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

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

- Project architecture
- Pillar applications
- HPC Workflows as a Service
PROJECT ARCHITECTURE
Users’ Communities

Pillar I: Digital twins

Pillar II: Climate

Pillar III: Urgent Computing

HPC Workflow as a Service

eFlows4HPC Software Stack

Architectural optimizations

Cloud Infrastructure

Federated HPC Infrastructure

Use
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 Description
- HPC Kernels & Simulators
- HPDA Frameworks
- ML Frameworks
- 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

1. Create Workflow
2. Store Computational Workflow as a simple python script. Input/output datasets described at Data Catalog
3. Deploy
4. share

-endpoint to invoke the Workflow

Data Catalog
Software Catalog
Workflow Registry
Alien4Cloud

Data Logistics Pipelines
PyCOMPSs Code

TOSCA Description
Dynamic Workflow Description

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

```python
@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/
TOSCA Modelization

Alien4cloud portal

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

Dockerfile

Container Image

Building Environ.

Builder Machine

Singularity Image

Workflow Registry

Workflow

step

TOSCA

spack.yaml

PyCOMPSs code

Software Catalog

Software Description

package.py (spack)

invoke.json

14/11/2022
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

• 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
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
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.

www.eFlows4HPC.eu

@eFlows4HPC eFlows4HPC Project