating this multiple times. The second workflow represents the work done for simulation of a single case, it is therefore a sub-workflow of the first. The workflows are illustrated in the figures following.
The workflows can be improved by:

1. Better orchestration of jobs.
2. Improved data handling.
3. In-situ analysis of results.
4. Early termination of non-contributing jobs.
5. Faster/better analysis of results.

These improvements will be sought using the following technologies/platforms

Improvement	**Technologies and computational resources used, actions**
Better orchestration	*ERT, ACROSS orchestration system.* ERT: will be modified to support ACROSS-created or curated API for orchestration, as well as advanced orchestration capabilities offered on the ACROSS platform.
Data handling	*OPM Flow, Damaris.* OPM Flow: will be modified to integrate Damaris, this will be used to implement parallel output. Damaris: will be adapted to enable simple end-user use of the Damaris-enhanced OPM Flow. New plugins will be written to support new output formats.
In-situ analysis	*OPM Flow, Damaris.* OPM Flow: will be modified to integrate Damaris (as above). Damaris: new plugins will be written to support domain-specific in-situ analysis needs.
Early termination	*ERT, OPM Flow* ERT: Building on in-situ analysis of incomplete results, one can terminate jobs that are not likely to contribute towards the overall solution (optimality, history match etc.). OPM Flow: Must be able to stop a run in a controlled way.

Improved analysis	*Python, TensorFlow* Improved data handling and in-situ capabilities will enable new analysis capabilities to be realized using common and standard tools but still within the in-situ analysis framework.

Table 7 Summary of WP7 improvements and technologies/platforms to be used

In addition, container technologies (Docker, Singularity) will be used to improve ease of deployment.

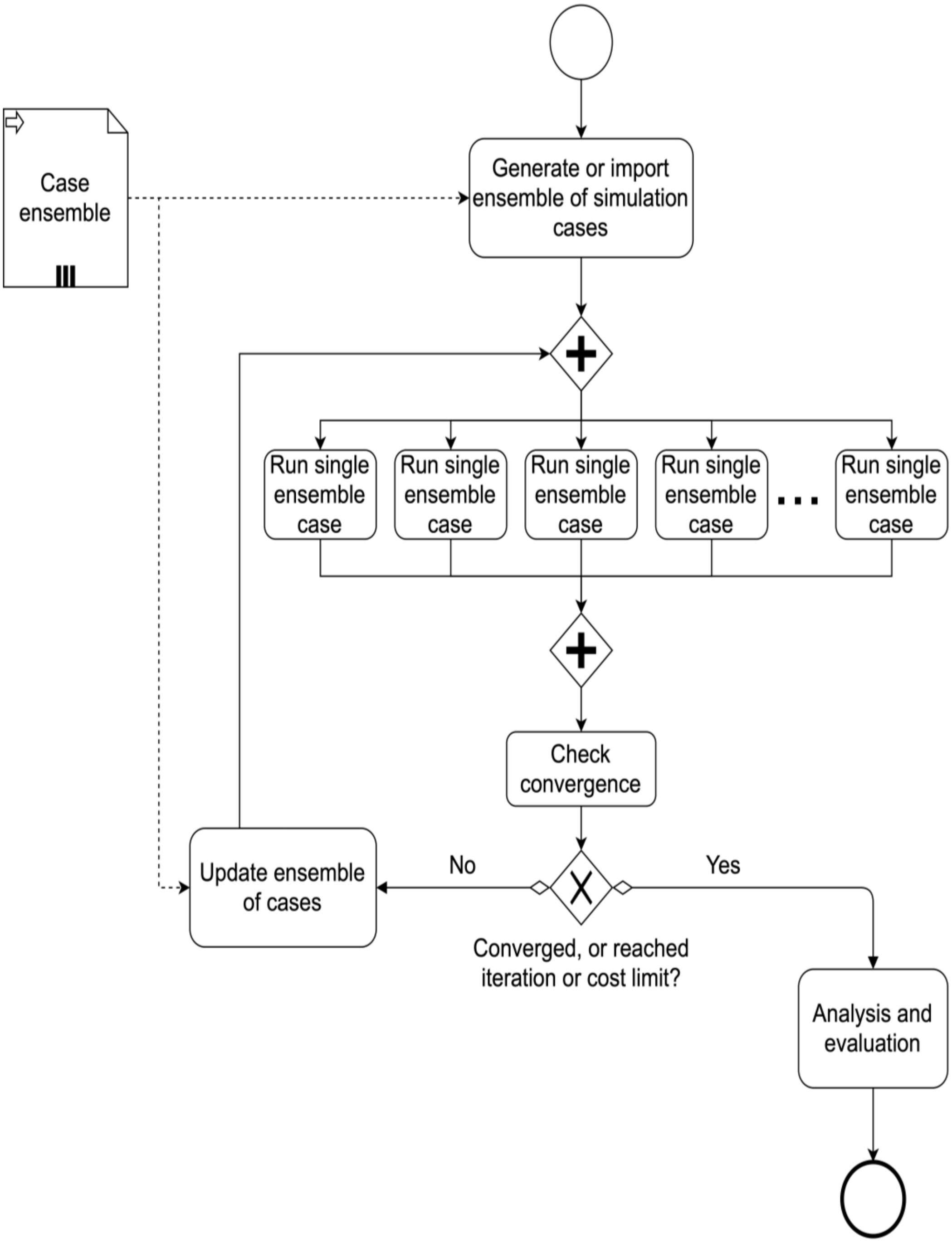


Figure 15 Workflow for high-level ensemble studies.

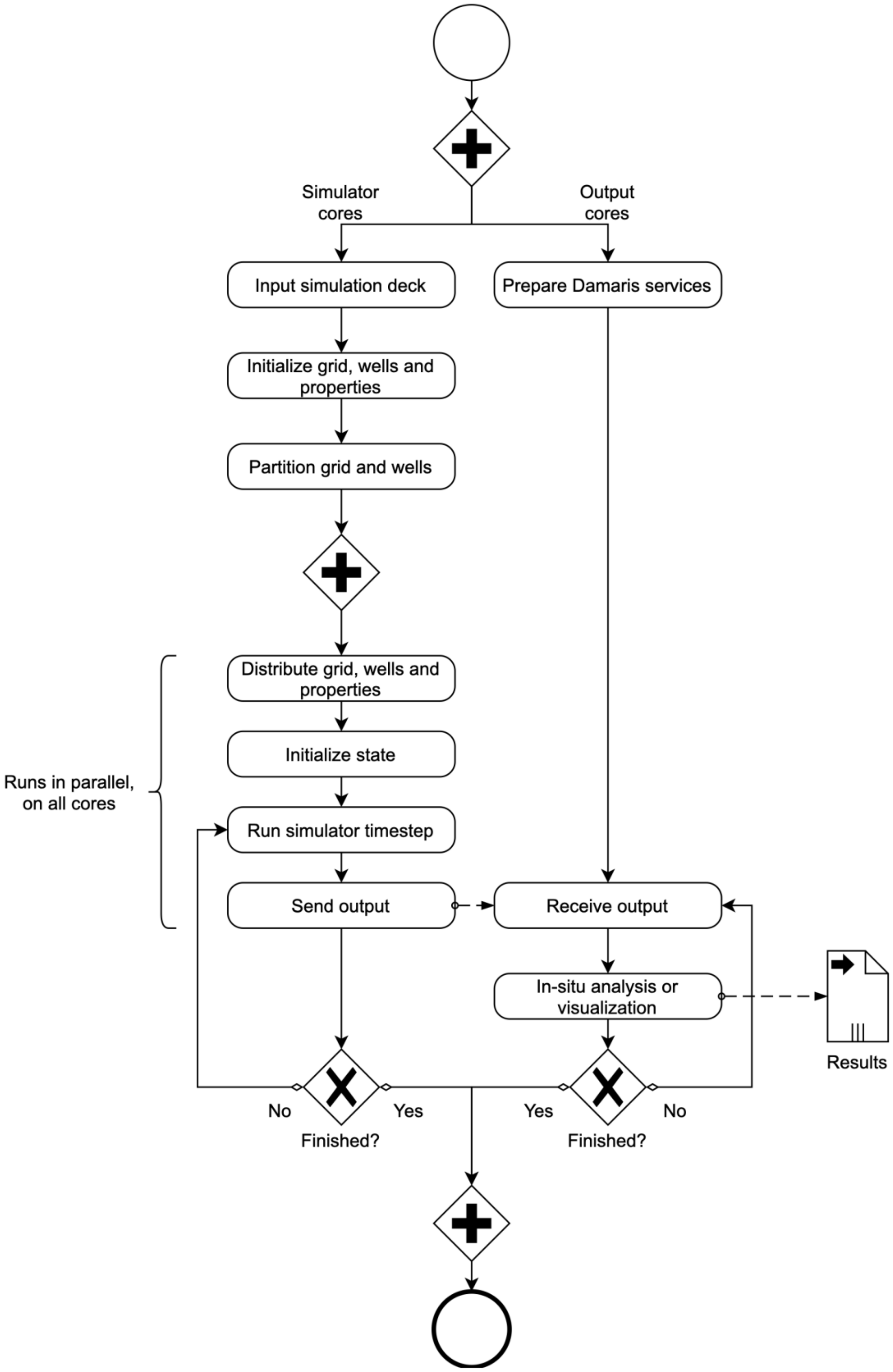


Figure 16 Workflow for simulation of a single case

2.3.2 Performance improvements and technology choices

In addition to seeking improvements to workflows and processes, we also seek performance improvements within the simulator software OPM Flow itself. This includes:

1. Improved parallel scaling.
2. Improved performance using hardware accelerators.
3. Exploiting dynamic coarsening and refinement to reduce runtimes.

The table gives an overview of the technology choices made to attain the sought improvements.

Improvement	Technologies or platforms used, actions
Parallel scaling	*OPM Flow, Damaris, MPI.* By integrating Damaris and using it to handle result output, OPM Flow will get parallel output capabilities, thereby eliminating an important bottleneck for parallel scaling.
Accelerators	*OPM Flow, GPUs, CUDA.* Acceleration of linear solver parts using CUDA has been chosen as the first target in this work. This will build on already existing experimental features of OPM Flow. Longer term, moving larger parts of the simulator to GPUs will be considered, and also using technologies other than CUDA, to avoid vendor lock-in.
Dynamic coarsening and refinement	*OPM Flow, the Dune numerical framework.* The seismic cube use case has a problem structure that is well suited for applying dynamic coarsening and refinement. This will be done initially by integrating OPM Flow with Dune grid implementations that already support dynamic refinement and load balancing.

Table 8 Summary of WP7 Performance improvements and technology choices

2.3.3 Benchmark results

KPI	Description	Target	Initial state
KPI-3.1	Improve OPM Flow runtime performance scaling when compared to today's parallel capabilities.	Efficient scaling to > 1000 processes	Efficient scaling to about 100 processes (see D2.1).
KPI-3.2	Carrying out flow simulations on large grids for long-term migration scenarios (> 1000 years)	No. of cells > 1x10^{8}	Successfully run on 18 million cell model for 300 years,
KPI-3.3	Running direct flow simulation on models consisting solely of processed seismic data, at high resolution, with automatic and dynamic coarsening / refinement.	No. of cells > 1x10^{7}	No dynamic coarsening or refinement available, early proof-of-concept exists.
KPI-3.4	Demonstrate analysis of simulation results in-situ using methods from the AI spectrum. Target is number of methods demonstrated.	3+	0
D7.1	Ability to successfully run history matching with no human intervention.	Test with Sl eipner Benc hmark	Not feasible without human intervention.

Table 9 Summary of WP7 benchmark results

3 ACROSS Platform Architecture

Figure 17 is the original image of the ACROSS Platform Architecture from the Description of Action document. This high-level overview of the architecture is still valid at the moment of writing this document, and it is still in line with the original requirements, requirements coming from the pilot's use-cases, and identified key technologies.

The proposed architecture contains two main pillars. Starting at the bottom of the architecture, WP3 deals with the heterogeneous hardware and acceleration support and is considered hardware and infrastructure layer of the platform. WP4 deals with the multi-level orchestration support and acts as the workflow orchestration layer of the platform. The interconnection of the hardware/infrastructure layer and orchestration layer acts as core of the entire ACROSS platform that will provide access to heterogeneous hardware and advance orchestration services.

The illustrated high-level architecture is still valid; however, what evolved is the detailed description of the internal ACROSS platform subsystem i.e., orchestration system architecture presented in Section 6.

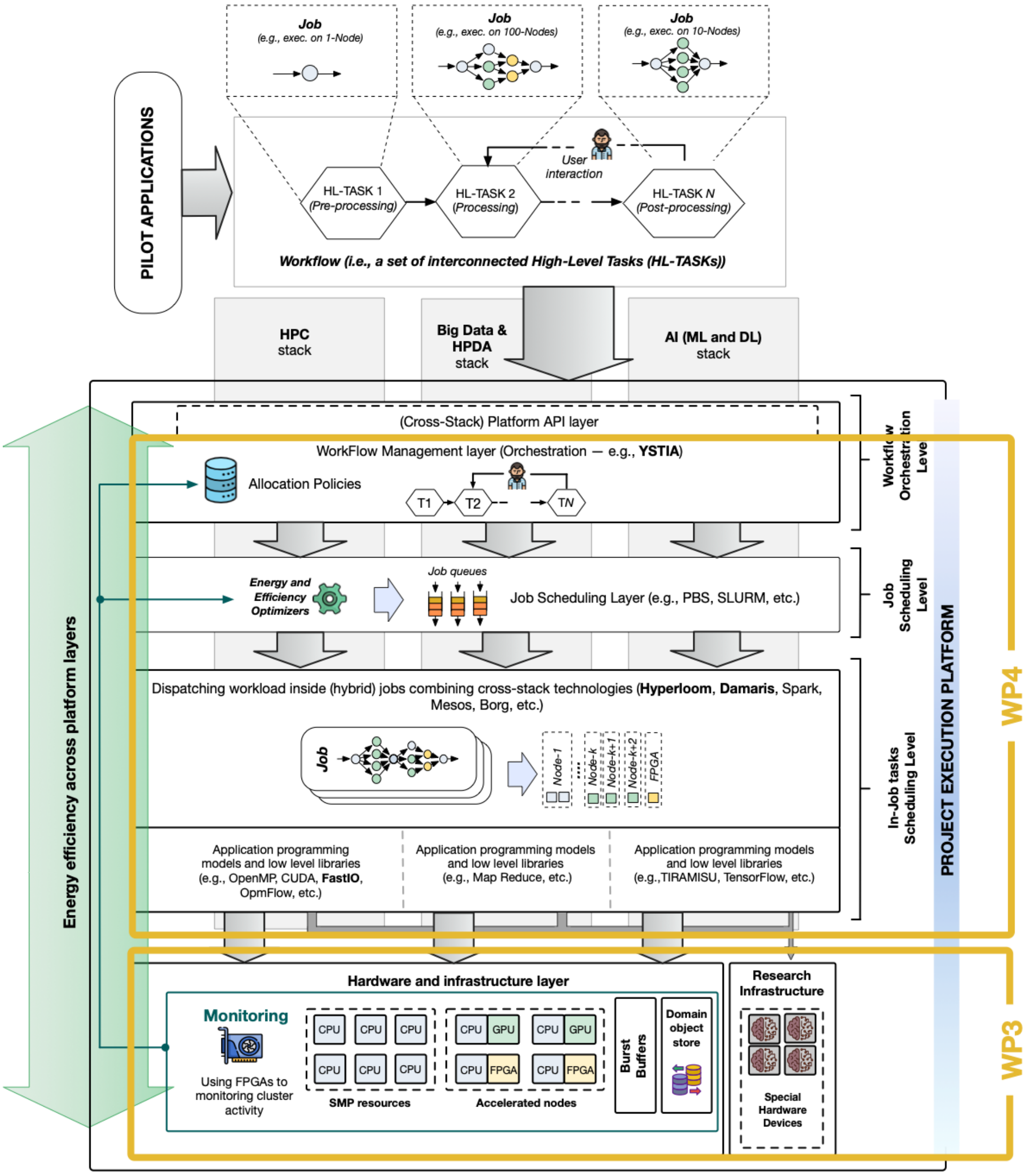

Job
(e.g., exec. on 1-Node)
Job
(e.g., exec. on 100-Nodes)
Job
(e.g., exec. on 10-Nodes)
PILOT APPLICATIONS
User interaction
HL-TASK 1 (Pre-processing)
HL-TASK 2 (Processing)
HL-TASK N (Post-processing)
Workflow (i.e., a set of interconnected High-Level Tasks (HL-TASKs))
HPC stack
Big Data & HPDA stack
AI (ML and DL) stack
(Cross-Stack) Platform API layer
WorkFlow Management layer (Orchestration — e.g., YSTIA)
Allocation Policies
T1
T2
TN
Workflow Orchestration Level
Energy and Efficiency Optimizers
Job queues
Job Scheduling Layer (e.g., PBS, SLURM, etc.)
Job Scheduling Level
Dispatching workload inside (hybrid) jobs combining cross-stack technologies (Hyperloom, Damaris, Spark, Mesos, Borg, etc.)
Job
Node-1
Node-k
Node-k+1
Node-k+2
FPGA
In-Job tasks Scheduling Level
Application programming models and low level libraries (e.g., OpenMP, CUDA, FastIO, OpmFlow, etc.)
Application programming models and low level libraries (e.g., Map Reduce, etc.)
Application programming models and low level libraries (e.g.,TIRAMISU, TensorFlow, etc.)
Energy efficiency across platform layers
PROJECT EXECUTION PLATFORM
WP4
WP3
Hardware and infrastructure layer
Monitoring
Using FPGAs to monitoring cluster activity
CPU
GPU
FPGA
SMP resources
Accelerated nodes
Burst Buffers
Domain object store
Research Infrastructure
Special Hardware Devices

Figure 17 ACROSS platform architecture

4 Key Hardware Technologies & platforms

The content of this section results from the co-design activities conducted within WP2. Co-design means jointly designing hardware and software architectures to meet quality, performance, cost and energy goals, therefore co-design should be a different methodology than the layered abstractions used in general-purpose computing. However, with the objectives of improving a set of pilots of various nature and creating at the same time an ecosystem capable of supporting a wide spectrum of applications, the co-design function shifts towards a more traditional meet-in-the-middle approach in which the hardware design will target an architecture as open as possible by integrating a rich set of heterogeneous computing technologies in a cooperative and scalable manner, leaving to the orchestration the task of exploiting it as optimally as possible according to the needs of a specific application.

This consideration leads us to consider a hardware architecture built on multi-core nodes enriched with a wide range of accelerators varying from general-purpose GPUs, FPGAs to more specific AI-acceleration devices like NNPs, VPUs and even neuromorphic emulators/simulators, while emphasizing the communication between these computing devices (in terms of data access, protocol, etc.) as well as their potential for further extension. The hardware will be abstracted by layers of software, which will have to be combined judiciously by the orchestrator for an optimal use of these computing technologies, according to their availability and their configuration.

During the reporting period, the technological choices related to the hardware support for the ACROSS platform (WP3) started, by means of the investigative work in common with the application pilots from which the following results have been derived:

- a) **Characteristics of Applications**
  - a. WP5 : Combustor Simulation pilot and Turbine Design pilot
    - i. A mix of commercial (STAR-CCM+) and proprietary software
    - ii. Computing Technology/Programming Model : CPU/MPI+
    - iii. Combination of simulation (URANS/LES), HPDA (MatLab) and AI (ANN)
  - b. WP6 : Hydro-meteorology and Hydro-climatology
    - i. A mix of proprietary and open source
    - ii. Criticality is on database access, huge data transfer and analysis
    - iii. Computing Technology/Programming Model : CPU/OpenMP+MPI
  - c. WP7 : OPM Flow Reservoir Simulation
    - i. GNU General Public License software
    - ii. Computing Technology/Programming Model : CPU/MPI
    - iii. Experiment in acceleration of Linear Solver part on GPUs (CUDA/CuSparse and OpenCL)
    - iv. Experiment in acceleration of preconditioner BICGStab section on Xilinx FPGA
- b) **Requirements and Action Items**
  - a. Common Requirements
    - i. Engineering Computation
    - ii. Handling of larger problems with more precision
    - iii. Performance improvement, portability and standardization
  - b. Current co-design actions
    - i. Cineca/WP5: Exploitation of G100 tuning (cache, vectorization, NUMA policies, interconnection…) for performance.
    - ii. Cineca/WP5: Acceleration of HPDA tools.
    - iii. ATOS-Cineca/WP5: ANN/SNN Modelling and Acceleration of training and inference
    - iv. Cineca/WP6: Exploitation of different G100 data storage technologies (IME, Intel Optane persistent memory) for improving I/O performances (via Fdb object store)
    - v. LINKS/WP7 targets Barbora and Karolina with Nividia/CUDA and AMD/HIP
      1. Focus is on the preconditioner of the large linear systems involved in the OPM WF.
      2. Analyze the current experimental GPU version of the preconditioner to identify performance issues.
      3. Investigate different available preconditioners/implementation.

On the identification of the key hardware technologies particular attention have been paid to the compliance with the EPI guideline and Exascale perspective. To this end, ATOS has done two presentations to ACROSS consortium.
The ATOS/Bull platform is under design, taking into account the needs of project pilots as well as SoA AI-related computing technologies. CINECA and IT4I computing resources that will be used in the project are presented below.

CINECA computing resources

Galileo100 cluster

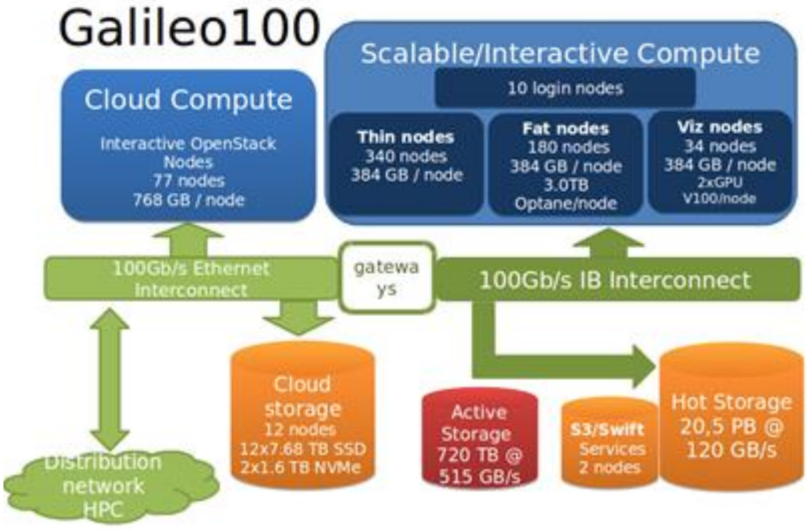


Figure 18 G100 System Architecture

The Galileo100 (G100) system architecture is depicted in Figure 18. In the following, the details are provided:

- 554 computing nodes each equipped with 2 CPU Intel CascadeLake 8260, sporting 24 cores each and running at 2.4 GHz, 384GB RAM, subdivided in:
  - 340 standard nodes ("thin nodes") with 480 GB of SSD storage
  - 180 data processing nodes ("fat nodes") with 2TB of SSD storage and 3TB of Intel Optane memory
- 34 GPU nodes with 2x NVIDIA GPU V100 with 100Gbs Infiniband interconnection and 2TB SSD.
- 77 computing server OpenStack for cloud computing (ADA CLOUD), 2x CPU 8260 Intel CascadeLake, 24 cores running at 2.4 GHz, 768 GB RAM, with 100Gbs Ethernet interconnection.
- 20 PB of active storage accessible from both Cloud and HPC nodes.
- 5 PB of fast storage for HPC system.
- 1 PB Ceph storage for Cloud (full NVMe/SSD)
- 720 TB fast storage (IME DDN solution)

CINECA, in collaboration with Intel, is supporting the WP6 activity regarding the exploitation of different G100 data storage technologies (IME, Intel Optane persistent memory, etc.) in order to manage and improve the I/O performances.

Leonardo supercomputer

CINECA will host one of the three pre-Exascale class supercomputers, which has been funded by the European Commission in the context of the EuroHPC-JU, and that will be hosted in the new main datacenter located in Bologna (Italy). This new machine will be based on the Atos BullSequana XH2000 architecture, and will be equipped with nearly 14,000 next generation NVIDIA Ampere architecture-based GPUs, NVIDIA Mellanox HDR InfiniBand, and over 100 petabytes of state-of-the-art storage capacity, which will provide 10 Exaflops (10 EFlop/s) of FP16 AI performance. This new machine will be capable of an aggregated HPL performance of 250 PFlop/s (HPL Linpack Performance (Rmax)), enabling the researchers and scientists to make new discoveries, and contribute to the management and mitigation of critical situations due to extreme events. The main features of Leonardo supercomputer are:

- 3 Modules: more than 136 BullSequana XH2000 Direct Liquid cooling racks
- 5000 computing nodes:
  - 3456 servers equipped with Intel Xeon Ice Lake and NVIDIA Ampere architecture GPUs
  - 1536 servers with Intel Xeon Sapphire processors
- 3+PB RAM
- 5PB of High Performance storage
- 100PB of Large Capacity Storage
- 1TB/s bandwidth
- 200Gb/s interconnection bandwidth
- 9MW
- PUE 1,08
- 1500+ m2 footprint

The ACROSS project will devise to leverage on the Leonardo system resources as part of the infrastructural support for the pilot workflows execution, whenever the system will become accessible and the resources will be available.

IT4Innovations computing resources

Below is the overview of the IT4Innovations' compute systems. The more detailed description of how users can apply for the IT4I computational resources is available on the IT4I web site in the section about Computing Resources Allocation: https://www.it4i.cz/en/for-users/computing-resources-allocation

Barbora cluster

Barbora, installed in the autumn of 2019, provides a theoretical peak performance of 849 TFlop/s. The computing system consists of:

- 189 standard computational nodes; each node is equipped with two 18-core Intel processors and 192 GB RAM.
- 8 compute nodes with GPU accelerators; each node is equipped with two 12-core Intel processors, four NVIDIA Tesla V100 GPU accelerators with 16 GB of HBM2 and 192 GB of RAM.
- 1 fat node is equipped with eight 16-core Intel processors and 6 TB RAM.

The supercomputer is built on the Bull Sequana X architecture and for cooling its standard compute nodes the direct liquid cooling technology is used. The computing network is built on the latest Infiniband HDR technology. The SCRATCH computing data storage capacity is 310 TB with 28 GB/s throughput using Burst Buffer acceleration. Another computing data storage is based on NVMe over Fabric with a total capacity of 22.4 TB dynamically allocated to compute nodes. It is also equipped with the ATOS/BULL Super Computer Suite cluster operation and management software solution as well as PBS PRO scheduler and resource manager.

NVIDIA DGX-2

NVIDIA DGX-2, was installed in the spring 2019 with 2 PFlop/s peak performance in AI. It is equipped with 16 powerful data centre accelerators – the NVIDIA Tesla V100 GPU. They are inter-connected with revolutionary NVSwitch 2.0 technology that delivers a total bandwidth of 2.4 TB/s. The systems include 512 GB of HBM2 memory. The NVIDIA DGX-2 also offers 30 TB of internal capacity on fast NVMe SSDs disks. Interconnection to the surrounding infrastructure is provided via eight 100 Gb/s Infiniband/Ethernet adapters.

One NVIDIA DGX-2 can replace 300 dual-socket servers with Intel Xeon Gold processors for deep neural network training (ResNet-50). NVIDIA DGX-2 is powered by the DGX software stack; NVIDIA-optimized and tuned AI software that runs the most popular machine learning and deep learning frameworks with maximized performance. The NVIDIA DGX-2 can also be used for traditional HPC workloads to deliver a theoretical peak performance of 130 TFlop/s.

Karolina cluster

Karolina was launched in 2021. The new supercomputer reaches a theoretical peak performance of 15.2 PFlop/s.

The supercomputer consists of 6 main parts:

- a universal partition for standard numerical simulations, which consists of 720 computer servers with a theoretical peak performance of 3.8 PFlop/s,
- an accelerated part with 72 servers, each of them being equipped with 8 GPU accelerators providing an aggregated performance of 11 PFlop/s for standard HPC simulations and up to 150 PFlop/s for artificial intelligence computations,
- a part designated for large dataset processing that will provide a shared memory of as high as 24 TB,
- 36 servers with an aggregated performance of 131 TFlop/s will be dedicated for providing cloud services,
- a high-speed network to connect all parts as well as individual servers at a speed of up to 200 Gb/s,
- fast data storages with capacity of more than 1 PB for high-speed user data storage with speed of 1 TB/s primarily for simulations as well as computations in the fields of advanced data analysis and artificial intelligence.

LUMI supercomputer

LUMI is a pre-Exascale supercomputer currently being constructed in CSC’s data center in Kajaani, Finland; it is one of the three pre-Exascale supercomputers which have been financed thanks to the EuroHPC Joint Undertaking (the other two being Leonardo -Italy, and MareNostrum5 -Spain). It will be the most powerful of the three, with a peak performance of over 550 PFlop/s. Its specifications include the following features:

- a CPU partition with approximately 200,000 AMD EPYC cores
- a GPU partition powered by AMD Instinct GPUs (over 550 PFlop/s)
- 80 PB parallel file system
- Interactive partition with 32 TB of memory and graphics GPUs for data analytics and visualization.
- Accelerated (Flash) storage and encrypted object storage services.

An important characteristic of the LUMI architecture is the use of AMD GPUs, i.e., accelerated hardware which cannot run software written in CUDA (which is available only for NVIDIA devices). This hardware therefore offers a testbed for software that can run on any accelerated devices, not just NVIDIA GPUs. The LUMI system is expected to be in production by the end of 2021.

5 Key software Technologies

The software tools identified as key software technologies for the Across platform, and in particular to satisfy the pilots use case requirements (cf. mapping matrix), are listed and described below.

Name	YSTIA / YORC
Developed by	Atos (BDS R&D)
License	Open Source, Apache 2.0
Website	https://ystia.github.io/
About	TOSCA based orchestrator (incl. TOSCA Forge Management, Front End, Application lifecycle and workflows management, hybrid Cloud/HPC support).

Table 10 YSTIA / YORC

From pilots requirements has been identified the need to orchestrate workflows mixing HPC, BD and AI operations. YSTIA/YORC is intended to be used for managing the higher level of the applications (workflows), i.e., handling the coarse-grained components deployment and execution of HPC, BD and AI high-level operations (i.e., steps in the application workflows —see the appendix). To build the global ACROSS orchestration solution, it will be integrated with other complementary software like StreamFlow and HyperQueue.

Name	Fast Machine Learning Engine (FMLE)
Developed by	Atos (BDS R&D)
License	Atos proprietary (subject to change)
Website	
About	ML/DL toolbox for HPC, Hide complexity of HPC jobs management for AI Model Management (training...). Provides GUI/CLI/API for managing model trainings, datasets.

Table 11 Fast Machine Learning Engine (FMLE)

In some of the Pilot use cases, it is required to manage AI models training phases. In these cases, FMLE can be used, as it provides high-level interfaces to handle these AI steps (potentially being part of the workflow). It may also be used from a data scientist end user perspective, in order to facilitate the AI models implementation.

Name	Workflow-aware Advanced Resource Planner (WARP)
Developed by	LINKS Foundation
License	This module will be developed under an Open-Source license
Website	N/A
About	This component allows to reserve HPC resources (in advance) by applying advanced and workflow-aware planning. This planning is based on the specific resources needed by each workflow to be executed and it combines information provided by the infrastructure monitoring systems, targeting the improvement of the energy efficiency and resource utilization.

Table 12 Workflow-aware Advanced Resource Planner (WARP)

WARP is a software component that will be designed in the context of ACROSS project, as one of the components of the ACROSS orchestrator. WARP is intended to plan the allocation of computational resources for the application workflows in an optimized way, by leveraging the advanced reservation capabilities offered by state-of-the-art batch schedulers. In such a way, it targets the improvement of the execution of the application workflows by matching in advance the required computational resources, as well as to improve the overall energy efficiency by better matching the application needs with the resource availability.

Name	High-End Application Execution Middleware (HEAppE)
Developed by	IT4Innovations national supercomputing center (www.it4i.cz)
License	Open-Source GPL-3.0 License
Website	http://heappe.eu/
About	HPC remote execution middleware. Implementation of an HPC-as-a-Service concept. Provides secure remote access to HPC without the need to manually register within the HPC center. For more information, see the official website.

Table 13 High-End Application Execution Middleware (HEAppE)

Utilization of HEAppE Middleware relates to task T2.5 - Simple access and service provisioning. HEAppE is able to provide simple and easy to use remote and secure access to HPC infrastructures via a REST API. Command templates will be created for relevant WP5, WP6, WP7 use-cases to support the orchestration layer and in the scope of WP2 and WP4 the evaluation of extension by energy consumption reports.

Name	HyperTools
Developed by	IT4Innovations national supercomputing center (www.it4i.cz)
License	Open-Source (various licenses)
Website	TBA (WIP)
About	HyperTools are a set of tools developed at IT4Innovations for running application on supercomputers. The tool set contains: • HyperQueue - a scheduler that transparently schedule tasks through HPC system schedulers like PBS or SLURM. https://github.com/It4innovations/hyperqueue • HyperLoom - a tool mainly for running external programs; https://github.com/It4innovations/hyperloom • RSDS - A Dask server reimplementation in Rust for running Python tasks. https://github.com/It4innovations/rsds • Quake - a tool for multi-node scheduling of MPI programs. https://code.it4i.cz/boh126/quake

Table 14 HyperTools

In the scope of the WP5 the possibility to use HyperTools as kernel execution platform will be evaluated. Evaluation of possibility to optimise/extend/replace ERT or its parts by HyperTools in WP4, WP7. In general the extension by energy consumption reports and by energy efficiency aware scheduling strategies will be done.

Name	StreamFlow
Developed by	CINI (University of Turin)
License	Open Source, LGPLv3
Website	https://streamflow.di.unito.it/
About	StreamFlow is a container-native Workflow Management System written in Python 3 and based on the Common Workflow Language (CWL) standard. It is

	designed for scheduling and coordinating different workflow steps on top of a diverse set of execution environments, ranging from practitioners' desktop machines to entire HPC centers. In particular, each step of a complex pipeline can be scheduled on the most efficient infrastructure, with the underlying runtime layer automatically taking care of worker nodes' life cycle, data transfers, and fault-tolerance aspects.

Table 15 StreamFlow

StreamFlow will be a component of the ACROSS orchestration architecture, developed in WP4, supporting the high-level applications workflows modeling and orchestrating the execution of the workflow steps (see the appendix) over the HPC/Cloud resources. It will be integrated with YSTIA/YORC to manage the workflow's steps deployment and with the HyperTools (mostly HyperQueue) to manage the low-level execution, also taking into account the scheduling plans produced by WARP.

Name	Damaris
Developed by	Inria (KerData team, Inria Rennes Bretagne-Atlantique)
License	Open Source, LGPL
Website	https://project.inria.fr/damaris/
About	Damaris is a middleware for asynchronous I/O and data management targeting large-scale, MPI-based HPC simulations. It is a middleware system that leverages dedicated cores in multicore nodes to offload data management tasks, including I/O, data compression, scheduling of data movements, in-situ analysis and visualization.

Table 16 Damaris

Damaris will be used in OPM Flow to handle result output. This will enable parallel output capabilities in OPM Flow and will mitigate a major I/O bottleneck that limits the scaling of the application.

Name	Exascale Monitoring (ExaMon)
Developed by	Francesco Beneventi <francesco.beneventi@unibo.it>, supported by the EU FETHPC project ANTAREX and EU ERC Project MULTITHERMAN
License	University of Bologna Proprietary
Website	https://docs.google.com/document/d/1QePNvI5kMCYeOX5PnoCk9lwrzRFBZgZkPdbS2P45E_8/edit?usp=sharing
About	ExaMon is a data collection and analysis platform oriented to the management of big data. Its main prerogatives are to manage in a simple way heterogeneous data, both in streaming and batch mode, and to allow the access to these data through a common interface. This simplifies the usability of data supporting applications such as real time anomaly detection, predictive maintenance and efficient resource and energy management using techniques in the domain of machine learning and artificial intelligence. Given its scalable and distributed nature, it is readily applicable to HPC systems, especially Exascale sized ones, which is also the primary use case it was designed on.

Table 17 Exascale Monitoring (ExaMon)

ExaMon solution has been developed with the aim of collecting data from nodes' sensors into a database, that can be queried later for performing performance and predictive analysis, as well as anomaly detection.

Name	FDB
Developed by	ECMWF

License	Open Source, Apache License Version 2.0
Website	https://github.com/ecmwf/fdb
About	FDB (Fields DataBase) is a domain-specific object store developed at ECMWF for storing, indexing and retrieving meteorological data. Designed mainly for GRIB data, it also support observations encoded in ODB data format and can be easily extended to support additional data format. Data indexing is based on metadata automaticaly extracted from the archived dataset. In case of GRIB data, each message is stored as a field and indexed trough semantic metadata (i.e. physical variables such as temperature, pressure, etc.). A set of fields can be retrieved specifying a request using a specific language developed for accessing the MARS Archive

Table 18 FDB

FDB has been developed as I/O stack for the IFS global-scale numerical-weather-prediction model, and has the main purpose to cope with large-scale time-critical ensemble simulations. In the context of ACROSS we aim at improving FDB performances by efficiently exploiting hierarchical data stores. FDB will also be adopted by ICON model to support climatological simulations at cloud-resolving resolution and by other WP6 numerical models as main tool for meteorological data handling.

6 Software Platform for HPC/BD/ML Applications Orchestration

The ACROSS project foresees the design and implementation of an Exascale-ready execution platform that is able to execute application workflows that combine large numerical simulations along with big data analytics, machine learning and deep learning steps; we refer to this type of workflows as cross-stack workflows. To this end, ACROSS will leverage on a broad range of hardware resources, which provide the basic substrate to efficiently execute the cross-stack workflows. As such, hardware differentiation and specialization (here, different processor architectures, as well as hardware accelerators like GPUs, FPGAs, Neural Network Processors, Neuromorphic devices, and various flavours of memory technologies are considered) are the key to achieve the highest ratio between performance and power consumption. However, the solely heterogeneity of computing resources is not sufficient to achieve the overall project objectives: an effective system to manage the execution of the workflows, as well as the allocation of computing resources is a must to extract the maximum performance possible from the underlying infrastructure. On the other hand, only targeting the specialization of the computing resources is not enough to expose to the application layer an execution environment which maximize the performance and the energy efficiency. Indeed, state-of-the-art supercomputers are based on optimized networking architectures, improved memory hierarchy (i.e., non-volatile memory (NVM) technologies allow to integrate new levels in the memory hierarchy: Storage-Class Memory provides access to memory pools larger than those ones based on DRAM, with an access time near (still higher) to that of DRAM devices) and fast storage subsystems (also here, NVM technologies provide the substrate for faster mass storage than that based on traditional spinning disks); as such, the orchestrator must take into account such subsystems (e.g., trying to minimize the traffic on the network due to job management and synchronization) in order to optimize the whole workflow execution. Also, cluster topology (i.e., how the computational, networking and storage resources are organized and served to the application layer; for instance, state-of-the-art supercomputers are organized in partitions like Cloud partition, booster/accelerated partition, etc.) provides relevant information for the overall optimization of the workflows' execution.

Work Package 4 – Multi-level Orchestrator Towards Heterogeneous Exascale Computing (WP4) oversees the designing phase of the software platform that will provide all the features needed to support the execution of the Pilot cross-stack workflows, leveraging on the set of software technologies reported in Section 4. To this end, the WP4 devised a multi-level architecture of the orchestrator that allows a more flexible execution of the (cross-stack) workflows, with an easier (and finer) control over the computational resources with respect to what the solely state-of-the-art batch schedulers (e.g., SLURM, PBS) allow. To this end, the devised ACROSS Orchestrator will allow to allocate computational resources by matching workflows' requirements and getting control over the allocated resources with a fine granularity, i.e., computing resources of a node (CPU cores and access to hardware accelerators) will be assigned in such a way they will serve for executing different part of the workflow.. Execution efficiency will be achieved by letting the ACROSS Orchestrator to have the visibility over the workflows. In addition, including the information gathered from the underlying HPC monitoring systems (i.e., HDEEM, ExaMon, etc.) it will be possible to optimize the execution of the workflows and improve the energy efficiency.

6.1 ACROSS Orchestrator architecture

The ACROSS Orchestrator has been architected in such a way the workflow execution and energy efficiency will be achieved through: i) optimized allocation of the computing resources by leveraging on the complete visibility over the whole set of workflows to be executed in contrast to the today mostly adopted job and resources management frameworks which have a job-level visibility [1][2]; ii) a fine control of the computing resources that allows to allocate slices of the resources (for the same user or within the same workflow) available on the HPC nodes (i.e., providing the capability of executing a task on a subset of the available CPU cores). In the former case, the ACROSS orchestrator will have visibility on a number of upcoming jobs and will try to leverage this knowledge to improve the resources allocation. In the latter case, the orchestrator will be able to improve the system throughput by executing more tasks (see the appendix for a definition of the concept of task and job adopted in ACROSS) concurrently in those cases resources at the node level are underused. Indeed, a software component (HyperQueue) is foreseen to schedule the tasks execution on acquired resources with a fine-grain resolution (e.g., binding the execution of a task to a group of cores within the same CPU socket). Cross-stack workflows containing ML/DL tasks require to take care of the operations involved in such tasks (e.g., training large ML/DL models using multiple GPUs and eventually multiple nodes) by adopting dedicated solutions for managing their execution and the allocation of needed resources. Furthermore, the initial steps involved in the definition of ML/DL models may require interactive execution sessions to avoid incurring in long waiting phases of the associated jobs due to the queuing system. A special case is also well represented by the integration of innovative hardware systems, like the case of neuromorphic architectures. In such case, the traditional tools used to manage the execution of workflows need to be complemented with a dedicated management framework. As such, this framework should be responsible for abstracting all the mechanisms involved in the creation and execution of DL models tailored for neuromorphic architectures (e.g., spiking neural networks - SNNs). To address all these cases, ACROSS Orchestrator includes dedicated modules that ease the management of all the phases associated to the design, training, and inference of a ML/DL model, with a special focus on the way of serving neuromorphic architectures (which will be targeted by WP3). In addition, interactive sessions will be supported by leveraging on resources served through the Cloud partitions made available as part of the ACROSS infrastructural layer. Supporting such flexibility requires the integration of different software modules. Figure 19 depicts the high-level architecture of the ACROSS Orchestrator, where the main software components and their relationships are represented.

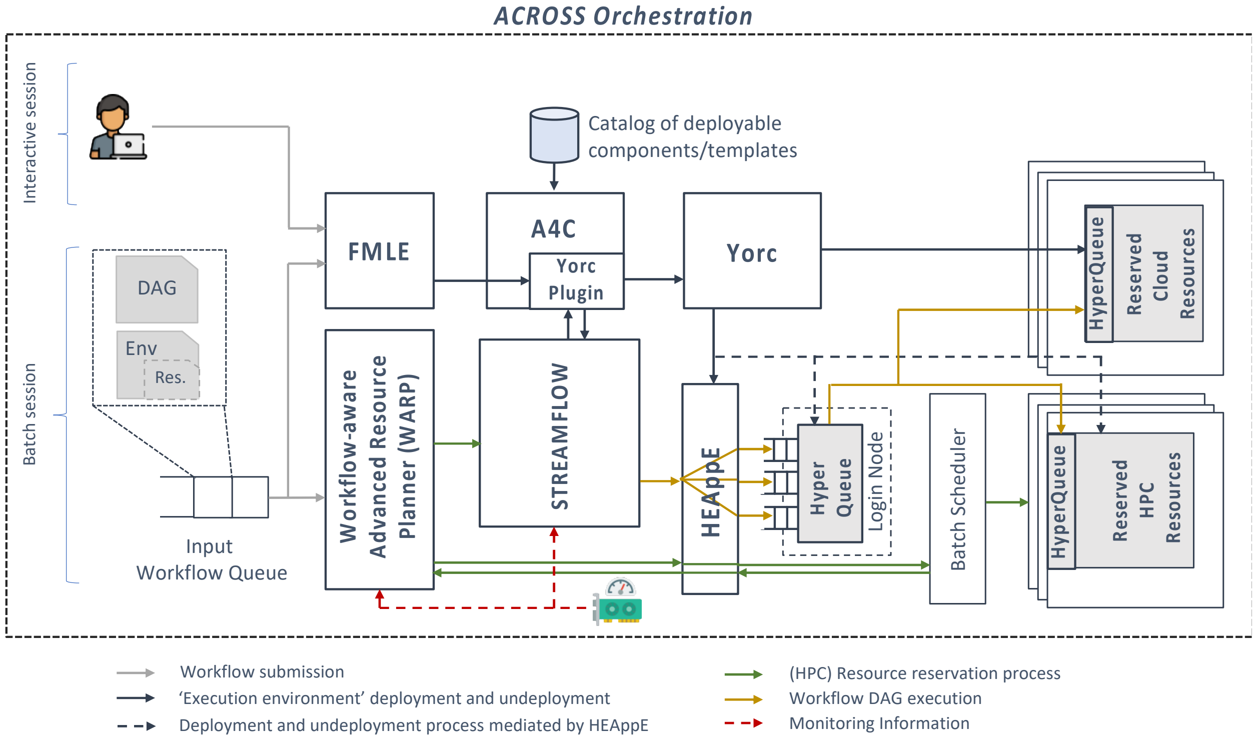


Figure 19 High-level ACROSS Orchestrator architecture

The Workflow-aware Advanced Resource Planner (WARP) is the main frontend of the orchestration. Indeed, it manages an input queue where the users submit their workflows. Workflows are described by two separated inputs: *i*) the description of the execution environment, which contains the description of the required computing resources (on both the Cloud and HPC infrastructure); *ii*) a description of the application workflow (this can be modelled as a directed acyclic graph (DAG)). The WARP internal planning logic will be designed in such a way that it will leverage the visibility on the future jobs in a workflow: for instance, it will be able to co-allocate resources for jobs that are logically sequential, but may coexist if a workflow is executed in a streaming fashion. To this end and to improve user experience, it will exploit the 'advanced reservation' capability exposed by state-of-the-art batch schedulers (e.g., SLURM, PBS), which allow to reserve (in advance) a given number of resources (green arrows in Figure 19) [3]. Doing so, the batch-scheduler allows to define a point in (the future) time where the requested resources will be available, and thus they could be assigned to a given user or group.

As such, until that point in time, the resources are used by the batch-scheduler to serve jobs that are currently in the managed queues. Figure 20 provides an example of such workflows-aware advanced resources reservation. On the left side, three different workflows (i.e., *W1*, *W2* and *W3*) are queued on the ACROSS Orchestrator; within each workflow, a group of jobs is highlighted along with the required resources to be executed (i.e., blocks *B1*, *B2*, *B3*, *B4*, *B5* and *B6*). The WARP module schedules the allocation of resource blocks over the time in such way parallel workflow execution should be maximized. For instance, in Figure 20, blocks pair B1, B2 (each belonging to a different submitted workflow for the execution), as well as group *B1, B3* and *B4 (*belonging to different workflows) are allocated on the HPC system at the same time, although they span over time with different durations. Concerning the reservation of Cloud resources, the WARP module leverages on the fact that Cloud environments (e.g., those one based on the OpenStack framework) can natively provide virtual resources on demand. Furthermore, the 'advanced reservation' capability of the batch scheduler is wrapped up by a calendar-based system, that provides a unified view of the number of reserved resources (i.e., number of required nodes, type of resources, association to a given workflow/job, etc.) and the time-span of such reservations.

Once the resources have been acquired, the StreamFlow engine can start performing the following steps:

1. Asking YSTIA (which is the collective name for the Alien4Cloud –A4C-, and YORC modules) to deploy the execution environment (including the usage of reserved HPC resources) –i.e., dark blue arrows in Figure 19;
2. Scheduling the execution of workflows' steps.

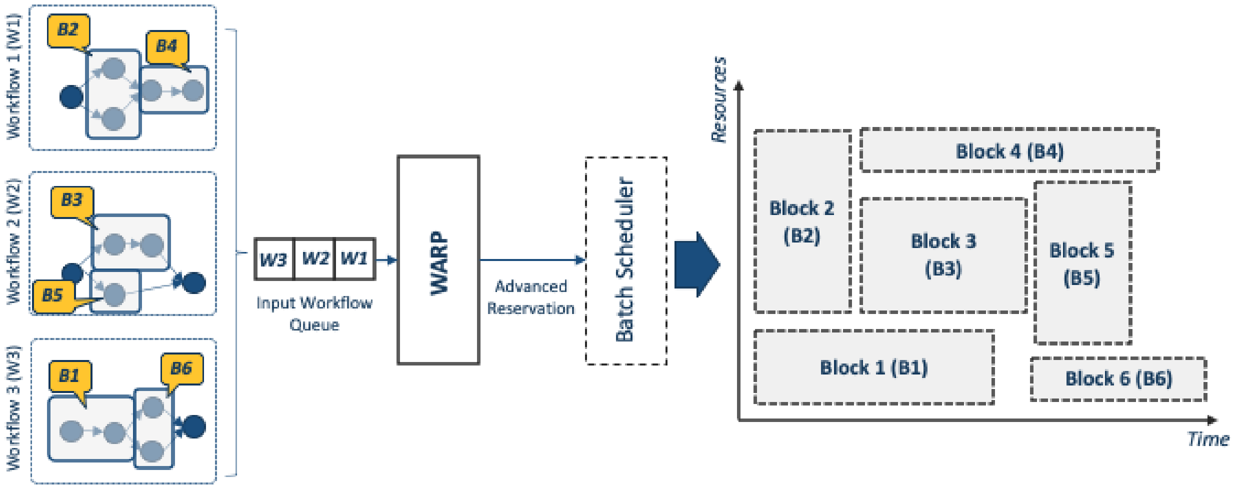


Figure 20 Example of workflows-aware resources reservation

Alien4Cloud and YORC provide the toolbox for managing all the steps for deploying the execution environment: specifically installing HyperQueue system on the HPC resources and on the Cloud resources, as well as (if required) spinning up virtual machines, containers, etc. Alien4Cloud leverages on a standardized approach (which is based on the TOSCA standard) to describe how the environment is made: a catalogue contains a set of 'software' components which are used to describe environmental building blocks (abstraction of the concept of a computational node, a virtual machine, a container, a specific software that has to run on a container or virtual machine, etc.); then, installation relationships and dependencies are also included. The catalogue also contains templates featuring the operations for correctly manage the installation, launch, stopping, and removal of such components. In this regard, Alien4Cloud exposes the functionalities that are needed to model such software components and the associated templates, as well as the interface with YORC, which is the deployment engine.

Secure access to the HPC/Cloud resources is of primary importance in modern HPC centers, since supercomputers are becoming part of larger cyber infrastructures, which make them more sensitive to attacks and data breaches. The HEAppE middleware is considered as part of the ACROSS orchestration solution, as it provides secure access to HPC (also including Cloud) resources through an HPC-as-a-Service (HPCaaS) service. To this end, HEAppE allows decoupling the set of users that own workflows and request execution resources, from the actual set of system accounts that perform jobs submission on the HPC clusters. HEAppE provides a uniform interface to interact with the HPC clusters, along with a straightforward integration with HyperQueue. This latter is a system designed to overcoming the limitation imposed by batch schedulers; for instance, the limited capability of ingesting a huge number of jobs in a relatively short amount of time [1], [2], the coarse-grain management of resources (meaning that resources within a node cannot be sliced among different tasks), the limited communication between running jobs and external software components, to mention a few. To this end, HyperQueue introduces a master-workers fine-grain management system. The master component runs on the login node(s) of the cluster and provide a flexible front-end to interact with. Specifically, it replicates the queuing mechanism as the main batch scheduler, but it also allows direct communication with worker agents running on reserved resources, without any specific limitations. Nicely, HyperQueue worker agents can be deployed on virtualized resources (Cloud), thus providing a unified system for controlling the underlying execution infrastructure. StreamFlow is a Workflow Management System (WMS) written in Python 3 and based on the Common Workflow Language (CWL) standard. It has been designed to support the concurrent execution of multiple communicating jobs (or even tasks) in a multi-agent ecosystem and to allow for the execution of hybrid workflows on top of Cloud/HPC infrastructures, with an execution model inspired by the dataflow model. StreamFlow provides the capability of executing (hybrid) workflows that can be conveniently expressed as DAGs, by matching the required resources for each vertex of the DAG (i.e., jobs). StreamFlow relies on an engine to support the execution of these kind workflows; as such, it leverages on the description of the workflow-DAG on one side, and the description of the execution environment on the other hand. Like the WARP module, the internal scheduling strategy of StreamFlow will benefit from the integration of infrastructural monitoring data.

Besides software components described here, pilot specific software will be used to govern the execution of the associated workflows. Such components are tailored for the specific needs of the pilot use cases and used to improve the workflow execution in cases like ensemble simulations (e.g., ERT), workload scheduler (e.g., Kronos) and in-situ/in-transit analysis (e.g., DAMARIS). With the aims of keeping flexibility and effectiveness of the solution, lightweight virtualization technologies, with a focus on containerization (e.g., Singularity, Docker, PodMan, etc.), will be investigated on both the HPC and Cloud also as a mean of simplifying the setup phase of the execution environments in those cases where less traditional hardware accelerators will be used (Neural Network Processors –e.g., Habana Gaudi/Goya, Neuromorphic devices, FPGAs).

7 Conclusions

The goal of D2.2 is to describe Milestone 2, ACROSS Key technological and Platform specification, which means:

- establishing a mapping of the requirements and constraints from pilots provided in Milestone 1 (D2.1) to some identified technologies;
- assemble the identified technologies in platforms supporting the ACROSS objectives.

The document first presents the mapping between pilots use cases and required SW and HW technologies. In this part, each pilot use case describes the improvements of their workflows, which are expected through the project, and the technologies that will be used to achieve them.

The initial global project architecture is presented as a reminder, as the two main layers, the higher-level software part and the lower-level hardware parts, respectively handled by WP4 and WP3, will be detailed later on in this document, as a result of Milestone 2. Indeed, the selected technologies and their assembly, result in a more detailed specification of these layers.

Then the key HW technologies and platforms are presented. Afterwhile the key software technologies that will be used are listed and described. Among these software technologies, a subset is used to build the ACROSS orchestration platform, that will provide an Exascale-ready execution framework for hybrid ML/BD/HPC workflows. The early phase architecture of this orchestration platform is depicted, although at this stage of the project it is not finalized and some design choices still to be validated.

The document presents the current status of ACROSS HW and SW platforms at this time of the project, it is not a final and completed design. Some choices are still to be validated, and along with the implementation phase, some new constraints may conduct to alternative or complementary design and technology choices.

REFERENCES

[1]	FLUX framework, Lawrence Livermore National Laboratory: https://computing.llnl.gov/projects/flux-building-framework-resource-management
[2]	Ahn, Dong H., et al. "Flux: Overcoming scheduling challenges for exascale workflows." *Future Generation Computer Systems* 110 (2020): 202-213
[3]	SLURM Advanced Resource Reservation Guide: https://slurm.schedmd.com/reservations.html

APPENDIX

The ACROSS project involves the integration of a large set of technologies which are connected to both the acceleration of application codes and the management of the workflows, and so covering the whole hardware and software stack. On the other hand, the integration of such technologies into a coherent and effective execution platform implies dealing with a large and rich set of entities and terms, which could have different meanings depending on which domain (HPC, Cloud, Orchestration, Pilot specific domains) they are considered in. As such, the reader may be confused, with the risk of not catching the right meaning of the terms and entities referred in this document. To this end, hereafter, we provide a description of the common terms used in the document and more generally referred by the ACROSS platform, as well as their relationships.

To summarize, the purpose of this appendix is twofold:

- Allowing the description of the Pilot application workflows in terms of (a hierarchy of) *elementary steps* (with a top/down approach).
- Defining the relationship among these elementary steps.

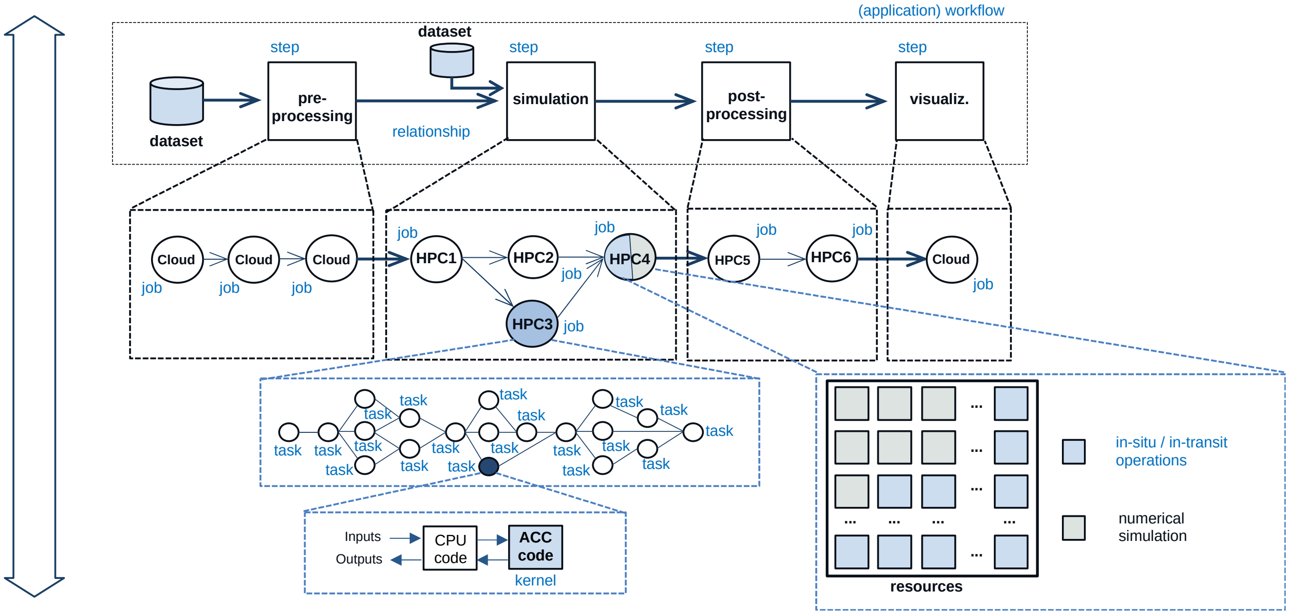


Figure 21 Hierarchical representation of a generic ACROSS application.

Figure 21 provides a graphical representation of the hierarchical organization of a generic ACROSS application. As such, in ACROSS we are going to consider an application as a composition of **steps**, which have relationships between one another. A relationship expresses a dependency among two-step (i.e., data dependency, temporal execution dependency). A workflow step defines a coarse (set of) operation(s) required to achieve the final goal of the application (e.g., completing the simulation/design of a complex aeronautic system, climate/weather modelling, etc.). Application steps and their relationships allow to define an **application workflow** (or simply **workflow**), which generally resemble the structure of a directed graph (i.e., nodes are the blocks and the edges are the relationships). Each step, at this level, may be (hierarchically) composed of a set of **jobs**, which have relationships among one another (i.e., data dependencies, execution dependencies). Jobs may require different execution resources, so we can have *HPC-jobs* (managed through batch-schedulers –e.g., PBS, SLURM), and *Cloud-Jobs* that can be run within a VM or a container.

Jobs may involve the execution of several fine-grained operations; we can refer to these operations as **tasks** (or in-job tasks). A given workflow may involve the analysis and/or visualization of data associated to a running numerical simulation as the simulation progresses. To this end, **in-situ operations** and **in-transit operations** can be executed. As mentioned, we can have two distinct operations: *visualization* and *analytic*:

- *In-situ* operations: operations share the resources with the numerical simulation in two possible modes:
    - Time sharing, where the same CPU cores process both the numerical simulation and the visualization/analytics.
    - Space sharing, where CPU cores in a given node are split in two sets: one set for numerical simulation, and the other one for visualization/analytics.
- *In-transit* operations: operations use separated resources (i.e. also time sharing) from those used by the numerical simulation (e.g., a group of nodes is exclusively used by numerical simulation, another group is used for visualization/analytic).

Tasks, generally organized as DAGs, are managed by low-level schedulers, which schedule their execution on the resources assigned by the batch schedulers or the Cloud system. Whenever a task contains piece of code that can be accelerated on specialized hardware (–e.g., GPUs, FPGAs, NNPs, etc.), we refer to this code as an acceleration kernel or simply **kernel**. Kernels have a clear input/output interface with the remainder code of the task that runs on a CPU.

The definition of an application workflow (i.e., the set of steps, jobs and their relationships) also implies the definition of the set of resources required to execute it. To this end, the **execution environment description** can be associated to the application workflow. It expresses the set of **components** (both software and hardware) and their relationships. It can be accompanied by a set of **operative sequences**, which defines the sequence of operations to be performed in order to *install* the components, *start/stop* the components, *execute* software components, etc.