SC22 Dallas, TX | hpc accelerates.

Making easier the development and deployment of application workflows with eFlows4HPC

Rosa M Badia, Barcelona Supercomputing Center

17th Workshop on Workflows in Support of Large-Scale Science, November 14, 2022, Dallas, TX, USA

eFlows4HPC

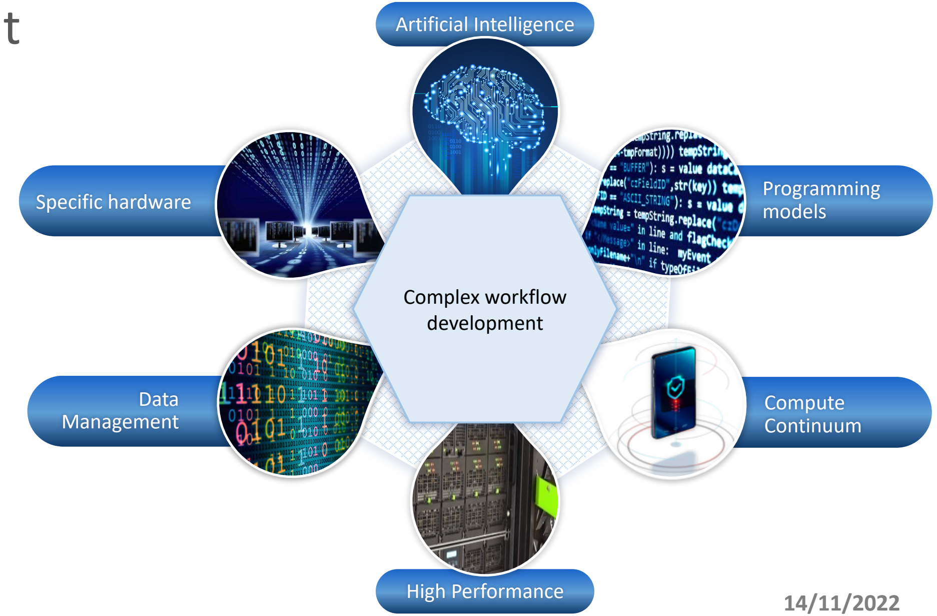
This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955558. The JU receives support from the European Union's Horizon 2020 research and innovation programme and Spain, Germany, France, Italy, Poland, Switzerland, Norway.

Complex workflows and complex infrastructures

- EuroHPC aims at developing a World Class Supercomputing Ecosystem in Europe
  - Procuring and deploying pre-exascale and petascale systems in Europe
- These systems will be capable of running large and complex applications
- Applications demand the composition of HPC, artificial intelligence and data analytics
- EuroHPC also funds software development projects:
  - eFlows4HPC

Main objectives

- Software stack that make easier the development of workflows
  - HPC, AI + data analytics
  - Reactive and dynamic workflows
  - Efficient resource management
- HPC Workflows as a Service:
  - Mechanisms to make it easier the use and reuse of HPC by wider communities

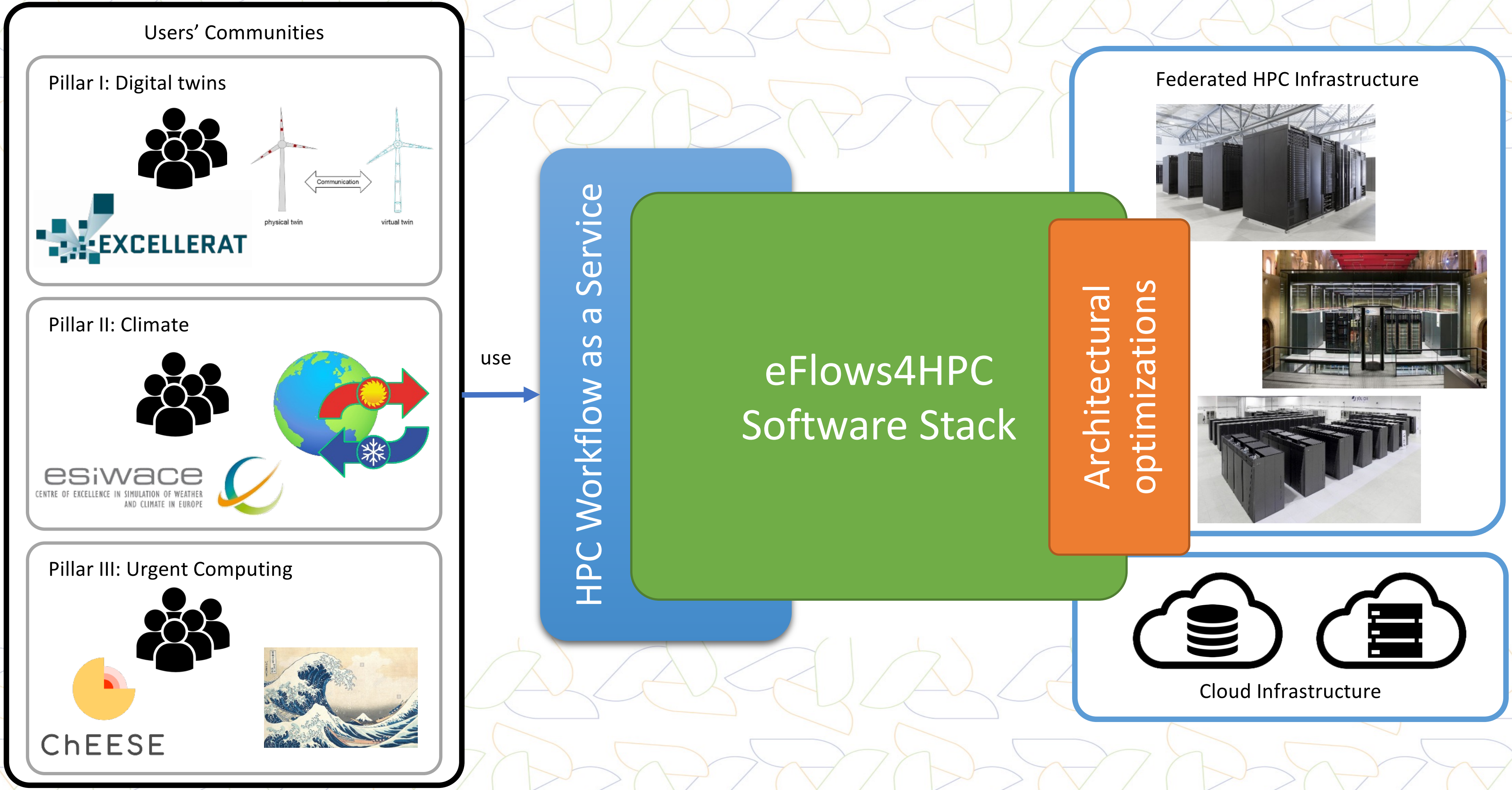


Outline

eFlows4HPC

- Project architecture
- Pillar applications
- HPC Workflows as a Service

PROJECT ARCHITECTURE



Users' Communities

Pillar I: Digital twins

Communication
physical twin
virtual twin
EXCELLERAT

Pillar II: Climate

esiwace
CENTRE OF EXCELLENCE IN SIMULATION OF WEATHER AND CLIMATE IN EUROPE

Pillar III: Urgent Computing

use

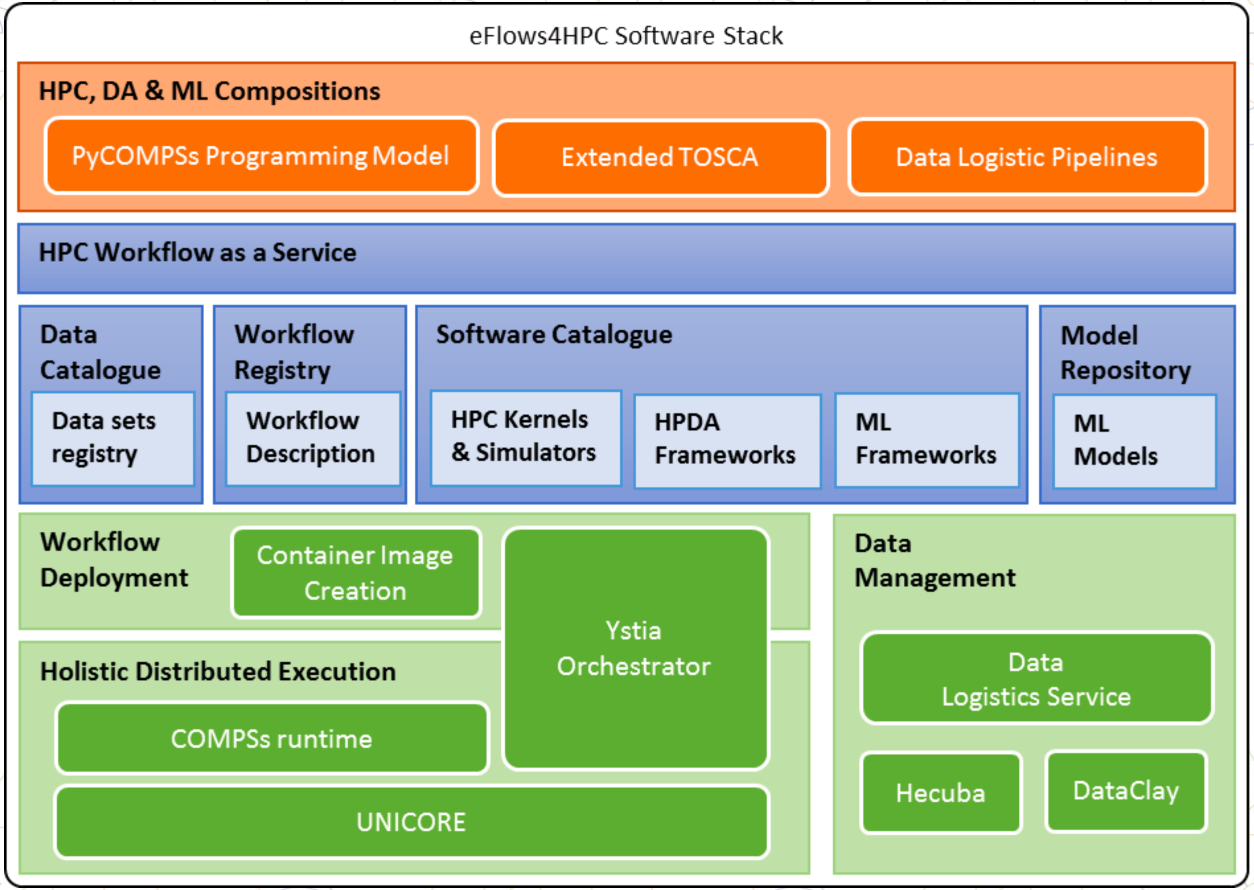
HPC Workflow as a Service

eFlows4HPC Software Stack

Architectural optimizations

Federated HPC Infrastructure

Cloud Infrastructure



eFlows4HPC Software Stack

HPC, DA & ML Compositions

PyCOMPSs Programming Model

Extended TOSCA

Data Logistic Pipelines

HPC Workflow as a Service

Data Catalogue

Data sets registry

Workflow Registry

Workflow Description

Software Catalogue

HPC Kernels & Simulators

HPDA Frameworks

ML Frameworks

Model Repository

ML Models

Workflow Deployment

Container Image Creation

Ystia Orchestrator

Holistic Distributed Execution

COMPSs runtime

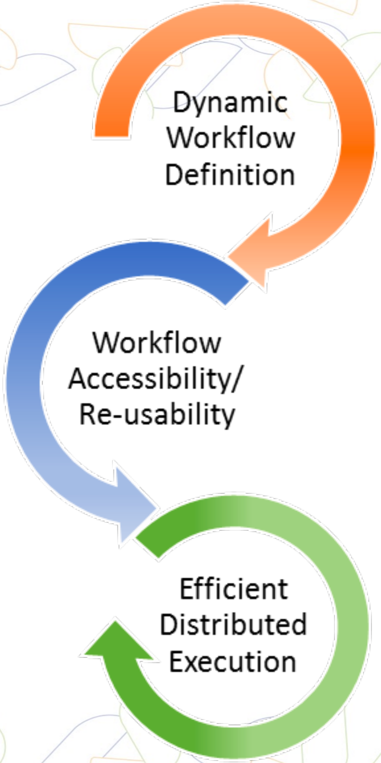
UNICORE

Data Management

Data Logistics Service

Hecuba

DataClay



Dynamic Workflow Definition

Workflow Accessibility/ Re-usability

Efficient Distributed Execution

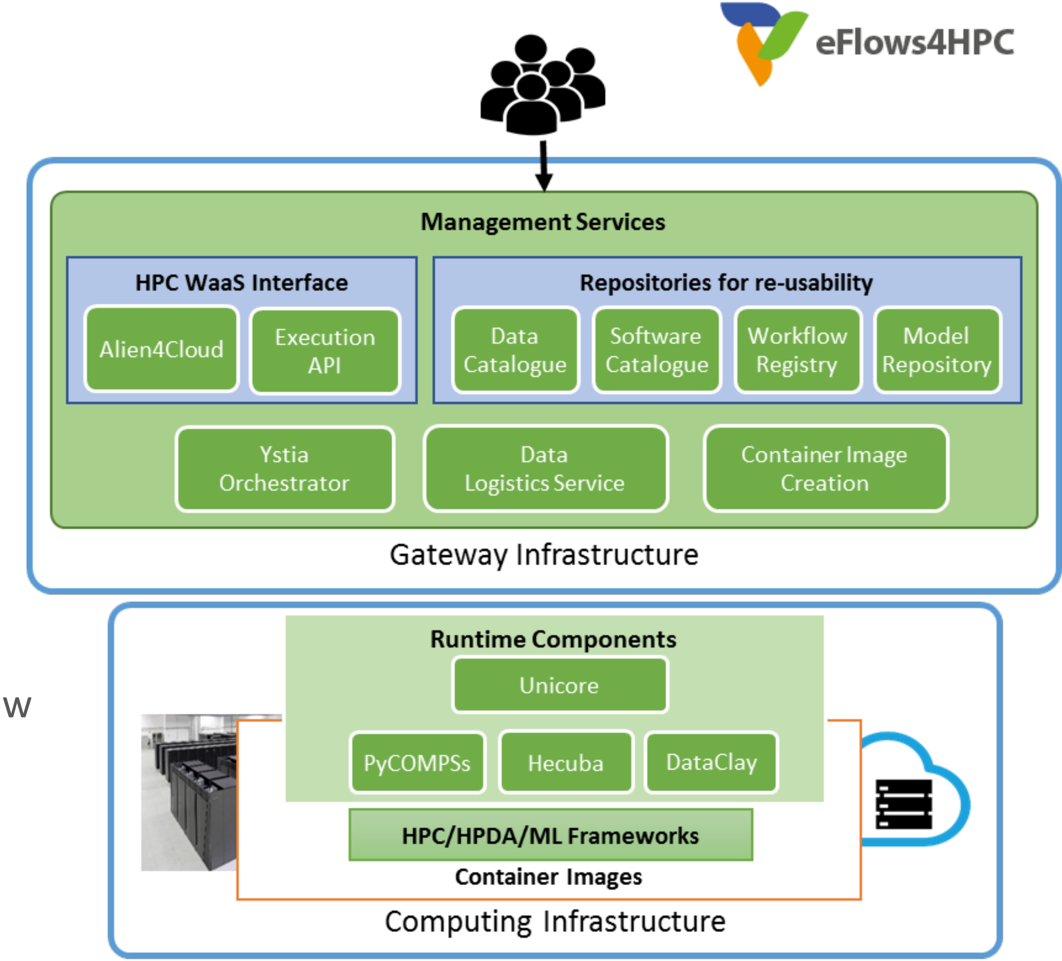
Software stack deployment

Gateway services

- Components deployed outside the computing infrastructure.
- Managing external interactions and workflow lifecycle

HPC and runtime Components

- Deployed inside the computing infrastructure to manage the workflow execution

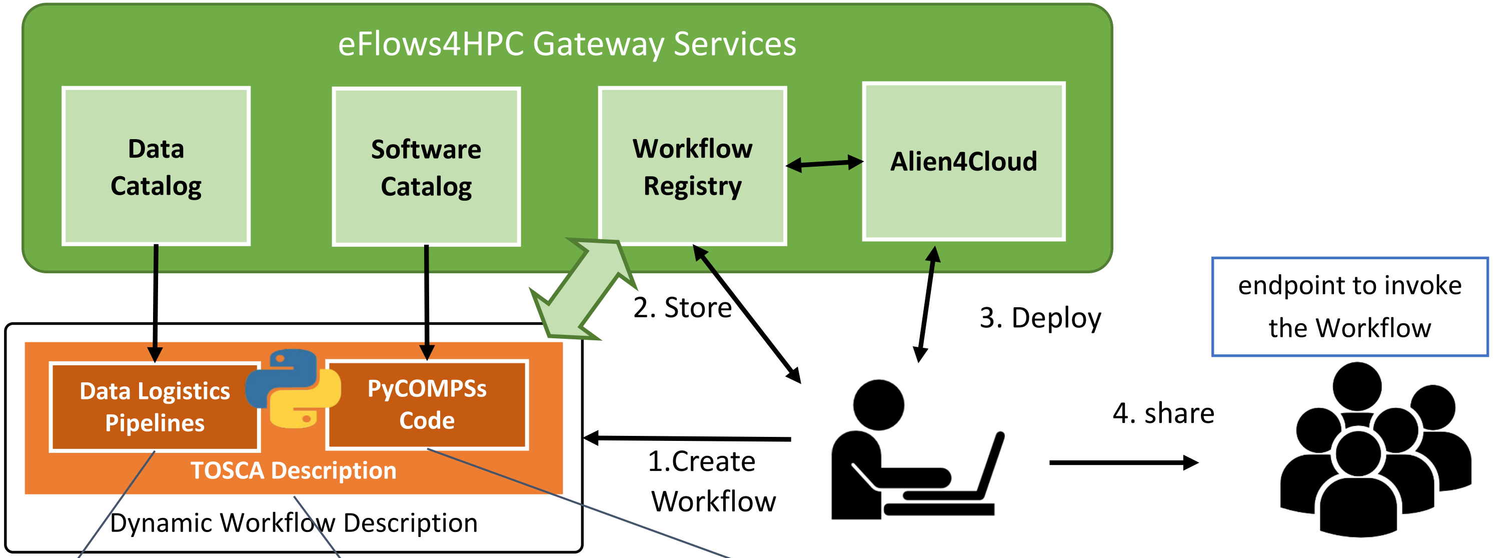


HPC WORKFLOWS AS A SERVICE

HPC Workflows as a Service

- Methodology split in four steps
  - Development
  - Deployment
  - Credential management
  - Execution

Workflow development overview



Description of data movements as python functions.
Input/output datasets described at Data Catalog

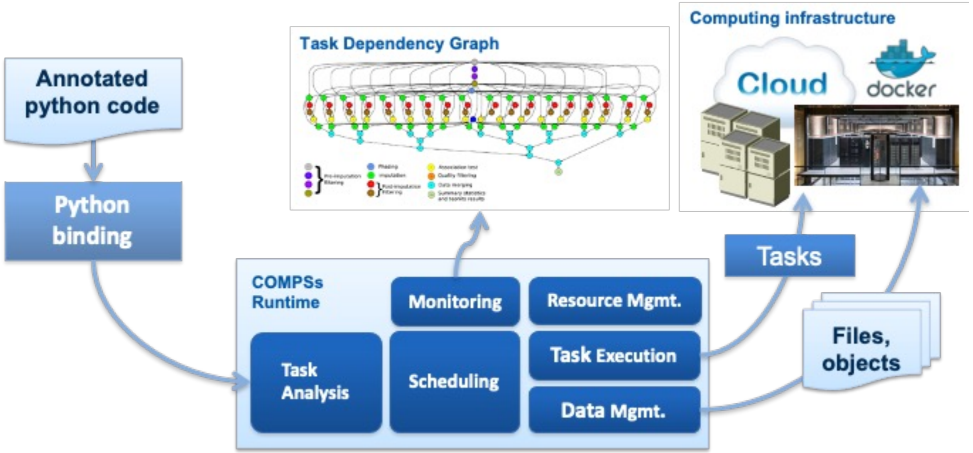
Computational Workflow as a simple python script.
Invocation of software described in the Software Catalog

Topology of the components involved in the workflow lifecycle and their relationship.

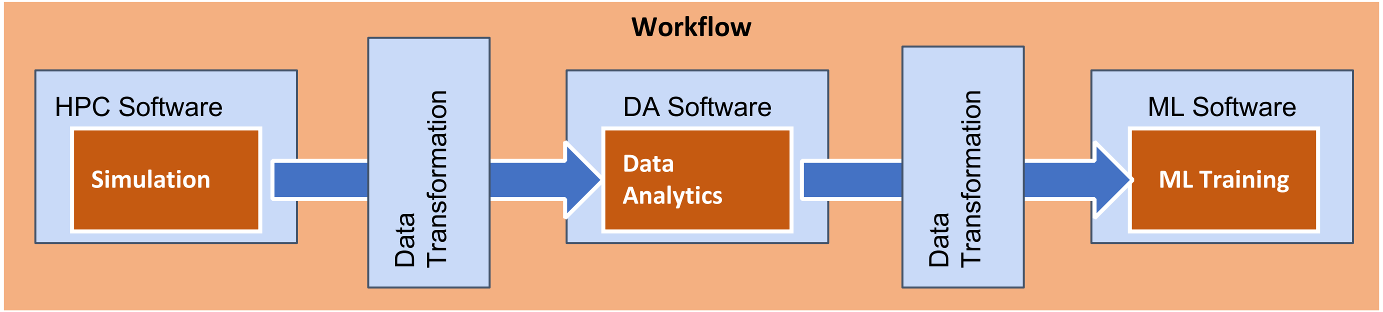
Main element: Workflows in PyCOMPSs

- Sequential programming, parallel execution
- General purpose programming language + annotations/hints
  - To identify tasks and directionality of data
- Task graph built at runtime
- Tasks can be sequential and parallel
  - threaded or MPI
- Offers to applications the illusion of a shared memory in a distributed system
- Agnostic of computing platform: clusters, clouds, containers
- Supported by runtime that performs all scheduling decisions and data management

```
@task(c=INOUT)
def multiply(a, b, c):
        c += a*b
```



Interfaces to integrate HPC/DA/ML



- Goal:
  - Reduce the required glue code to invoke multiple complex software steps
  - Developer can focus in the functionality, not in the integration
  - Enables reusability
- Two paradigms:
  - Software invocation
  - Data transformations

```
#workflow steps defined as tasks
@data_transformation (input_data, transformation description)
@software (invocation description)
def data_analytics (input_data, result):
    pass

#workflow body
simulation (input_cfg, sim_out)
data_analytics (sim_out, analysis_result)
ml_training (analysis_result, ml_model)
```

Data Catalogue and Data Logistics Service

Data Catalogue:

- Lists datasets used and created by the workflow according to FAIR principles
- Provides metadata to make data movement pipelines more generic

Data Pipelines:

- Formalization of data movements for transparency and reusability
- Stage-in/out, image transfer

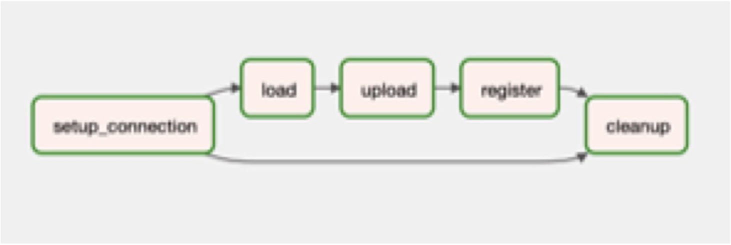
Data Logistics Services (DLS):

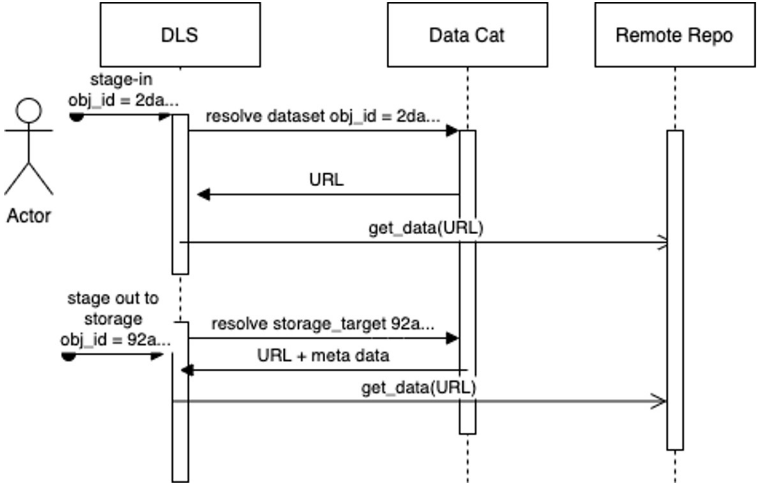
- Performs the execution of data pipelines at deployment and execution time

Production Ready Services:

- https://datacatalogue.eflows4hpc.eu
- https://datalogistics.eflows4hpc.eu/

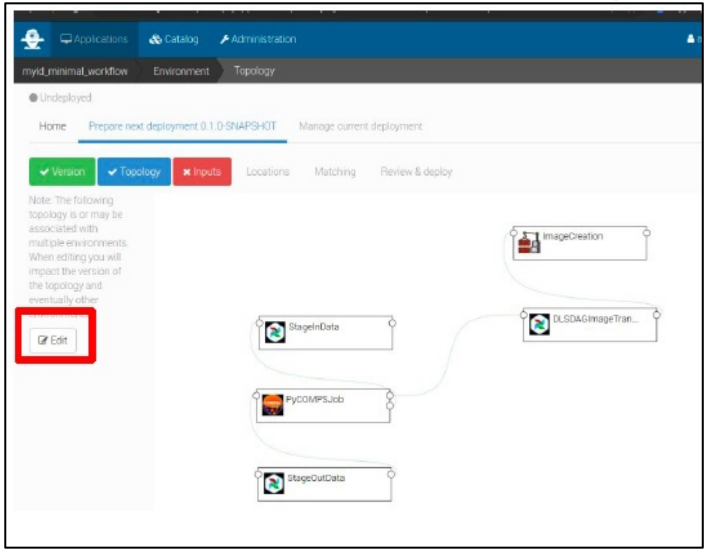
Data pipeline





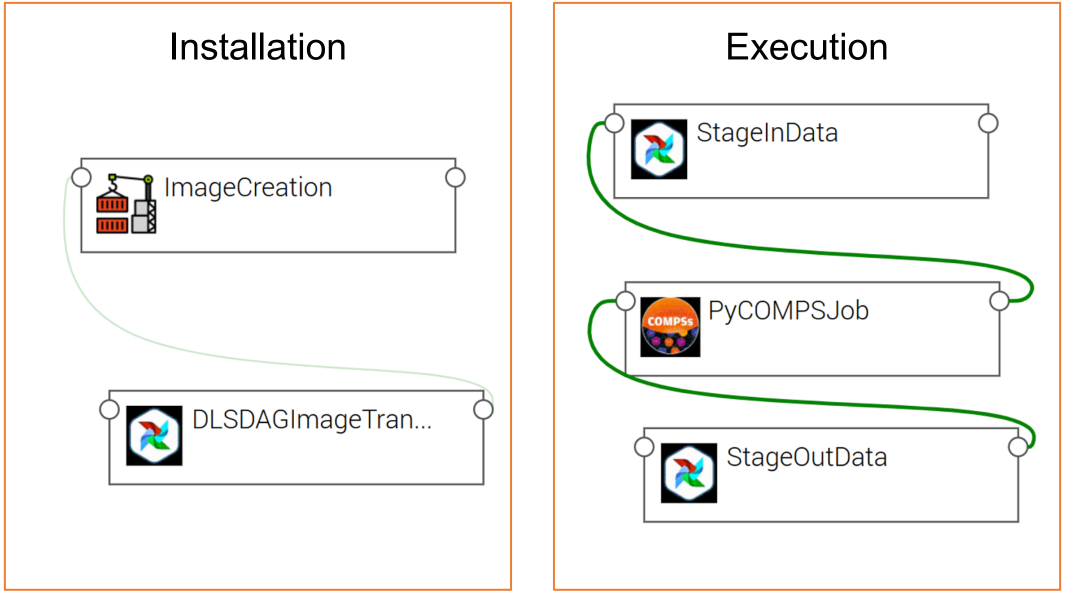
TOSCA Modelization

Alien4cloud portal



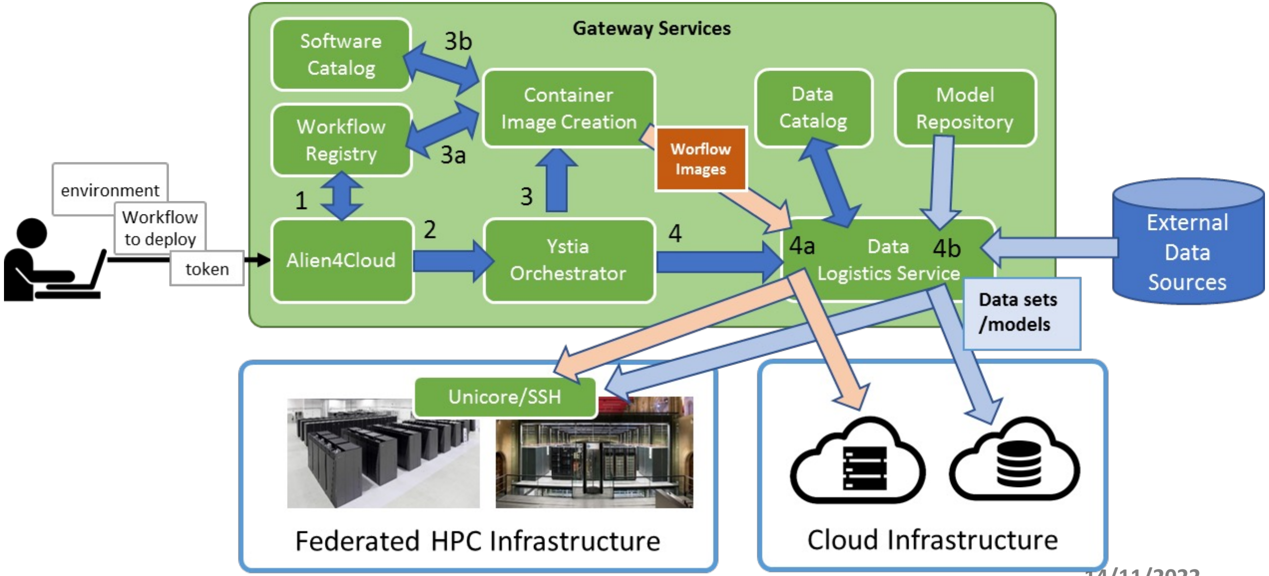
Note: The following topology is or may be associated with multiple environments. When editing you will impact the version of the topology and eventually other

Topology of the different components involved in the Workflow lifecycle



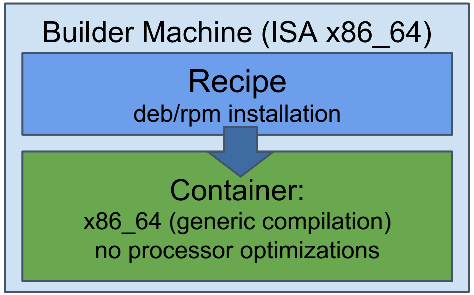
Deployment

- Deployment orchestrated by Ystia Orchestrator (Yorc)
- Workflow information retrieved from registry
- Deployment of workflow components in the computing infrastructures
  - HPC containers built with easybuild/Spack
- Data Logistic Service
  - Workflow images
  - Data stage-in and stage-out
  - Periodical transfers of data outside HPC systems



HPC Ready Containers

Standard container image creation

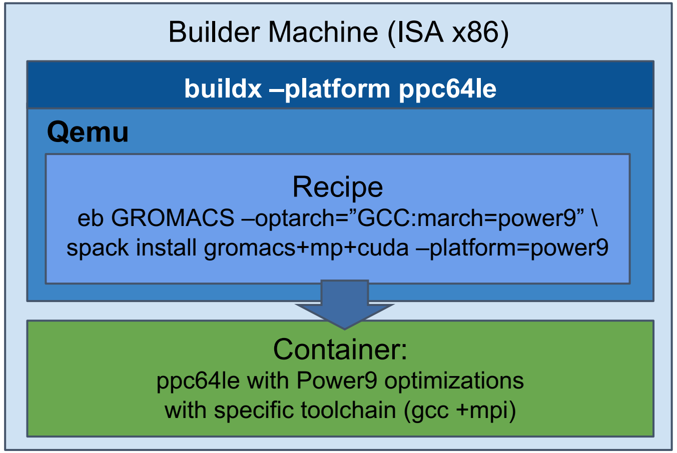


Builder Machine (ISA x86_64)

Recipe
deb/rpm installation

Container:
x86_64 (generic compilation)
no processor optimizations

eFlows4HPC approach



Builder Machine (ISA x86)

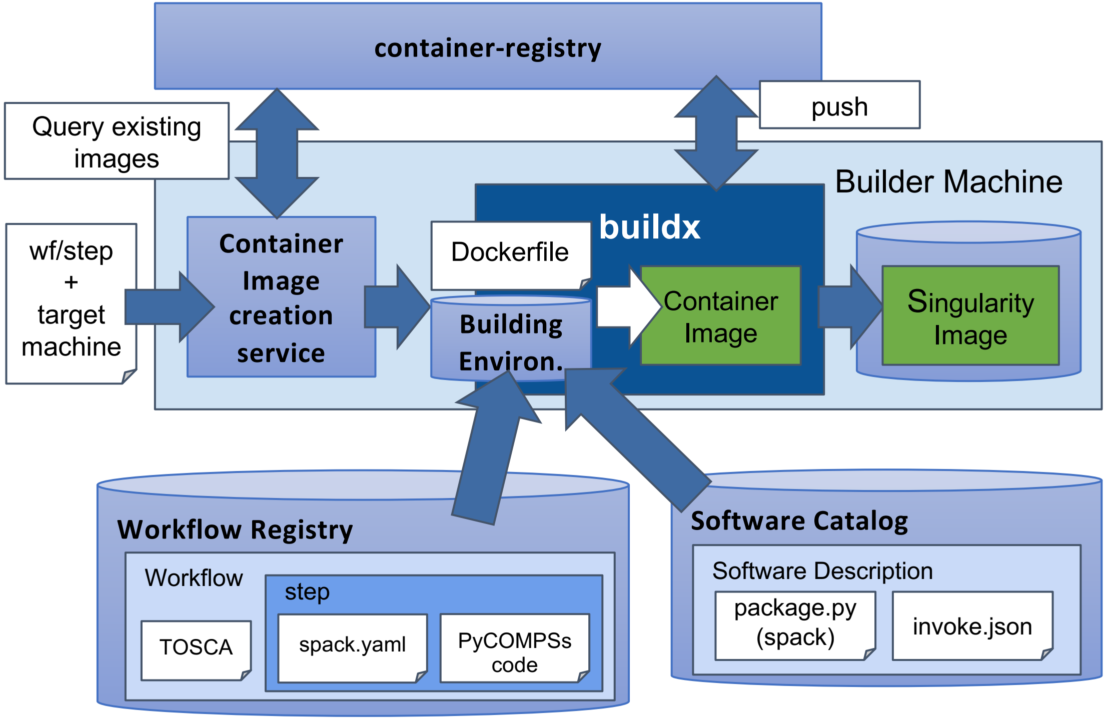
buildx –platform ppc64le

Qemu

Recipe
eb GROMACS –optarch="GCC:march=power9" \
spack install gromacs+mp+cuda –platform=power9

Container:
ppc64le with Power9 optimizations
with specific toolchain (gcc +mpi)

Service to automate the Container Image Creation



container-registry

Query existing images

push

Builder Machine

wf/step + target machine

Container Image creation service

Dockerfile

Building Environ.

buildx

Container Image

Singularity Image

Workflow Registry

Workflow

TOSCA

step

spack.yaml

PyCOMPSs code

Software Catalog

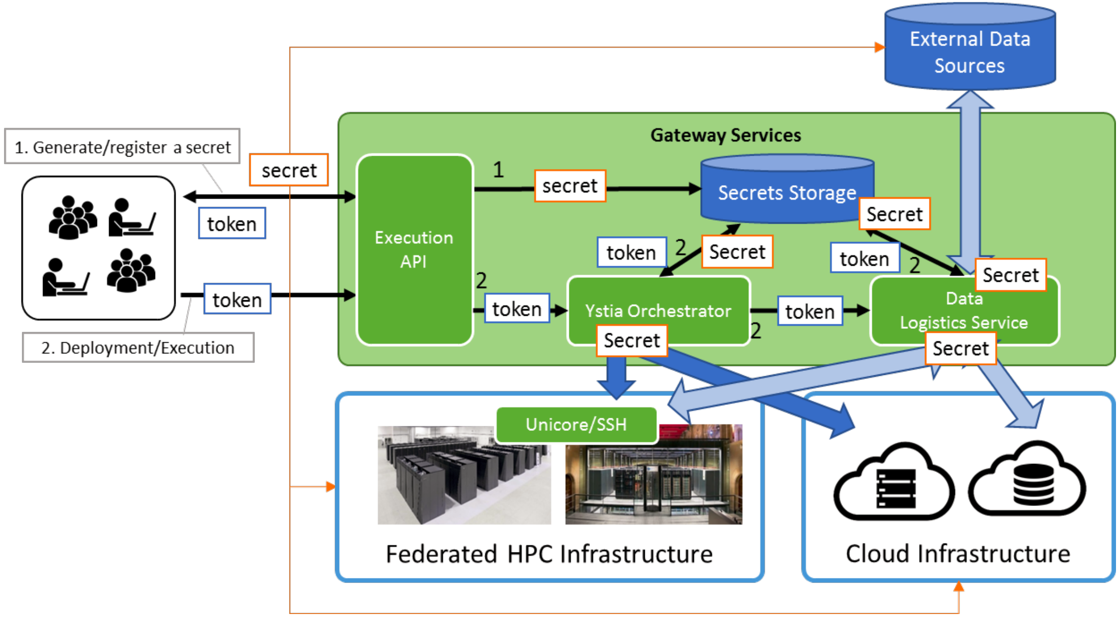
Software Description

package.py (spack)

invoke.json

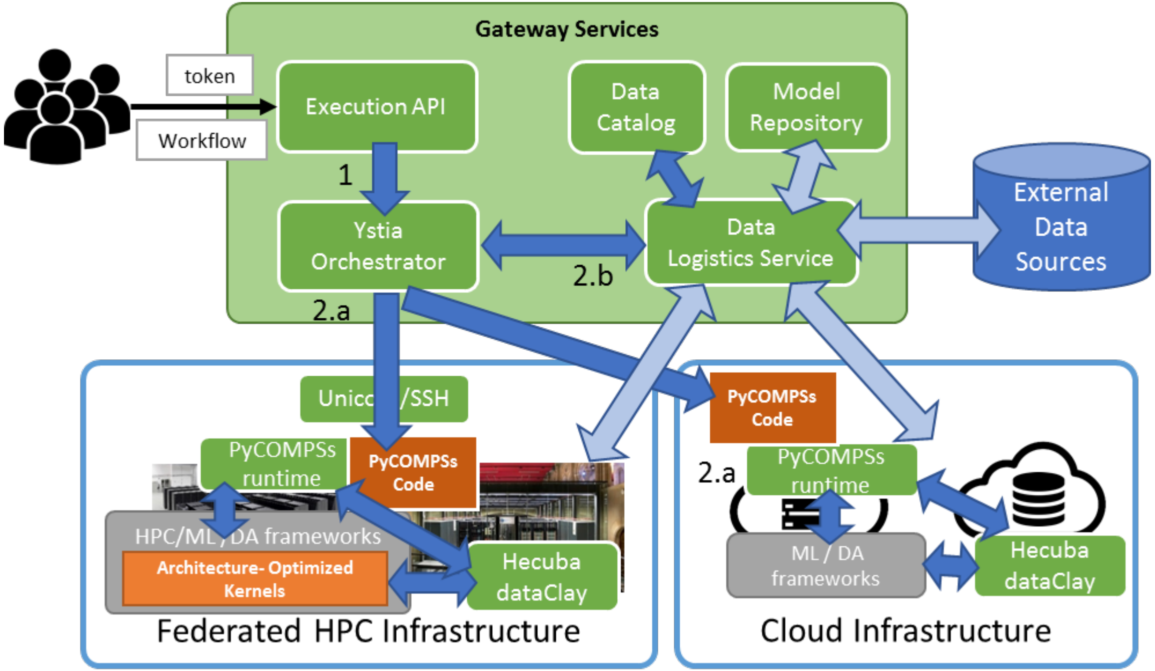
Credential management

- Prior to executing the workflows, users have to configure their access credentials
- Users' certificates managed by an Execution API
  - Provides a few methods to register and access credentials or generate a new secret
  - HashiCorp Vault for secret (SSH keys) management
- User authorizes adding credentials in the HPC cluster
- Credentials identified by a token attached to the user's workflow invocation.



Operation- Workflow Execution

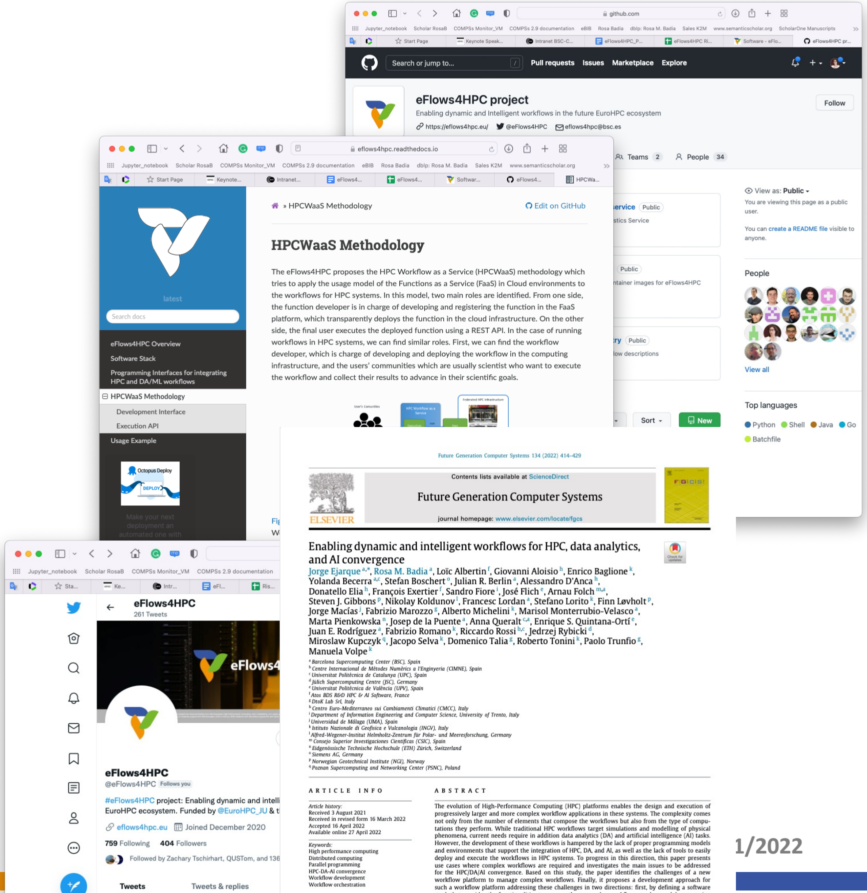
- Submission of the execution of the workflow processes to the HPC infrastructure
- PyCOMPSs orchestrates different task types
  - HPC (MPI), ML, DA
- Dynamic execution
  - Runtime task-graph
  - Task-level FT
  - Exceptions
- Data management
  - Persistent storage
- Optimized kernels
  - EPI, GPU, FPGA



Project main achievements

https://eflows4hpc.eu/software/

- Requirements and software architecture
- Definition and implementation of abstractions to support the integration of different stack components
- Design and development of a minimal workflow
- Design and first version of the HPCWaaS methodology
- Design and implementation of the Data Catalogue
- Design and implementation of first version of Pillars' workflows.
- First release of project software and documentation available
- Set of internal trainings about software stack components and HPCWaaS
- Good visibility: articles, keynote presentations, media



eFlows4HPC project

Enabling dynamic and Intelligent workflows in the future EuroHPC ecosystem

https://eflows4hpc.eu/ @eFlows4HPC eflows4hpc@bsc.es

You are viewing this page as a public user.

You can create a README file visible to anyone.

» HPCWaaS Methodology — Edit on GitHub

HPCWaaS Methodology

The eFlows4HPC proposes the HPC Workflow as a Service (HPCWaaS) methodology which tries to apply the usage model of the Functions as a Service (FaaS) in Cloud environments to the workflows for HPC systems. In this model, two main roles are identified. From one side, the function developer is in charge of developing and registering the function in the FaaS platform, which transparently deploys the function in the cloud infrastructure. On the other side, the final user executes the deployed function using a REST API. In the case of running workflows in HPC systems, we can find similar roles. First, we can find the workflow developer, which is charge of developing and deploying the workflow in the computing infrastructure, and the users' communities which are usually scientist who want to execute the workflow and collect their results to advance in their scientific goals.

eFlows4HPC

@eFlows4HPC Follows you

#eFlows4HPC project: Enabling dynamic and intelli
EuroHPC ecosystem. Funded by @EuroHPC_JU & t

eflows4hpc.eu Joined December 2020

759 Following 404 Followers

Followed by Zachary Tschirhart, QUSTom, and 136

Future Generation Computer Systems 134 (2022) 414–429

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

ELSEVIER

Enabling dynamic and intelligent workflows for HPC, data analytics, and AI convergence

Jorge Ejarque [a,*], Rosa M. Badia [a], Loïc Albertin [f], Giovanni Aloisio [h], Enrico Baglione [k], Yolanda Becerra [a,c], Stefan Boschert [o], Julian R. Berlin [a], Alessandro D'Anca [h], Donatello Elia [h], François Exertier [f], Sandro Fiore [i], José Flich [e], Arnau Folch [m,a], Steven J. Gibbons [p], Nikolay Koldunov [l], Francesc Lordan [a], Stefano Lorito [k], Finn Løvholt [p], Jorge Macías [j], Fabrizio Marozzo [g], Alberto Michelini [k], Marisol Monterrubio-Velasco [a], Marta Pienkowska [n], Josep de la Puente [a], Anna Queralt [c,a], Enrique S. Quintana-Ortí [e], Juan E. Rodríguez [a], Fabrizio Romano [k], Riccardo Rossi [b,c], Jedrzej Rybicki [d], Miroslaw Kupczyk [q], Jacopo Selva [k], Domenico Talia [g], Roberto Tonini [k], Paolo Trunfio [g], Manuela Volpe [k]

[a] Barcelona Supercomputing Center (BSC), Spain
[b] Centre Internacional de Mètodes Numèrics a l'Enginyeria (CIMNE), Spain
[c] Universitat Politècnica de Catalunya (UPC), Spain
[d] Jülich Supercomputing Centre (JSC), Germany
[e] Universitat Politècnica de València (UPV), Spain
[f] Atos BDS R&D HPC & AI Software, France
[g] DtoK Lab Srl, Italy
[h] Centro Euro-Mediterraneo sui Cambiamenti Climatici (CMCC), Italy
[i] Department of Information Engineering and Computer Science, University of Trento, Italy
[j] Universidad de Málaga (UMA), Spain
[k] Istituto Nazionale di Geofisica e Vulcanologia (INGV), Italy
[l] Alfred-Wegener-Institut Helmholtz-Zentrum für Polar- und Meeresforschung, Germany
[m] Consejo Superior Investigaciones Cientificas (CSIC), Spain
[n] Eidgenössische Technische Hochschule (ETH) Zürich, Switzerland
[o] Siemens AG, Germany
[p] Norwegian Geotechnical Institute (NGI), Norway
[q] Poznan Supercomputing and Networking Center (PSNC), Poland

ARTICLE INFO

Article history:
Received 3 August 2021
Received in revised form 16 March 2022
Accepted 16 April 2022
Available online 27 April 2022

Keywords:
High performance computing
Distributed computing
Parallel programming
HPC-DA-AI convergence
Workflow development
Workflow orchestration

ABSTRACT

The evolution of High-Performance Computing (HPC) platforms enables the design and execution of progressively larger and more complex workflow applications in these systems. The complexity comes not only from the number of elements that compose the workflows but also from the type of computations they perform. While traditional HPC workflows target simulations and modelling of physical phenomena, current needs require in addition data analytics (DA) and artificial intelligence (AI) tasks. However, the development of these workflows is hampered by the lack of proper programming models and environments that support the integration of HPC, DA, and AI, as well as the lack of tools to easily deploy and execute the workflows in HPC systems. To progress in this direction, this paper presents use cases where complex workflows are required and investigates the main issues to be addressed for the HPC/DA/AI convergence. Based on this study, the paper identifies the challenges of a new workflow platform to manage complex workflows. Finally, it proposes a development approach for such a workflow platform addressing these challenges in two directions: first, by defining a software

Conclusions

- There is a need for providing tools for the development of complex workflows that include HPC modeling and simulation, artificial intelligence components and big data
- eFlows4HPC aims at providing a software stack that supports the development, deployment and execution of complex and dynamic workflows
- The HPCWaaS aims to provide a functionality similar for FaaS in cloud for complex workflows in HPC to make it easier the adoption of HPC technologies

Project partners

BSC Barcelona Supercomputing Center
Centro Nacional de Supercomputación

CIMNE R

JÜLICH
Forschungszentrum

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Bull
atos technologies

DtoK LAB

cmcc
Centro Euro-Mediterraneo sui Cambiamenti Climatici

Inria

SISSA
Scuola Internazionale Superiore di Studi Avanzati

INSTYTUT CHEMII BIOORGANICZNEJ
Polskiej Akademii Nauk

UNIVERSIDAD DE MÁLAGA

AWI ALFRED-WEGENER-INSTITUT
HELMHOLTZ-ZENTRUM FÜR POLAR- UND MEERESFORSCHUNG

ISTITUTO NAZIONALE DI GEOFISICA E VULCANOLOGIA

ETH zürich

SIEMENS

NGI

eFlows4HPC

Enabling dynamic and Intelligent workflows in the future EuroHPC ecosystem

www.eFlows4HPC.eu

@eFlows4HPC eFlows4HPC Project

This project has received funding from the European High-Performance Computing Joint Undertaking (JU) under grant agreement No 955558. The JU receives support from the European Union’s Horizon 2020 research and innovation programme and Spain, Germany, France, Italy, Poland, Switzerland, Norway.