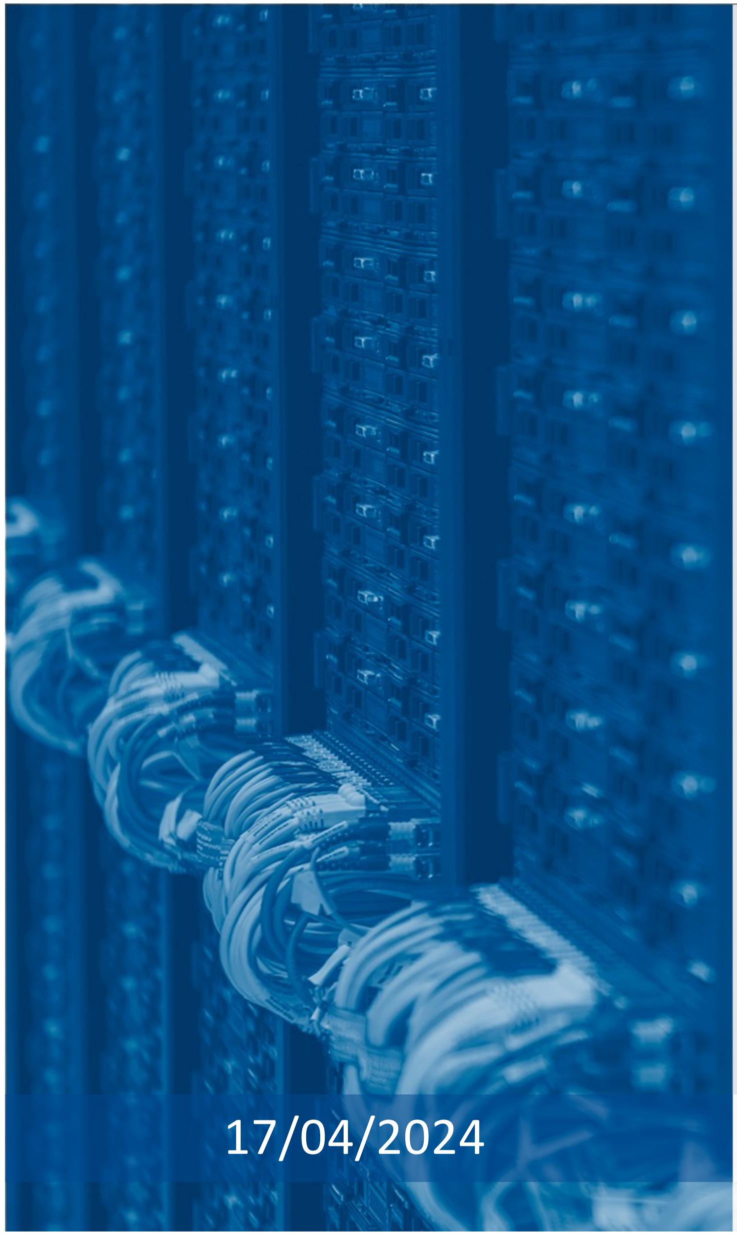


Barcelona Supercomputing Center
Centro Nacional de Supercomputación

# Workflow Provenance recording for Automatic Reproducibility

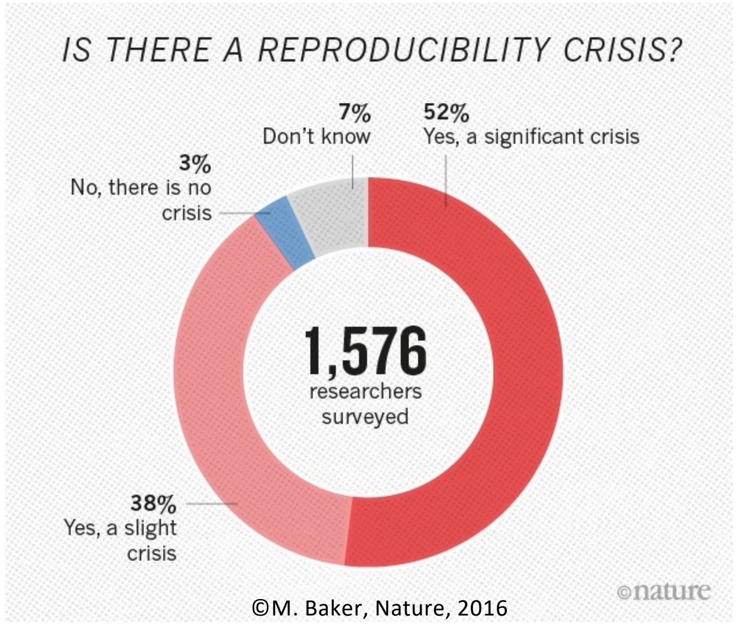
Raül Sirvent

17/04/2024

16th JLESC Workshop, Kobe, Japan

## 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

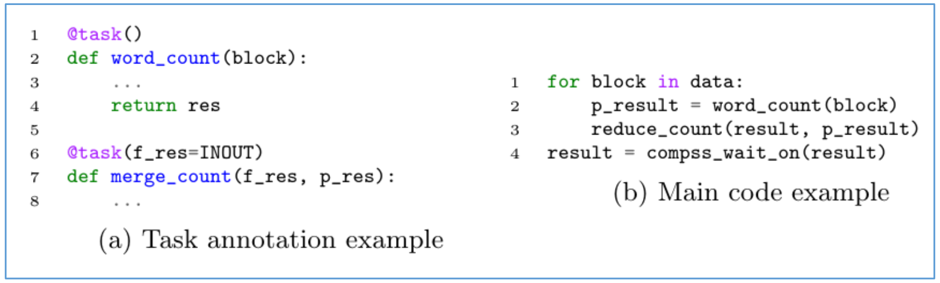


©M. Baker, Nature, 2016

- **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)

# Baseline: 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



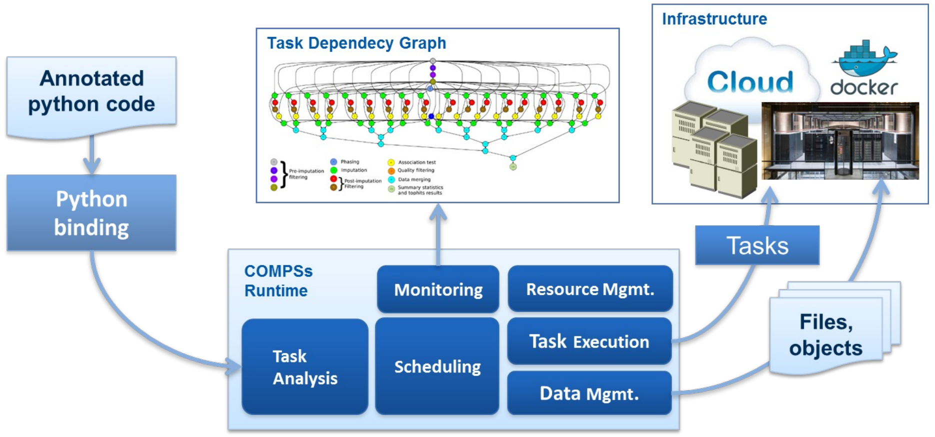
```
@task()
def word_count(block):
    ...
    return res

@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

## Baseline: 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: easier **generation**
  - WorkflowHub: demonstrates **interoperability**

RO-Crate 1.1

WorkflowHub

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

## Baseline: 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)

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

## WMS/tools using WRROC for Provenance Recording

COMPSs

Galaxy

StreamFlow

SAPPORO WES

AUTOSUBMIT

- runcrate

nextflow

WfExS

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

- 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

Findable Accessible Interoperable Reusable

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, **uploading the crate** or **using GitHub**
- Generate a DOI, cite your results in papers

Barcelona Supercomputing Center
Centro Nacional de Supercomputación

## Future Work

- Automatic reproducibility through RO-Crate:
  - PyCOMPSs CLI, WfExS, Chameleon(?),...
- JLESC
  - People working on Reproducibility
    - ANL – Kate Keahey
    - INRIA – Alexandru Costan

References:

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

eFlows4HPC
Enabling dynamic and Intelligent workflows in the future EuroHPC ecosystem

DT-GEO

MINISTERIO DE CIENCIA E INNOVACIÓN
Financiado por la Unión Europea NextGenerationEU
Plan de Recuperación, Transformación y Resiliencia
AGENCIA ESTATAL DE INVESTIGACIÓN

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