Optimising workflow lifecycle management: development, HPC-ready containers deployment and reproducibility

Raül Sirvent, Rosa M Badia

18/11/2024

Tutorial website

https://github.com/bsc-wdc/Tutorial_SC24
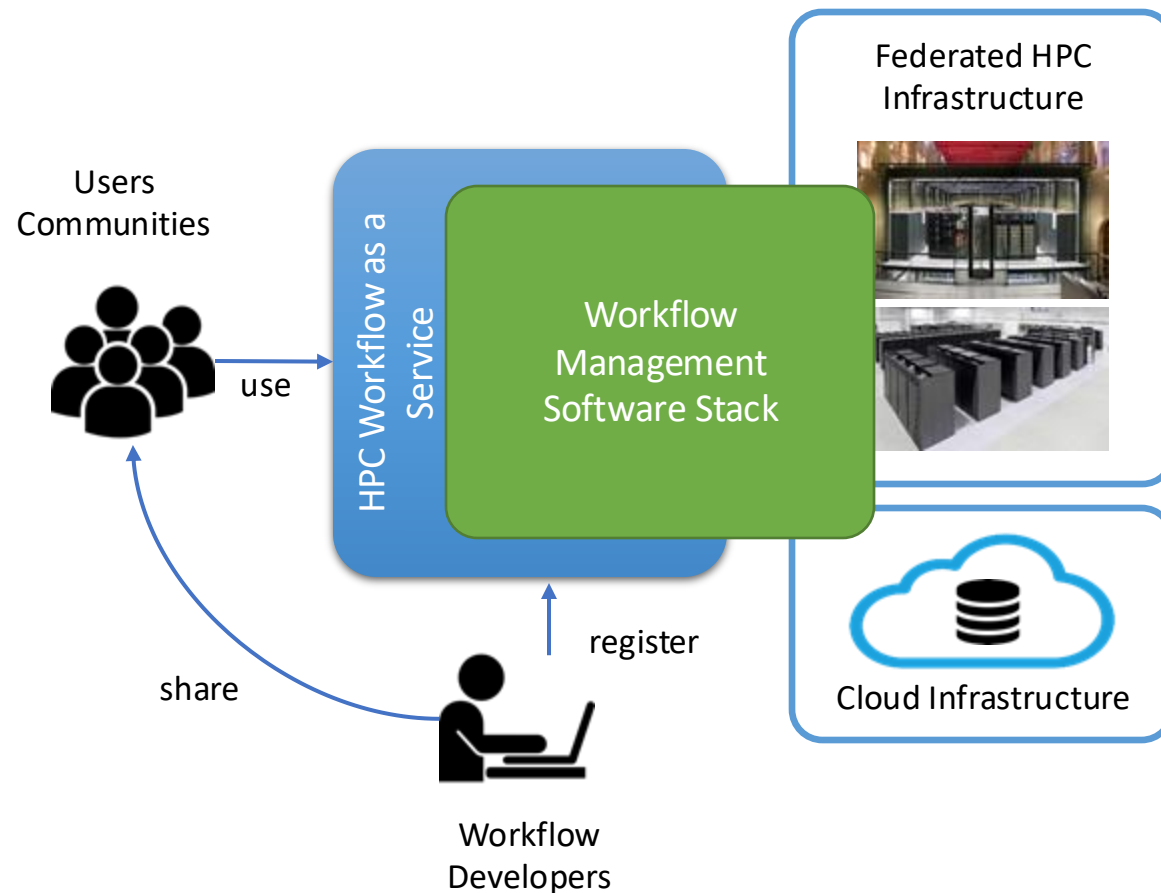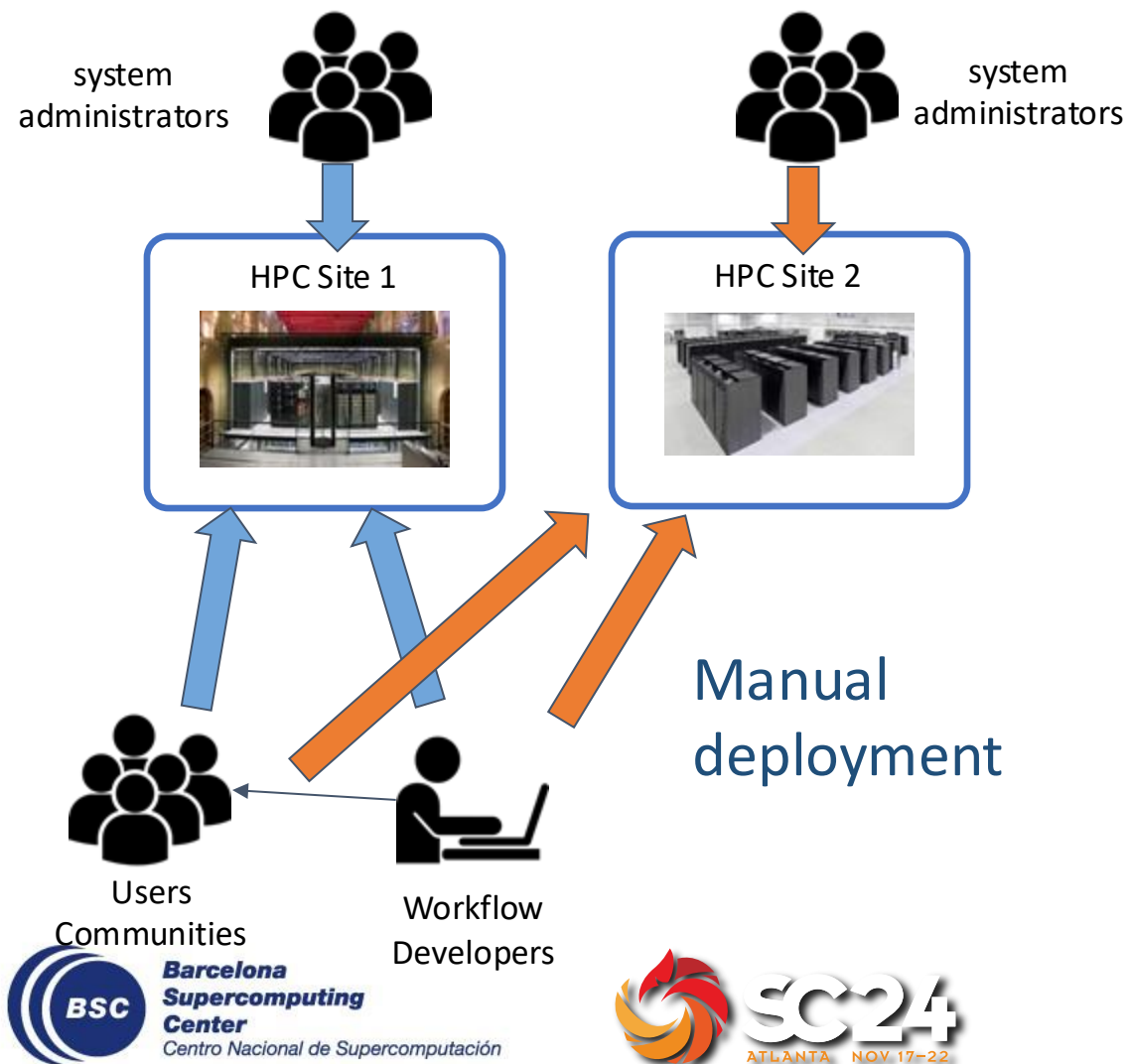
Agenda

8:30 – 8:45	Overview of tutorial agenda	Rosa M Badia
8:45 – 9:15	Part 1.1: Hybrid HPC+AI+DA workflow development with PyCOMPSs - Context of the workflows at BSC - Overview of workflow development with PyCOMPSs - Extensions for the integration of HPC with AI and DA - Sample workflows	Rosa M Badia
9:15 – 9:45	Part 1.2: Workflows' reproducibility through provenance - Motivation for workflow provenance - Design of the recording mechanism - Sharing experiments for reproducibility	Raül Sirvent
9:45 - 10:00	Part 1.3: HPC ready container images - Motivation for architecture specific containers - Overview of the Container Image Creation service - Example of HPC ready container generation	Rosa M Badia
10:00 - 10:30	Coffee break	

BSC
Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

SC24
ATLANTA, NOV. 17–22

Agenda

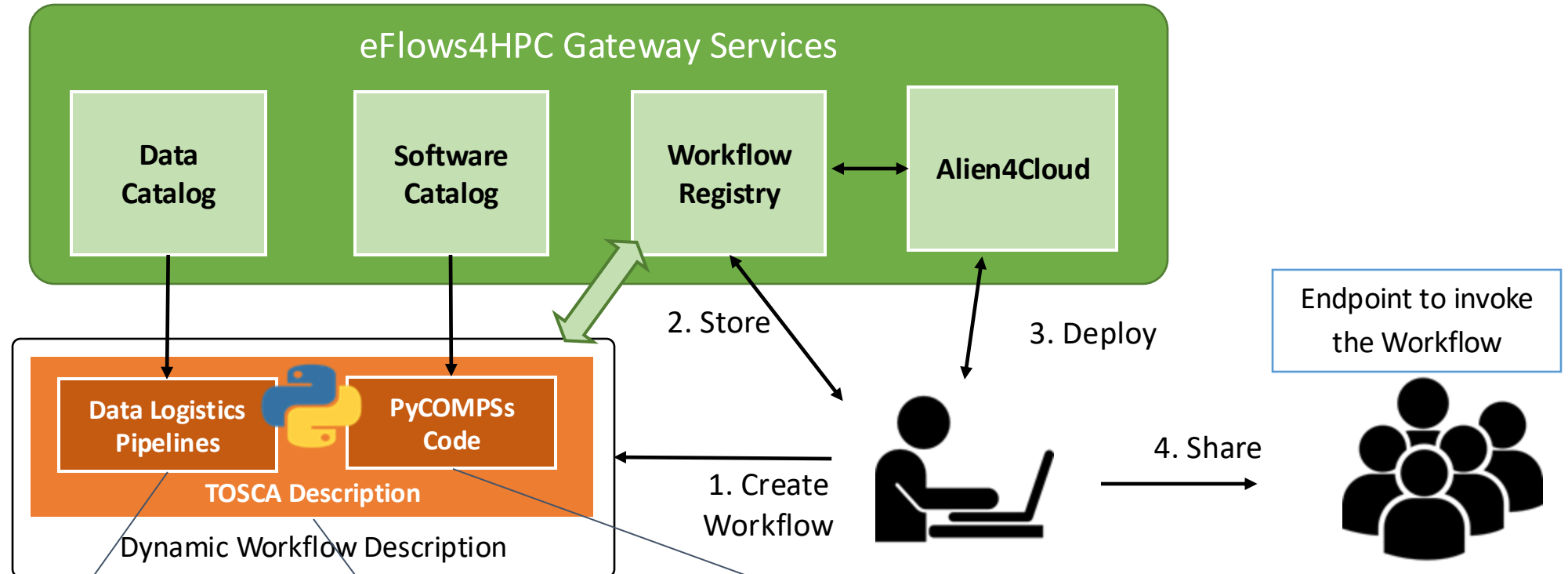|		
10:30 – 10:45	Hands-on preparation (credentials distribution, how to access, etc)	All presenters
10:45 – 11:15	Part 2.1: Hands-on session: Sample workflows with PyCOMPSs, execution with containers, task-graph generation, tracefile generation (optional)	Rosa M Badia
11:15 – 11:55	Part 2.2: Hands-on session: How to automatically record workflow provenance and use it to share experiments in WorkflowHub	Raül Sirvent
11:55 - 12:00	Tutorial conclusions	All presenters

Deployment in HPC Environments

Current approach

system administrators

system administrators

HPC Site 1

HPC Site 2

Manual deployment

Users Communities

Workflow Developers

Users Communities

use

HPC Workflow as a Service

Workflow Management Software Stack

Federated HPC Infrastructure

register

share

Workflow Developers

Cloud Infrastructure

Can we design something like FaaS for Complex Workflows for HPC?

BSC *Barcelona Supercomputing Center* Centro Nacional de Supercomputación

SC24 ATLANTA NOV 17-22

HPCWaaS: Workflow lifecycle overview

eFlows4HPC Gateway Services

Data Catalog

Software Catalog

Workflow Registry

Alien4Cloud

2. Store

3. Deploy

Endpoint to invoke the Workflow

Data Logistics Pipelines

PyCOMPSs Code

TOSCA Description

Dynamic Workflow Description

1. Create Workflow

4. Share

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.
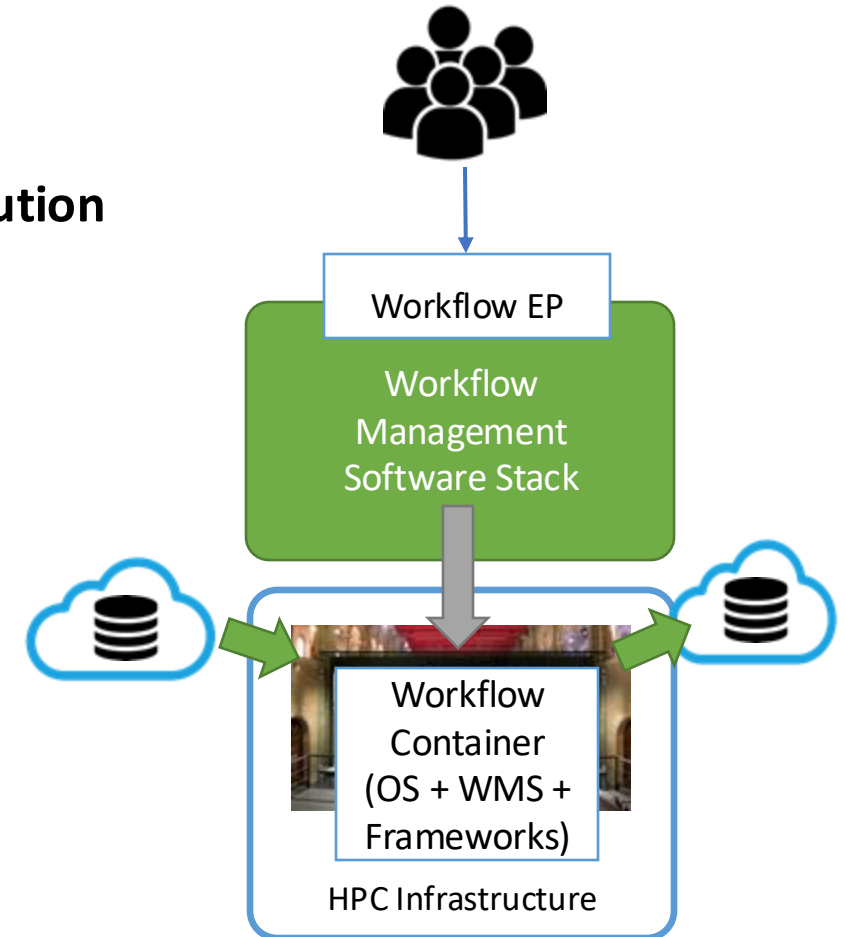
FaaS vs HPCWaaS
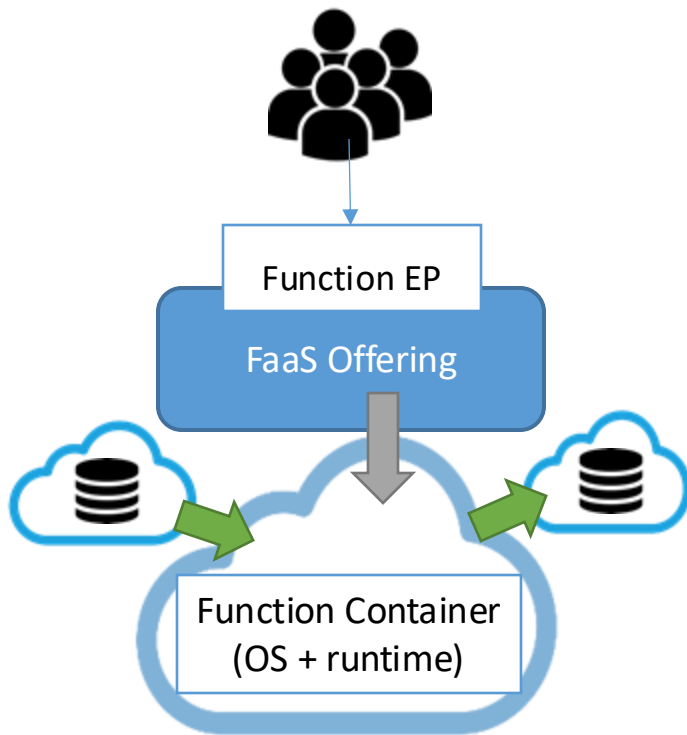
Similarities
- Easy to use for final user
- **Automated deployment & execution**
- Data integration
- **Containers**

Differences
- HPC policies & requirements
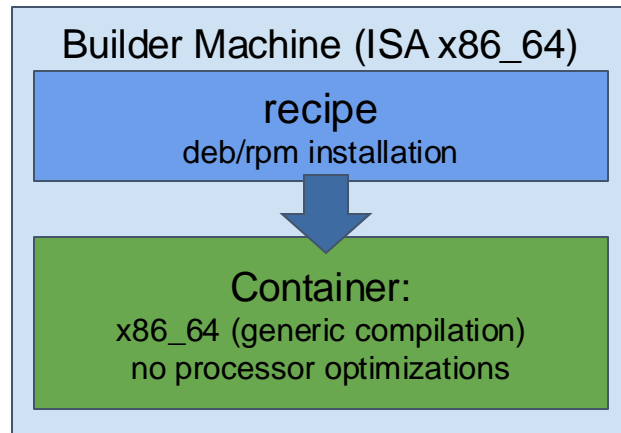- Deployment and Execution Complexity
- **Performance needs**

Function EP

FaaS Offering

Function Container
(OS + runtime)

Workflow EP

Workflow Management Software Stack

Workflow Container
(OS + WMS + Frameworks)

HPC Infrastructure

Containers and HPC

eFlows4HPC

Standard container image creation

Builder Machine (ISA x86_64)

recipe
deb/rpm installation

Container:
x86_64 (generic compilation)
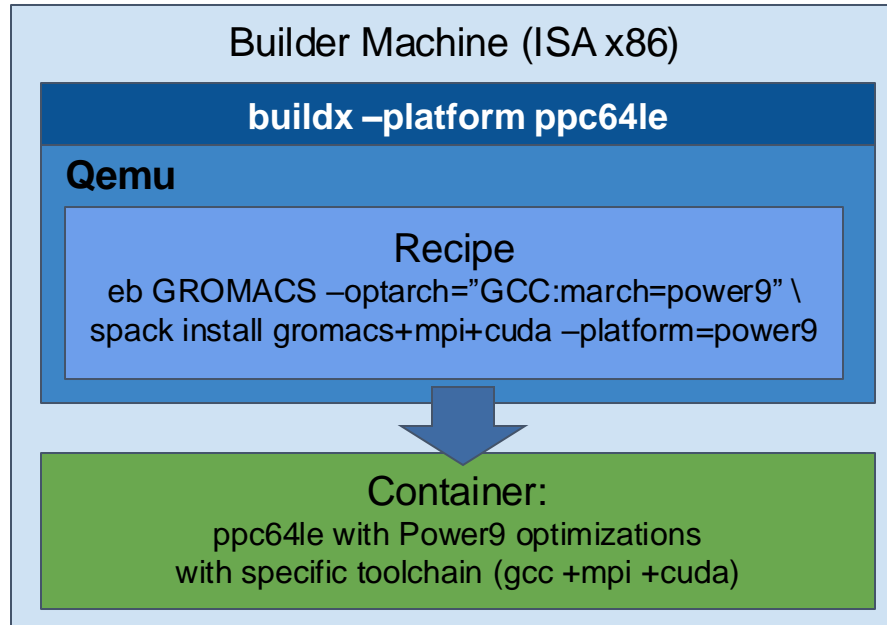no processor optimizations

- Simplicity for deployment
  - Just pull or download the image
- Trade-Off performance/portability
  - Architecture Optimizations
- Accessing Hardware from Containers
  - MPI Fabric /GPUs
- Host-Container Version Compatibility

Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

SC24
ATLANTA, NOV 17-22

HPC Ready Containers

eFlows4HPC

eFlows4HPC approach

Builder Machine (ISA x86)

buildx –platform ppc64le

Qemu

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

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

- Methodology to allow the creation containers for specific HPC system
  - Leverage HPC and Multi-platform container builders

- It is tight to do by hand but let's automate!
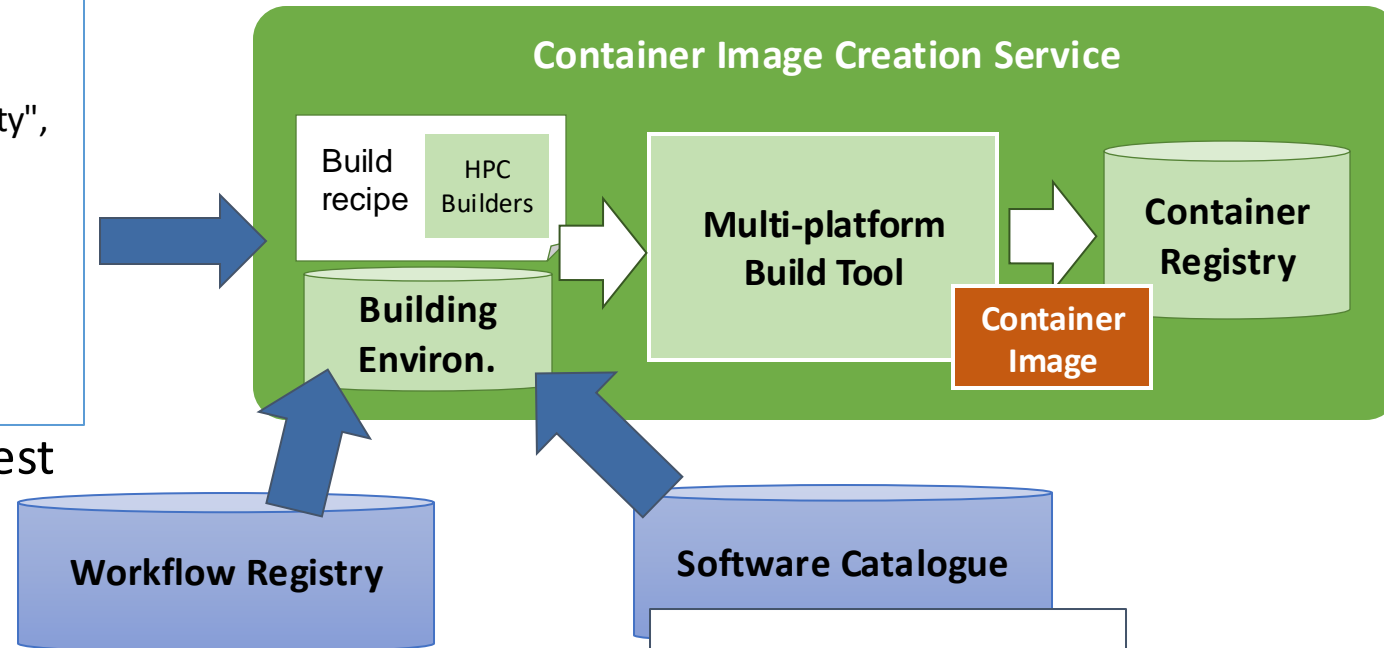
HPC Ready Containers

Workflow step + target system

```
{
"machine": {
        "platform": "linux/amd64",
        "architecture": "skylake",
        "container_engine": "singularity",
        "mpi":"openmpi@4.1.1"
},
"workflow":"tutorial",
"step_id" : " HPC_AI_training",
"workflow_yaml" : "eflows4hpc.yaml",
}
```

Service request

Container Image Creation Service

Build recipe

HPC Builders

Building Environ.

Multi-platform Build Tool

Container Image

Container Registry

Workflow Registry

Software Catalogue

Installation Description
(as HPC Builder Package)

package.py

```
1       apt:
2               - graphviz
3               - libbz2-dev
4       spack:
5               specs:
6                       - compss@3.3.2
7                       - py-dislib@master
8                       - alya@master
9       pip:
10              - pyyaml
11              - pydoe
12              - pandas
13              - pillow
```

eflows4hpc.yaml

CAELESTIS workflow

- Example of HPC ready container generation

- Sample request (json):

Yaml file describing modules involved in the workflow

Target: MareNostrum4 architecture

```json
{
"machine": {
        "platform": "linux/amd64",
        "architecture": "skylake",
        "container_engine": "singularity",
        "mpi":"openmpi@4.1.1"
},
"workflow":"tutorial",
"step_id" : "HPC_AI_training",
"workflow_yaml" : "eflows4hpc.yaml",
"force": "True",
"push" : "False"
}
```

Location in workflow-registry

workflow-registry / tutorial / HPC_AI_training / **eflows4hpc.yaml**

Jorge Ejarque add acm_summer_school workflow

Code   Blame   16 lines (16 loc) · 293 Bytes

```yaml
1    apt:
2        - graphviz
3        - libbz2-dev
4    spack:
5        specs:
6            - compss@3.3.2
7            - py-dislib@master
8            - alya@master
9    pip:
10       - pyyaml
11       - pydoe
12       - pandas
13       - pillow
14       - rocrate
```

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

SC24
ATLANTA NOV.17-22

CAELESTIS workflow

Specific spack packages

workflow-registry / tutorial / HPC_AI_training / eflows4hpc.yaml

Jorge Ejarque add acm_summer_school workflow

Code | Blame   16 lines (16 loc) · 293 Bytes

```
1    apt:
2            - graphviz
3            - libbz2-dev
4    spack:
5            specs:
6                    - compss@3.3.2
7                    - py-distlib@master
8                    - alya@master
9    pip:
10           - pyyaml
11           - pydoe
12           - pandas
13           - pillow
14           - rocrate
15           - pickle5
16           - contextvars
```

software-catalog / packages / compss / package.py

Jorge Ejarque update versions

Code | Blame   72 lines (62 loc) · 3.16 KB   Raw

```
1    # Copyright 2013-2021 Lawrence Livermore National Security, LLC and other
2    # Spack Project Developers. See the top-level COPYRIGHT file for details.
3    #
4    # SPDX-License-Identifier: (Apache-2.0 OR MIT)
5
6    # ----------------------------------------------------------------------
7    # If you submit this package back to Spack as a pull request,
8    # please first remove this boilerplate and all FIXME comments.
9    #
10   # This is a template package file for Spack.  We've put "FIXME"
11   # next to all the things you'll want to change. Once you've handled
12   # them, you can save this file and test your package like this:
13   #
14   #     spack install compss
15   #
16   # You can edit this file again by typing:
17   #
18   #     spack edit compss
19   #
20   # See the Spack documentation for more information on packaging.
21   # ----------------------------------------------------------------------
22
23   from spack import *
24
25
26 ∨ class Compss(Package):
27       """COMP Superscalar programming model and runtime."""
28
29       # Add a proper url for your package's homepage here.
30       homepage = "https://compss.bsc.es"
31       url      = "https://compss.bsc.es/repo/sc/stable/COMPSs_2.10.tar.gz"
32
```

BSC **Supercomputing Center** Centro Nacional de Supercomputación

SC24 ATLANTA, NOV 17-22

CAELESTIS workflow

Specific spack packages

workflow-registry / tutorial / HPC_AI_training / **eflows4hpc.yaml**

Jorge Ejarque add acm_summer_school workflow

Code Blame   16 lines (16 loc) · 293 Bytes

```
1    apt:
2        - graphviz
3        - libbz2-dev
4    spack:
5        specs:
6            - compss@3.3.2
7            - py-dislib@master
8            - alya@master
9    pip:
10       - pyyaml
11       - pydoe
12       - pandas
13       - pillow
14       - rocrate
15       - pickle5
16       - contextvars
```

software-catalog / packages / py-dislib / **package.py**

Jorge Ejarque update versions

Code Blame   59 lines (51 loc) · 2.5 KB   Raw

```
1    # Copyright 2013-2021 Lawrence Livermore National Security, LLC and other
2    # Spack Project Developers. See the top-level COPYRIGHT file for details.
3    #
4    # SPDX-License-Identifier: (Apache-2.0 OR MIT)
5
6    # -------------------------------------------------------------------
7    # If you submit this package back to Spack as a pull request,
8    # please first remove this boilerplate and all FIXME comments.
9    #
10   # This is a template package file for Spack.  We've put "FIXME"
11   # next to all the things you'll want to change. Once you've handled
12   # them, you can save this file and test your package like this:
13   #
14   #     spack install py-dislib
15   #
16   # You can edit this file again by typing:
17   #
18   #     spack edit py-dislib
19   #
20   # See the Spack documentation for more information on packaging.
21   # -------------------------------------------------------------------
22
23   from spack import *
24
25
26   class PyDislib(PythonPackage):
27       """FIXME: Put a proper description of your package here."""
```

CAELESTIS workflow

Specific spack packages

workflow-registry / tutorial / HPC_AI_training / eflows4hpc.yaml

Jorge Ejarque add acm_summer_school workflow

| Code | Blame | 16 lines (16 loc) · 293 Bytes |

```
1    apt:
2        - graphviz
3        - libbz2-dev
4    spack:
5        specs:
6            - compss@3.3.2
7            - py-dislib@master
8            - alya@master
9    pip:
10       - pyyaml
11       - pydoe
12       - pandas
13       - pillow
14       - rocrate
15
16
```

software-catalog / packages / alya / package.py

FernandoVN98 Added alya package

| Code | Blame | 59 lines (49 loc) · 2.03 KB | | Raw |

```
1    # Copyright 2013-2022 Lawrence Livermore National Security, LLC and other
2    # Spack Project Developers. See the top-level COPYRIGHT file for details.
3    #
4    # SPDX-License-Identifier: (Apache-2.0 OR MIT)
5
6    # -------------------------------------------------------------------
7    # If you submit this package back to Spack as a pull request,
8    # please first remove this boilerplate and all FIXME comments.
9    #
10   # This is a template package file for Spack.  We've put "FIXME"
11   # next to all the things you'll want to change. Once you've handled
12   # them, you can save this file and test your package like this:
13   #
14   #     spack install alya
15   #
16   # You can edit this file again by typing:
17   #
```
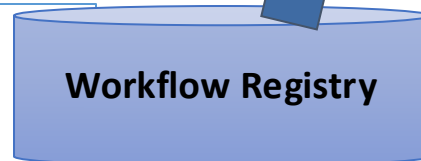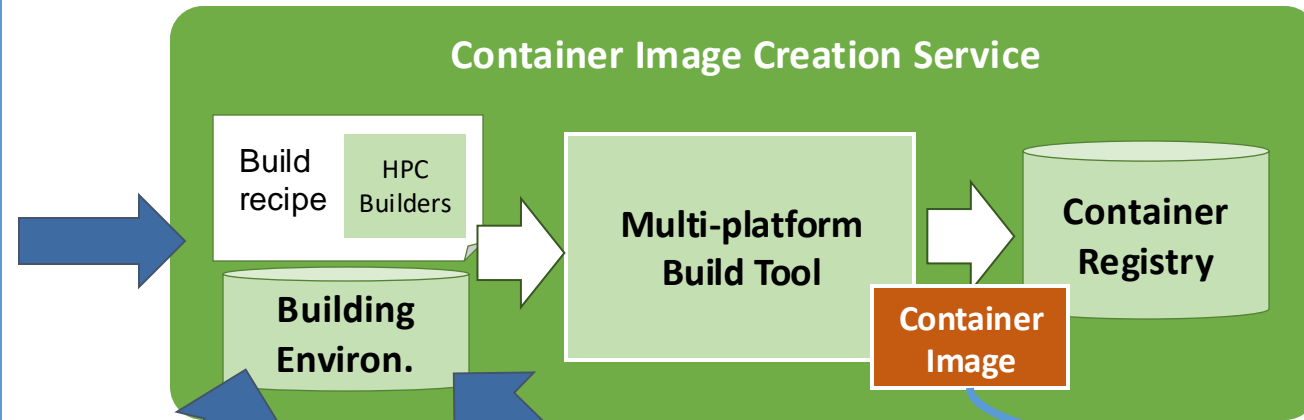
Some taken from defaul Spack repository

```
22
23   from spack import *
24   import os
25
26 v class Alya(CMakePackage):
```

![BSC Supercomputing Center - Centro Nacional de Supercomputación logo]

![SC24 Atlanta Nov 17-22 logo]

HPC Ready Containers

eFlows4HPC

Service request

```
{
"machine": {
        "platform": "linux/amd64",
        "architecture": "skylake",
        "container_engine": "singularity",
        "mpi":"openmpi@4.1.1"
},
"workflow":"tutorial",
"step_id" : "HPC_AI_training",
"workflow_yaml" : "eflows4hpc.yaml",
}
```

Container Image Creation Service

Build recipe

HPC Builders

Building Environ.

Multi-platform Build Tool

Container Image

Container Registry

Workflow Registry

Software Catalogue

Installation Description
(as HPC Builder Package)

sc24_workflow_tutorial.sif

```
1    apt:
2          - graphviz
3          - libbz2-dev
4    spack:
5          specs:
6                 - compss@3.3.2
7                 - py-dislib@master
8                 - alya@master
9    pip:
10         - pyyaml
11         - pydoe
12         - pandas
13         - pillow
```

eflows4hpc.yaml

package.py

SC24

ATLANTA, NOV 17-22

15

Further Information

- Project page: http://www.bsc.es/compss
  - Documentation
  - Virtual Appliance for testing & sample applications
  - Tutorials
- Source Code
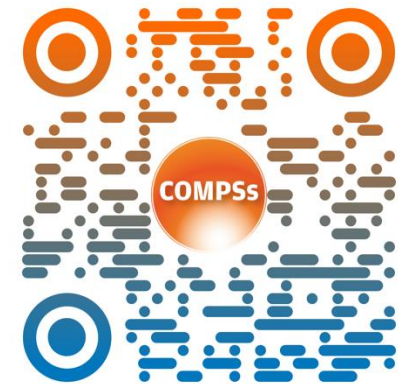
    https://github.com/bsc-wdc/compss

- Docker Image

    https://hub.docker.com/r/compss/compss

- Applications

    https://github.com/bsc-wdc/apps

    https://github.com/bsc-wdc/dislib

- Dislib
  - https://dislib.readthedocs.io/en/latest/

ACKs

Thanks!

rosa.m.badia@bsc.es