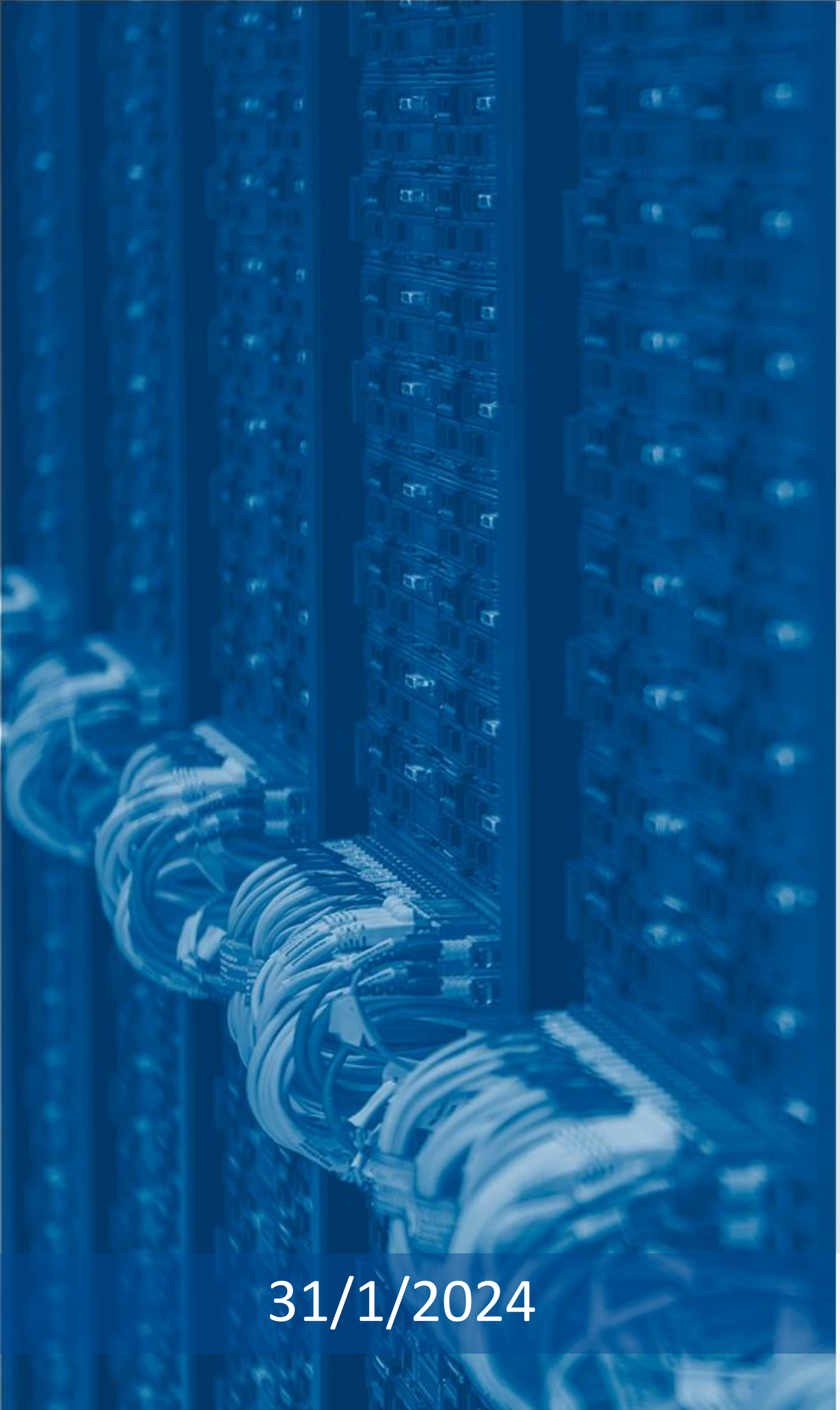


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Provenance with PyCOMPSs (hands-on included)

Raül Sirvent

31/1/2024

BSC Training Course: Programming Distributed Computing Platforms with COMPSs

Outline

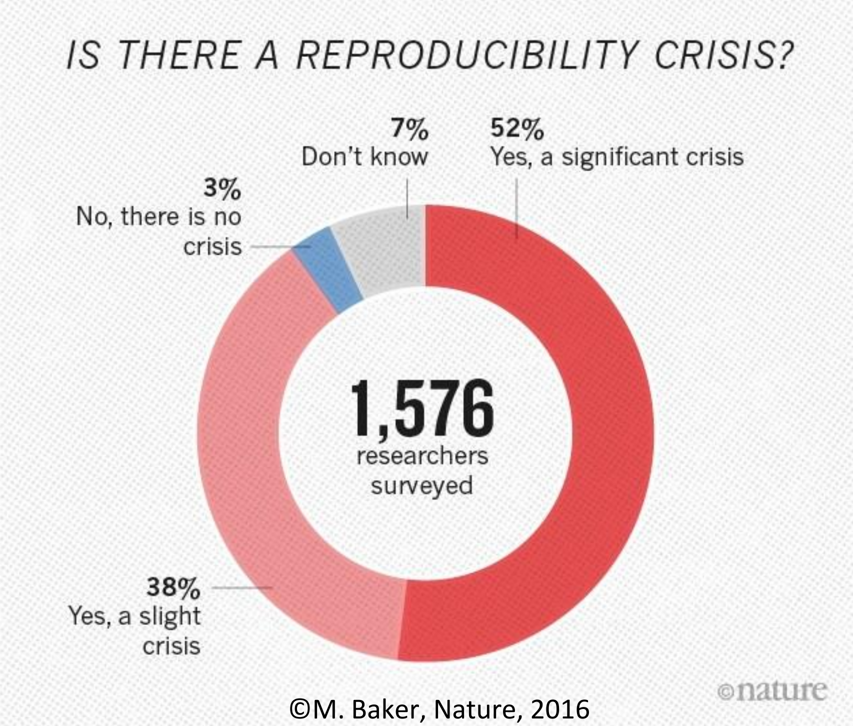
- Motivation and Background
- Design of Workflow Provenance recording
- Using Workflow Provenance with COMPSs
- Inspecting registered metadata
- Hands-on exercises

Motivation and Background

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Motivation

- Large number of **Scientific Workflows** experiments
  - Keep track of results - **Governance**
- **Reproducibility** crisis in scientific papers
  - Conferences now request artifacts
    - E.g. SC Reproducibility Initiative
- **Provenance recording** can help with both problems



©M. Baker, Nature, 2016

- **Provenance:** The chronology of the origin, development, ownership, location, and changes to a system or system component and associated data
  - Need to record metadata
  - Our focus: Workflow Provenance (data + software)

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

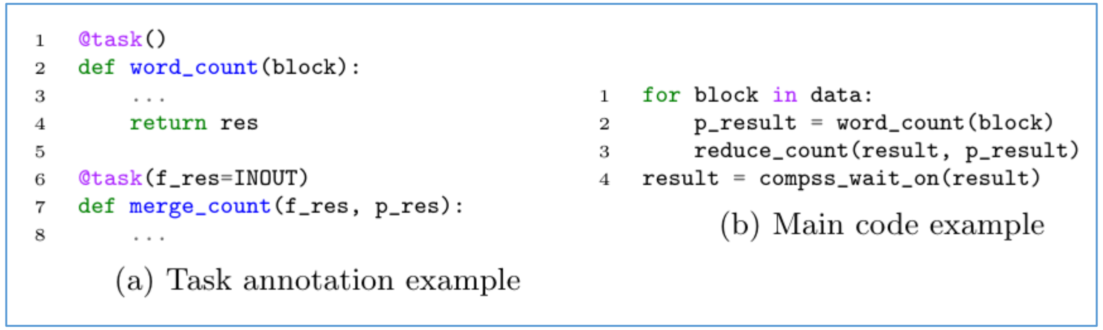
Motivation

- Provenance is **MORE** than just Reproducibility
  - **Governance** (availability, usability, consistency, …) (**FAIR Workflows**)
  - **Replicability** (exchange inputs)
  - **Knowledge extraction** (queries, mining)
  - **Traceability** (validation/verification, visualisation)

- Our claim: desired features for **Workflow Provenance** registration
  - **Automatic:** lower user burden
  - **Efficient:** no overheads
  - **Scalable:** large workflows (both tasks and data assets used)

Background: COMPSs

- **Sequential** programming, **parallel** execution
- **General purpose** programming language + **annotations/hints** (identify tasks and directionality of data)
- Builds a **task graph** at runtime (potential concurrency)
- Tasks can be **sequential**, **parallel** (threaded or MPI)
- Offers to applications a **shared memory illusion** in a distributed system (Big Data apps support)
- Support for **persistent storage**
- **Agnostic** of computing platform: enabled by the runtime for **clusters**, **clouds** and **container** managed clusters



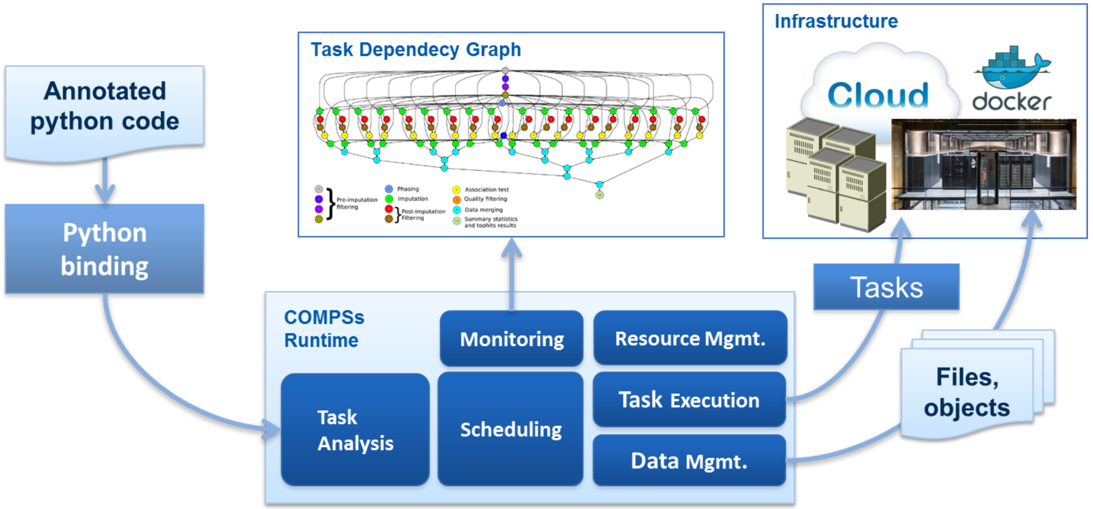
```
@task()
def word_count(block):
    ...
    return res

@task(f_res=INOUT)
def merge_count(f_res, p_res):
    ...
```

(a) Task annotation example

```
for block in data:
    p_result = word_count(block)
    reduce_count(result, p_result)
result = compss_wait_on(result)
```

(b) Main code example



- **Advanced features:** heterogeneous infrastructures, task constraints, streamed data, task faults, task exceptions, checkpointing, elasticity

Background: Research Object Crate

- Package research data + metadata
- Evolution from:
  - **Research Object**: describe digital and real-world resources
  - **DataCrate**: aggregate data with metadata
- Lightweight format
  - Both **machines** and **humans** can read it
- JSON Linked Data (JSON-LD)
  - Vocabulary: Schema.org
  - Structure:
    - **Root Data Entity**
    - **Data Entities** (files, directories)
    - **Contextual Entities** (non-digital elements)
- Strong ecosystem, we use:
  - ro-crate-py library
  - WorkflowHub

RO-Crate 1.1

WorkflowHub

Background: RO-Crate Profiles

- RO-Crate is very **generic** (wide scope)
  - Profiles enable **Interoperability**
    - Set of conventions, types and properties (MUST, SHOULD, …)
- **Workflow RO-Crate** profile
  - MUST **ComputationalWorkflow**, **mainEntity** (Root Dataset)
  - SHOULD **WorkflowSketch**
- **Workflow Run RO-Crate** profile collection (MUST **CreateAction**)
  - Process Run Crate (set of tools)
  - Workflow Run Crate (computational workflow)
  - Provenance Run Crate (detailed computational workflow)

Workflow RO-Crate

Workflow Run RO-Crate

Simone Leo et al. “Recording provenance of workflow runs with RO-Crate” arXiv preprint arXiv:2312.07852 (Dec 2023)

Design of Workflow Provenance recording

Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

Design Requirements

- Target HPC workflows (commonly large)
- Provenance representation format
  - Simple but able to represent complex workflows
- **Automatic** provenance registration (no explicit annotations)
- **Efficient** provenance registration (avoid overheads at run time)
- **Scale** to large workflows (thousands of files and tasks)

COMPSs + Workflow RO-Crate

Workflow Run RO-Crate + WorkflowHub

COMPSs runtime modifications

COMPSs

- Flags –p or --provenance trigger it after execution
- Can be manually invoked if provenance generation time becomes an issue (i.e., extreme large workflows)

dataprovenance.log

- Lightweight approach: record file accesses, generate provenance later

```
3.3
lysozyme_in_water.py
App_Profile.json
file://s01r2b54-ib0/home/bsc19/bsc19057/DP_Test_3_demo/dataset/2hs9.pdb IN
file://s01r2b54-ib0/home/bsc19/bsc19057/DP_Test_3_demo/output/2hs9.gro OUT
file://s01r2b54-ib0/home/bsc19/bsc19057/DP_Test_3_demo/output/2hs9.top OUT
...
```

After application finishes…

generate_COMPSs_RO-Crate.py

ro-crate-info.yaml

ro-crate-py 0.9.0

COMPSs_RO-Crate_[uuid]/

- It's the *crate*
- ro-crate-metadata.json
- Application source files, command line arguments, workflow image and profile

WorkflowHub

F Findable

A Accessible

I Interoperable

R Reusable

generate_COMPSs_RO-Crate.py features

- Detects and records **COMPSs version** used and the **mainEntity**
  - Looks for alternatives, if not found
- Automatically detects overall **inputs** and **outputs** of the workflow
  - Discards intermediate generated results as inputs
- Respects application **source files** sub-directory structure
- If data persistence, machine paths translated to crate paths
  - Identifies **common paths** to correctly arrange files
    - E.g. inputs/00/input_file.txt
- If no persistence: **URIs** to files are generated, **size** and **modification** date of files are stored to record the file version

Using Workflow Provenance with COMPSs

Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

Steps to record and publish Workflow Provenance in COMPSs

- Install ro-crate-py (if needed)
- Provide YAML information file
- Run with -p or --provenance
  - The ***crate*** is generated (a sub-folder COMPSs_RO-Crate_[uuid]/)
- Publish it at WorkflowHub, using the crate
- Generate a DOI, cite your results in papers

Install ro-crate-py

- pip install rocrate
- pip install rocrate --user
    - Typically, installs the library in ~/.local/
- pip install -t install_path rocrate
    - Specify target directory

https://github.com/ResearchObject/ro-crate-py

YAML information to be provided

- Non-automatically gathered info: **ro-crate-info.yaml**
- Sections:
  - COMPSs Workflow Information
  - Authors
  - Submitter
- Data persistence: True or False
- No inputs/outputs are provided, automatically detected by the provenance generation script

```yaml
COMPSs Workflow Information:
  name: COMPSs Matrix Multiplication
  description: Blocks as hypermatrix
  license: Apache-2.0
  sources: [src/, ~/java/matmul/xml/,
   ~/java/matmul/pom.xml, Readme]
  data_persistence: True
Authors:
  - name: Rosa M. Badia
    e-mail: Rosa.M.Badia@bsc.es
    orcid: https://orcid.org/0000-0003-2941-5499
    organisation_name: Barcelona Supercomputing Center
    ror: https://ror.org/05sd8tv96
Submitter:
  name: Raül Sirvent
  e-mail: Raul.Sirvent@bsc.es
  orcid: https://orcid.org/0000-0003-0606-2512
  organisation_name: Barcelona Supercomputing Center
  ror: https://ror.org/05sd8tv96
```

Run your COMPSs application

- runcompss -p
- enqueue_compss -p
- pycompss run -p

- Either **-p** or **--provenance**

- Post-process automatically triggered after the end of the application
- Log and time statistics are provided
  - grep PROVENANCE
- If provenance generation fails for any reason:
  - Still possible to invoke it manually (commands provided in the output log)

```
...
PROVENANCE | RO-Crate writing to disk TIME: 0.01987314224243164 s
PROVENANCE | Workflow Provenance generation TOTAL EXECUTION TIME: 0.04113888740539551 s
PROVENANCE | COMPSs Workflow Provenance successfully generated in sub-folder:
    COMPSs_RO-Crate_d64966ac-fe34-463a-88fc-f97047c21a99/
PROVENANCE | ENDED WORKFLOW PROVENANCE SCRIPT
```

The Crate (resulting folder)

- application_sources/
- dataset/
- complete_graph.svg
- App_Profile.json
- compss_submission_command_line.txt
- ro-crate-metadata.json

```
|-- App_Profile.json
|-- application_sources
|   |-- Readme
|   |-- pom.xml
|   |-- src
|   |   `-- main
|   |       `-- java
|   |           `-- matmul
|   |               |-- arrays
|   |               |   |-- …
|   |               |   `-- Matmul.java
|   |               |-- files
|   |               |   |-- Block.class
|   |               |   |-- Block.java
|   |               |   |-- Matmul.class
|   |               |   |-- Matmul.java
|   |               |   |-- MatmulImpl.class
|   |               |   |-- MatmulImpl.java
|   |               |   |-- MatmulItf.class
|   |               |   `-- MatmulItf.java
|   |               `-- objects
|   |                   |-- …
|   |                   `-- Matmul.java
|   `-- xml
|       |-- project.xml
|       `-- resources.xml
|-- complete_graph.svg
|-- compss_submission_command_line.txt
|-- dataset
|   |-- …
|   `-- C.1.1
|-- ro-crate-info.yaml
`-- ro-crate-metadata.json

10 directories, 41 files
```

Publish your results with WorkflowHub

- **zip** –r crate.zip COMPSs_RO-Crate_[uuid]/
- Login to WorfklowHub
- Create -> Workflow
  - **Upload**/Import Workflow RO-Crate tab -> Local file (crate.zip)
  - Click Register
- Review automatically obtained information
- Select the visibility of your workflow in the Sharing tab (for both general public, and for teams selected)
- Click **Register** again

Cite your results with WorkflowHub

- Freeze your workflow version
  - Overview tab -> Citation box -> Freeze version
  - Actions menu -> Freeze version
- Generate DOI
  - **IMPORTANT:** make sure your version is final
  - Citation box -> Generate a DOI
  - Actions menu -> Generate a DOI
  - Select Mint DOI
- The **final generated DOI** for the workflow results can be found in the Citation box

https://doi.org/10.48546/workflowhub.workflow.484.1

SC Conference Reproducibility Initiative

- Artifacts Available ✓
  - Artifacts used in the research (including data and code) are permanently archived in a **public repository** that assigns a **global identifier** and **guarantees persistence**, and are made available via **standard open licenses** that maximize artifact availability

- Artifacts Evaluated-Functional ✓
  - Documentation: Are the artifacts sufficiently **documented** to enable them **to be exercised** by readers of the paper?
  - Completeness: Do the submitted artifacts **include all of the key components** described in the paper?
  - Exercisability: Do the submitted artifacts **include the scripts and data needed to run the experiments** described in the paper, and can the software be successfully executed?

- Results Replicated
  - Reproduce Behavior: determine the equivalent or approximate behavior on available hardware
  - Reproduce the Central Results and Claims of the Paper

Inspecting registered metadata

Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

Inspecting registered metadata

```
"@id": "application_sources/matmul_files.py",
"@type": ["File", "SoftwareSourceCode", "ComputationalWorkflow"],
"contentSize": 1948,
"description": "Main file of the COMPSs workflow source files",
"encodingFormat": "text/plain",
"image": {"@id": "complete_graph.svg"},
"name": "matmul_files.py",
"programmingLanguage": {"@id": "#compss"}
```

```
"@id": "#compss",
"@type": "ComputerLanguage",
"alternateName": "COMPSs",
"citation": "https://doi.org/10.1007/s10723-
   013-9272-5",
"name": "COMPSs Programming Model",
"url": "http://compss.bsc.es/",
"version": "3.3"
```

Workflow RO-Crate

```
"@id": "complete_graph.svg",
"@type": ["File", "ImageObject", "WorkflowSketch"],
"about": {"@id": "application_sources/matmul_files.py"},
"contentSize": 6681,
"description": "The graph diagram of the workflow, automatically generated by COMPSs runtime",
"encodingFormat": [[”image/svg+xml",{"@id": "https://www.nationalarchives.gov.uk/PRONOM/fmt/92"}]],
"name": "complete_graph.svg"
```

Inspecting registered metadata

Auxiliary Files

```
"@id":
   "application_sources/matmul_tasks.py",
"@type": ["File", "SoftwareSourceCode"]
"contentSize": 1549,
"description": "Auxiliary File",
"encodingFormat": "text/plain",
"name": "matmul_tasks.py"
```

Command line arguments

```
"@id": "compss_submission_command_line.txt",
"@type": "File",
"contentSize": 709,
"description": "COMPSs command line execution
   command (runcompss), including flags and
   parameters passed",
"encodingFormat": "text/plain",
"name": "compss_submission_command_line.txt"
```

COMPSs Task Profiling

```
"@id": "App_Profile.json",
"@type": "File",
"contentSize": 247,
"description": "COMPSs application Tasks profile",
"encodingFormat": ["application/json",{"@id":"https://www.nationalarchives.gov.uk/PRONOM/fmt/817"}],
"name": "App_Profile.json"
```

Inspecting registered metadata

Persistent Data

```
"@id": "dataset/A.0.0",
"@type": "File",
"contentSize": 16,
"dateModified": "2023-09-07T09:20:20",
"name": "A.0.0",
"sdDatePublished": "2023-09-07T09:20:27+00:00"
```

Non-Persistent Data

```
"@id": "file://s07r1b33-ib0/home/bsc19/bsc19057/DP_Test_3_demo/dataset/133l.pdb",
"@type": "File",
"contentSize": 116154,
"dateModified": "2022-04-20T13:20:58",
"name": ”133l.pdb",
"sdDatePublished": "2022-10-18T08:03:08+00:00"
```

```
"@id": "file://s02r2b26-ib0/home/bsc19/bsc19057/DP_Test_3_demo/config/energy.selection"
```

Hostname

Location path in hostname

Inspecting registered metadata

Workflow Run RO-Crate

CreateAction

```
"@id": "#COMPSs_Workflow_Run_Crate_marenostrum4_SLURM_JOB_ID_30132875",
"@type": "CreateAction",
"actionStatus": {"@id": "http://schema.org/CompletedActionStatus"},
"agent": {"@id": "https://orcid.org/0000-0003-0606-2512"},
"description": "Linux s01r2b48 4.4.59-92.20-default #1 SMP Wed May 31 14:05:24 UTC 2017 (8cd473d)
   x86_64 x86_64 x86_64 GNU/Linux SLURM_JOB_NAME=matmul-DP COMPSS_PYTHON_VERSION=3.9.10
   SLURM_JOB_QOS=debug SLURM_MEM_PER_CPU=1880 COMPSS_BINDINGS_DEBUG=1 SLURM_JOB_ID=30132875
   SLURM_JOB_USER=bsc19057 COMPSS_HOME=/apps/COMPSs/3.2/ SLURM_JOB_UID=2952
   SLURM_SUBMIT_DIR=/gpfs/home/bsc19/bsc19057/COMPSs-DP SLURM_JOB_NODELIST=s01r2b48
   SLURM_JOB_GID=2950 SLURM_JOB_CPUS_PER_NODE=48 COMPSS_MPIRUN_TYPE=impi SLURM_SUBMIT_HOST=login3
   SLURM_JOB_PARTITION=main SLURM_JOB_ACCOUNT=bsc19 SLURM_JOB_NUM_NODES=1
   COMPSS_MASTER_NODE=s01r2b48 COMPSS_WORKER_NODES=",
"endTime": "2023-09-07T09:46:26+00:00",
"instrument": {"@id": "application_sources/matmul_files.py"},
"name": "COMPSs matmul_files.py execution at marenostrum4 with JOB_ID 30132875",
```

Inspecting registered metadata

Workflow Run
RO-Crate

CreateAction

```
"object": [{"@id": "dataset/A.0.0"}, {"@id": "dataset/A.0.1"}, {"@id": "dataset/A.1.0"},
   {"@id": "dataset/A.1.1"}, {"@id": "dataset/B.0.0"}, {"@id": "dataset/B.0.1"}, {"@id":
   "dataset/B.1.0"}, {"@id": "dataset/B.1.1"}, {"@id": "dataset/C.0.0"}, {"@id":
   "dataset/C.0.1"}, {"@id": "dataset/C.1.0"}, {"@id": "dataset/C.1.1"}],

"result": [{"@id": "dataset/C.0.0"}, {"@id": "dataset/C.0.1"}, {"@id": "dataset/C.1.0"},
   {"@id": "dataset/C.1.1"}, {"@id": "./"}],

"subjectOf": ["https://userportal.bsc.es/"]
```

Conclusions

- **FAIR HPC workflows** combining COMPSs + RO-Crate + WorkflowHub
  - WMS that use RO-Crate (Galaxy, Nextflow, Streamflow, Sapporo, Autosubmit)
- Paper* experiments show
  - We provide **automatic** provenance registration (whenever possible)
  - We are **efficient** (no run time overhead appreciated)
  - We can **scale** and deal with large workflows (shown by use cases)
- Future Work
  - Integration with: WfExS, ROHub (RO-Crate)
  - Automatic reproducibility with the PyCOMPSs CLI
  - Governance and Knowledge extraction

*Raül Sirvent et al. "Automatic, Efficient and Scalable Provenance Registration for FAIR HPC Workflows" In: 2022 IEEE/ACM Workshop on Workflows in Support of Large-Scale Science (WORKS). IEEE, 2022. p. 1-9.

Hands-on Exercises

Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

Preliminary steps

- Find help in the Manual:
  - COMPSs ReadTheDocs -> Tools -> Workflow Provenance
- Create your WorkflowHub account
  - Open https://workflowhub.eu/
  - Click “Register”
    - “Log in using GitHub” or
    - Register with your e-mail
      - Mandatory: First name, Last name, e-mail. Recommended: ORCID
      - Confirm registration with received e-mail
- Join “COMPSs Tutorials” team (“eFlows4HPC” Space)
  - “Join a Team”
    - Search for ”COMPSs Tutorials”
    - Organization:
      - Search for your institution not only by acronym, also with full words
      - Can try also: Browse -> Organizations -> Country (filter)
      - If not found: Create -> Organization

- **List of commands used in the Exercises: /apps/COMPSs/TUTORIALS/2024_BSC_TRAINING_COMPSS_MN4/PROVENANCE_COMMANDS.txt**

Exercise 1: Publish a Workflow Run

- Choose a COMPSs example from the ones you have previously run in the Cluster Hands-on session
  - **Easy:** Lysozyme in Water (any version) (**mandatory dataset_small/**)
  - **Medium:** Cholesky (can play with SIZE and BSIZE), K-means, Clustering Comparison
  - **Hard:** Wordcount (both reduce and merge)
- Create/edit ro-crate-info.yaml
  - cp /apps/COMPSs/TUTORIALS/2024_BSC_TRAINING_COMPSS_MN4/ro-crate-info.yaml .
  - Establish yourself as ‘Submitter’
    - If you don’t have an ORCID, just remove the ‘Submitter’ section, the Author will be considered the Submitter

Exercise 1: Publish a Workflow Run

- ro-crate-info.yaml

```
COMPSs Workflow Information:
  name: Lysozyme in water sample
  description: Lysozyme in water sample COMPSs application
  license: Apache-2.0
  sources: [src/, launch.sh]
  data_persistence: True
Authors:
  - name: Rosa M. Badia
    e-mail: Rosa.M.Badia@bsc.es
    orcid: https://orcid.org/0000-0003-2941-5499
    organisation_name: Barcelona Supercomputing Center
    ror: https://ror.org/05sd8tv96
Submitter:
  name: Raül Sirvent
  e-mail: Raul.Sirvent@bsc.es
  orcid: https://orcid.org/0000-0003-0606-2512
  organisation_name: Barcelona Supercomputing Center
  ror: https://ror.org/05sd8tv96
```

Minimal information

Odd usernames: True

Even usernames: False

Hint for Wordcount: look for the optional parameters of ro-crate-info.yaml in the manual

Exercise 1: Publish a Workflow Run

- Edit launch.sh script: add “--provenance” option
- Submit: Example ./launch.sh 2 10 false $(pwd)/config/ $(pwd)/dataset_small/ $(pwd)/output/
  - **Change number of nodes, or other parameters**
    - **Lysozyme in water: launch.sh, launch_full.sh, launch_full_no_mpi.sh or launch_full_singularity.sh**
    - **Cholesky: SIZE, BSIZE**
    - **Wordcount: launch.sh or launch_merge.sh**
- Check time and result of provenance generation (compss_XXXX.out)
  - What was the longest time during generation?
- Zip and upload RO-Crate
  - zip –r my_app_crate.zip COMPSs_RO-Crate_XXX/
- Copy the file to your laptop. Run, from your laptop:
  - scp nct01XXX@mn3.bsc.es:~/my_app_folder/my_app_crate.zip ~/Desktop/
- Go to WorkflowHub -> Contribute
    - Upload/Import Workflow RO-Crate -> Select local file
    - Briefly inspect imported metadata
    - Select Team, check Sharing permissions, click Register

Exercise 1: Publish a Workflow Run

- DOI generation and reference (**DON’T DO THIS NOW!!!!!!**)
  - Freeze version
  - Generate a DOI
  - Share the obtained DOI (e.g. use it as a reference in a paper)

- **Example: ROM Workflow DOI generated live**

Exercise 2: Inspect a previous published execution

- Find your own published workflow
  - My Items -> Workflows
- Can you understand the metadata (**ro-crate-metadata.json**)? Ask, if not
  - Identify the 3 main parts of the JSON: Root Data Entity, Data Entities, Contextual Entities
  - ro-crate-metadata.json interesting keywords: mainEntity, ComputationalWorkflow, WorkflowSketch, #compss, CreateAction (object, result)
  - Observe the CreateAction in detail
- Questionnaire:
  - Who ran this code? Where? When? With which COMPSs version?
  - What is the name of the main application source file?
  - What were the inputs and outputs used or generated in this workflow run?
  - Can you say how many cluster nodes were used for the run? (Hint: 3 locations)
  - Where can you find detailed profiling of the application? (Hint: 2 locations)
  - Are the data assets included in the package?
  - What was the command used to run this workflow? What were the parameters passed to the application?
  - Navigate the workflow diagram

Exercise 2 (extra): Inspect a previous published execution

- Browse for other COMPSs Workflows at WorkflowHub
  - Browse -> Workflows
    - Workflow type: filter by “COMPSs”
    - Team: filter by “COMPSs Tutorials” (**or don’t**)
- Inspect the metadata (keywords: mainEntity, ComputationalWorkflow, WorkflowSketch, #compss, CreateAction (object, result))
- Questionnaire:
  - Who ran this code? Where? When? With which COMPSs version?
  - What is the name of the main application source file?
  - What were the inputs and outputs used or generated in this workflow run?
  - Can you say how many cluster nodes were used for the run? (Hint: 3 locations)
  - Where can you find detailed profiling of the application? (Hint: 2 locations)
  - Are the data assets included in the package?
  - What was the command used to run this workflow? What were the parameters passed to the application?
  - Navigate the workflow diagram

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

Exercise 3: Repeat Exercises 1 and 2 with data_persistence flipped

- If your data_persistence was True, set it to False
- If it was False, set it to True

- Re-execute your application
- Publish your new run to WorkflowHub
- Compare the runs
  - dataset/ folder???
  - ro-crate-metadata.json -> check data assets
    - Search for a specific file you know
    - Or look for "CreateAction"

Exercise 4: Reproduce an execution from another participant in the tutorial

- Browse for COMPSs Workflows at WorkflowHub
  - Browse -> Workflows
    - Workflow type: filter by “COMPSs”
    - Team: filter by “COMPSs Tutorials”
  - **Select preferably one with data_persistence=True (i.e. has a dataset/ folder)**
- Click the workflow, click “Download RO-Crate”
- Create a directory at the supercomputer
  - mkdir ~/reproduced_app/
- Copy the file from your laptop to the supercomputer
  - scp ~/Desktop/workflow-XXX-X.crate.zip nct01XXX@mn3.bsc.es:~/reproduced_app/your_app_crate.zip
- At the supercomputer, in the ~/reproduced_app/ folder
  - unzip workflow-XXX-X.crate.zip

Exercise 4: Reproduce an execution from another participant in the tutorial

- Understand inputs/outputs location (**ro-crate-metadata.json**)
  - Check CreateAction -> object and result
- Example with dataset/
  - Inputs:
    - dataset/lysozyme_in_water/config/
    - dataset/lysozyme_in_water/dataset_small/
  - Outputs:
    - dataset/lysozyme_in_water/output/
- Example without dataset/
  - Inputs:
    - file://s02r1b59-ib0/home/nct01/nct00XXX/lysozyme_in_water/config/
    - file://s02r1b59-ib0/home/nct01/nct00XXX/lysozyme_in_water/dataset_small/
  - Outputs
    - file://s02r1b59-ib0/home/nct01/nct00XXX/lysozyme_in_water/output/
- Check parameters used in the run (**compss_submission_command_line.txt**)
  - Specifically, the num_nodes parameter

Exercise 4: Reproduce an execution from another participant in the tutorial

- **CREATE A NEW OUTPUT FOLDER (avoid overwriting the recorded one)**
  - cd application_sources/
  - mkdir new_output/
- Resubmit the application with the correct paths:
  - chmod ugo+x launch.sh
  - Provenance recording can be deactivated for this run (remove --provenance)
  - Persistence example:
    - ./launch.sh 2 10 false ../dataset/lysozyme_in_water/config/ ../dataset/lysozyme_in_water/dataset_small/ **new_output/**
  - Non-persistence example:
    - ./launch.sh 2 10 false /home/nct01/nct00XXX/lysozyme_in_water/config/ /home/nct01/nct00XXX/lysozyme_in_water/dataset_small **new_output/**
- Compare results:
  - diff new_output/ ../dataset/lysozyme_in_water/output/
  - diff new_output/ /home/nct01/nct00XXX/lysozyme_in_water/output
  - **Are they identical? Why?**

eFlows4HPC
Enabling dynamic and Intelligent workflows
in the future EuroHPC ecosystem

DT-GEO

Barcelona
Supercomputing
Center
Centro Nacional de Supercomputación

Thank you for your attention

https://compss-doc.readthedocs.io/en/latest/Sections/05_Tools/04_Workflow_Provenance.html

Raul.Sirvent@bsc.es

MINISTERIO
DE CIENCIA
E INNOVACIÓN

Financiado por
la Unión Europea
NextGenerationEU

Plan de Recuperación,
Transformación y
Resiliencia

AGENCIA
ESTATAL DE
INVESTIGACIÓN