WfCommons

A framework for enabling scientific workflow research and development

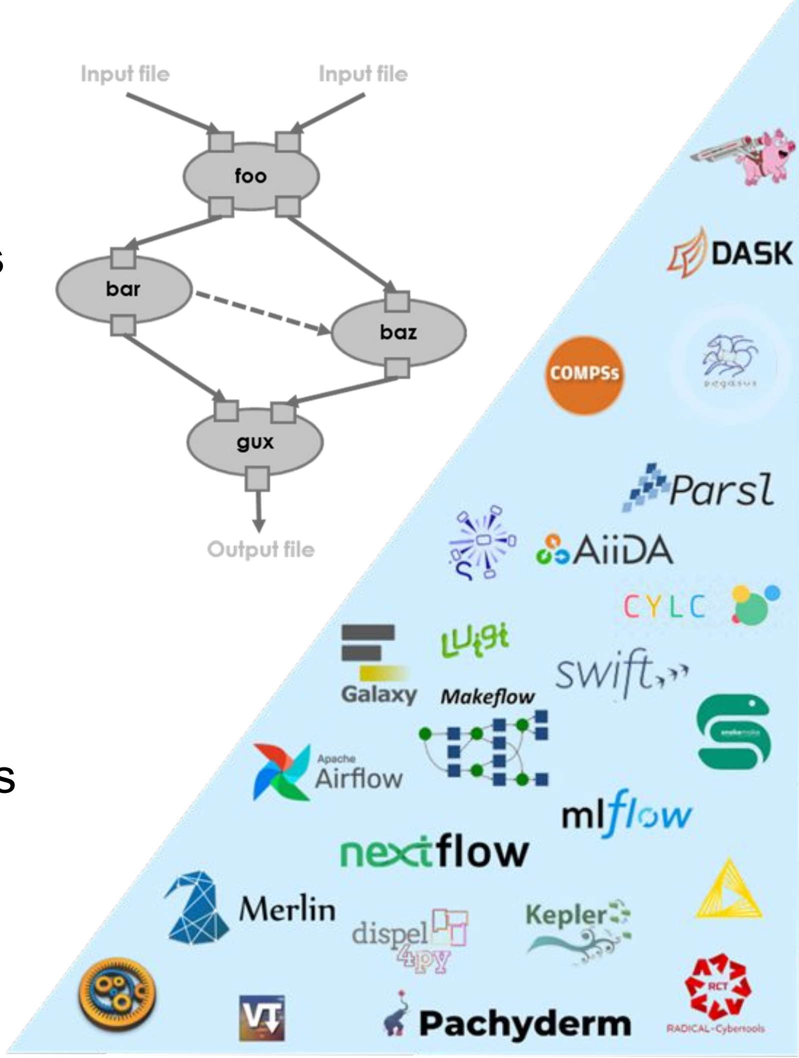
Frédéric Suter
Rafael Ferreira da Silva
Henri Casanova
Tainã Coleman

Classical Scientific Workflows

An abstract description of a scientific process

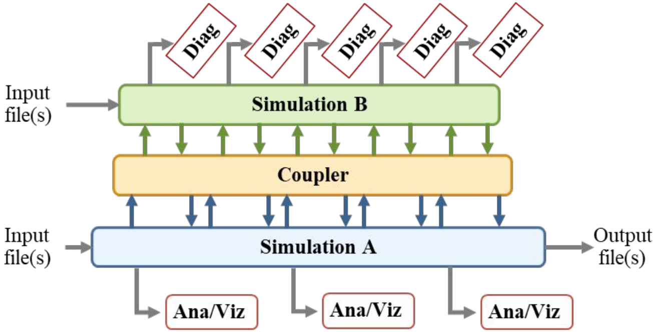
- ➢ **Directed Acyclic Graphs (DAGs)**
  - ○ **Tasks:** Functions, standalone kernels
  - ○ **Data:** file-based transfers
  - ○ **Dependencies:** Flow or control

- ➢ **1 Workflow = 1 Application**
  - ○ Well defined structure
  - ○ Full interoperability between components
  - ○ No intrusion in kernel codes
  - ○ Evolve as a whole

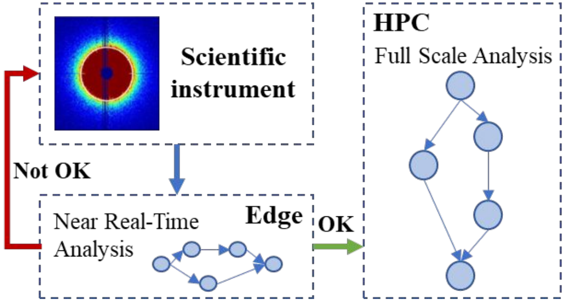


Modern Scientific Workflows

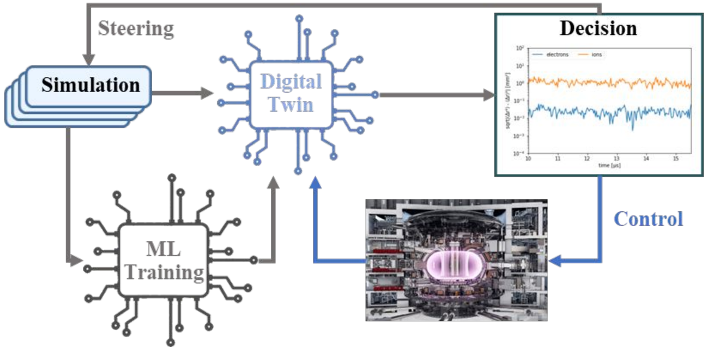
Strong Code Coupling and Analytics



Edge-to-HPC Multi-Stage Analysis



Digital Twins



Research Challenges

- Modern workflows bring new challenges
  - Dynamic loose coupling of components
  - Periodic data production/consumption
  - Cross-facility dimension
  - Near real-time constraints
  - Command-and-control
  - Coordination of AI and HPC

- In addition of the classical ones
  - Resource management
  - Data management
  - Scheduling and orchestration

- All require **experiments to evaluate the quality** of the proposed algorithms, systems, designs, …

WfCommons Problem Statement

https://wfcommons.org

- **Problem**: Not enough data (workflows) to run experiments and draw convincing conclusions
  - Just create more workflows!
    - Time/resources/expertise demand
    - Green computing, energy consumption

- **Solution**: create synthetic data (workflows) for experimentation
  - How?
  - Is it realistic?

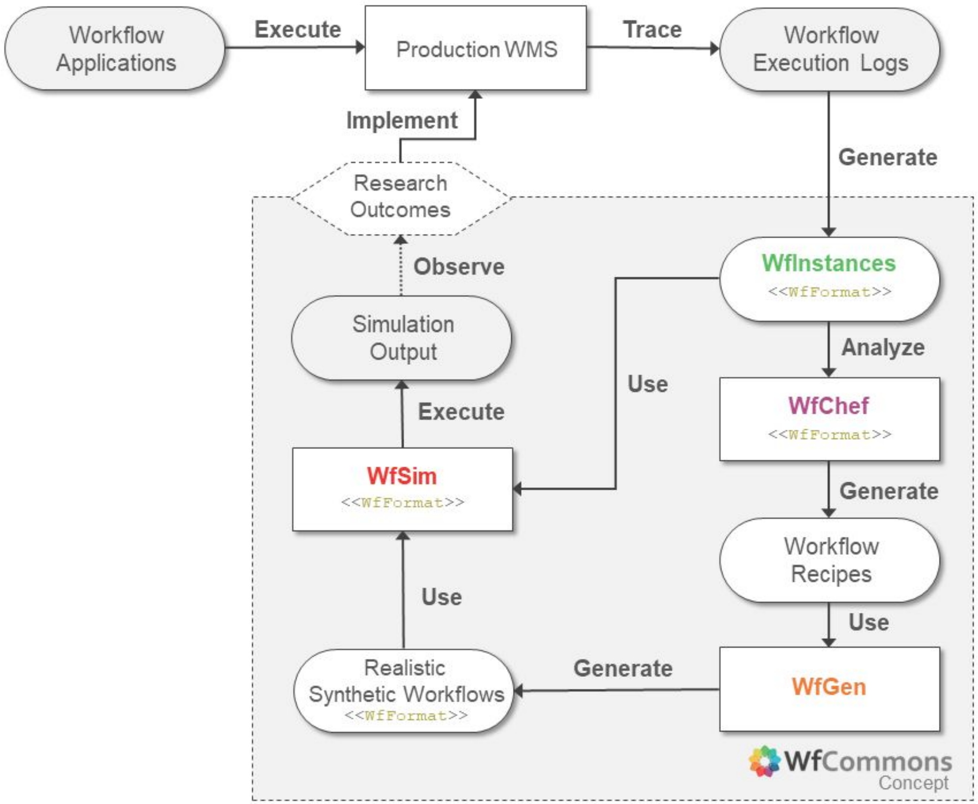
WfCommons

https://wfcommons.org

WfCommons is a framework that provides a collection of tools for analyzing **real workflow execution traces**, producing realistic **synthetic workflow execution traces**, and **benchmarking** / **simulating** workflow executions.

WfCommons: A framework for enabling scientific workflow research and development

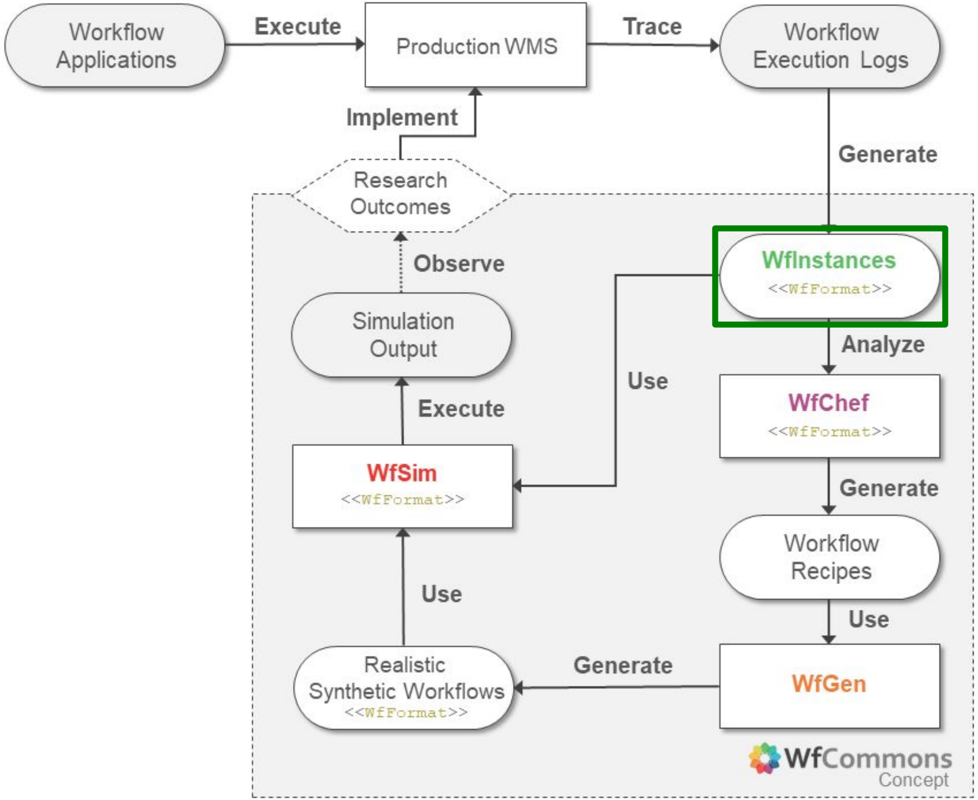
Coleman, T., Casanova, H., Pottier, L., Kaushik, M., Deelman, E., & da Silva, R. F. Future generation computer systems 128 (2022).



WfCommons

https://wfcommons.org

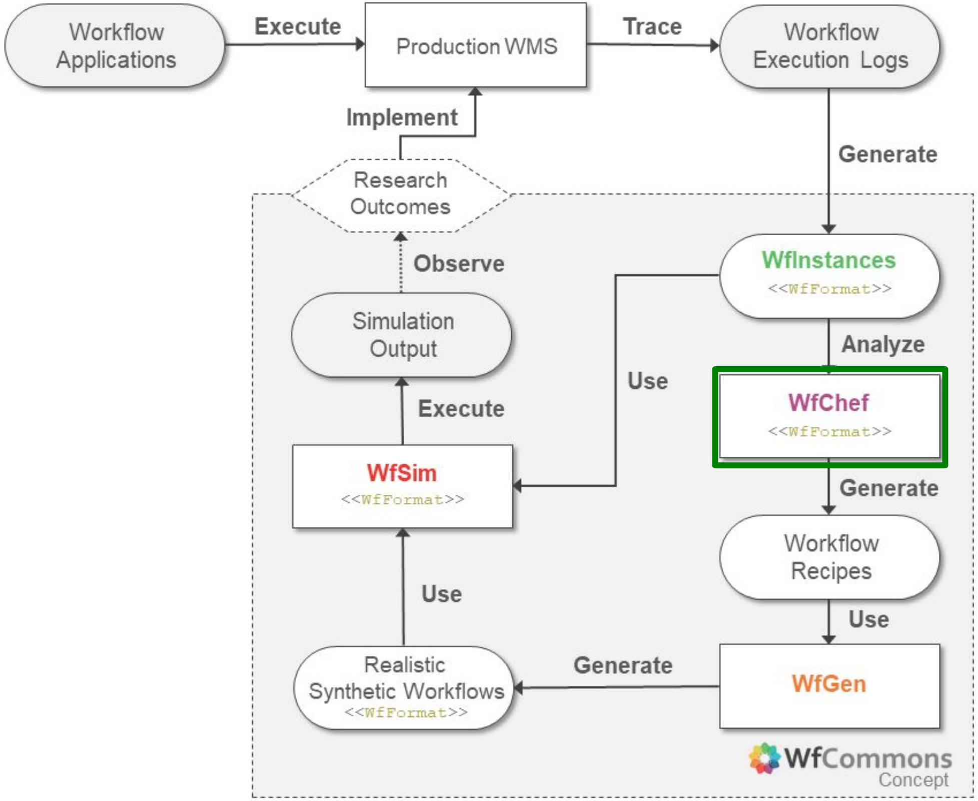
WfInstances: Workflow instances in a common format for representing workflow execution instances called **WfFormat**.



WfCommons

https://wfcommons.org

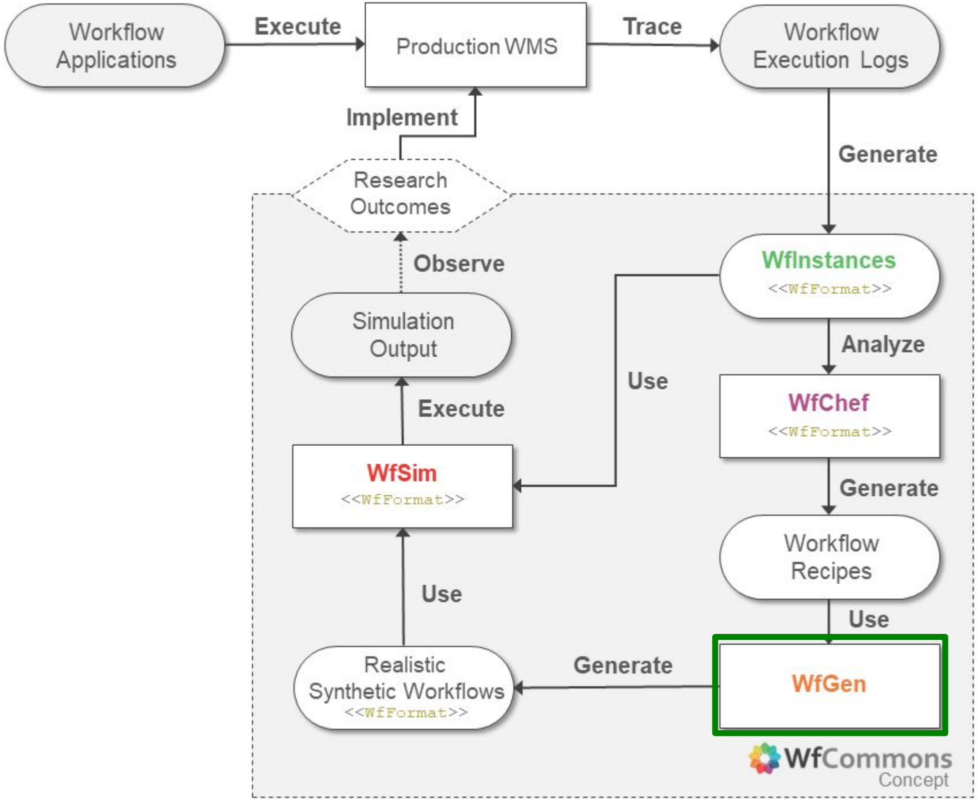
WfChef: Automated generator of realistic workflow generators



WfCommons

https://wfcommons.org

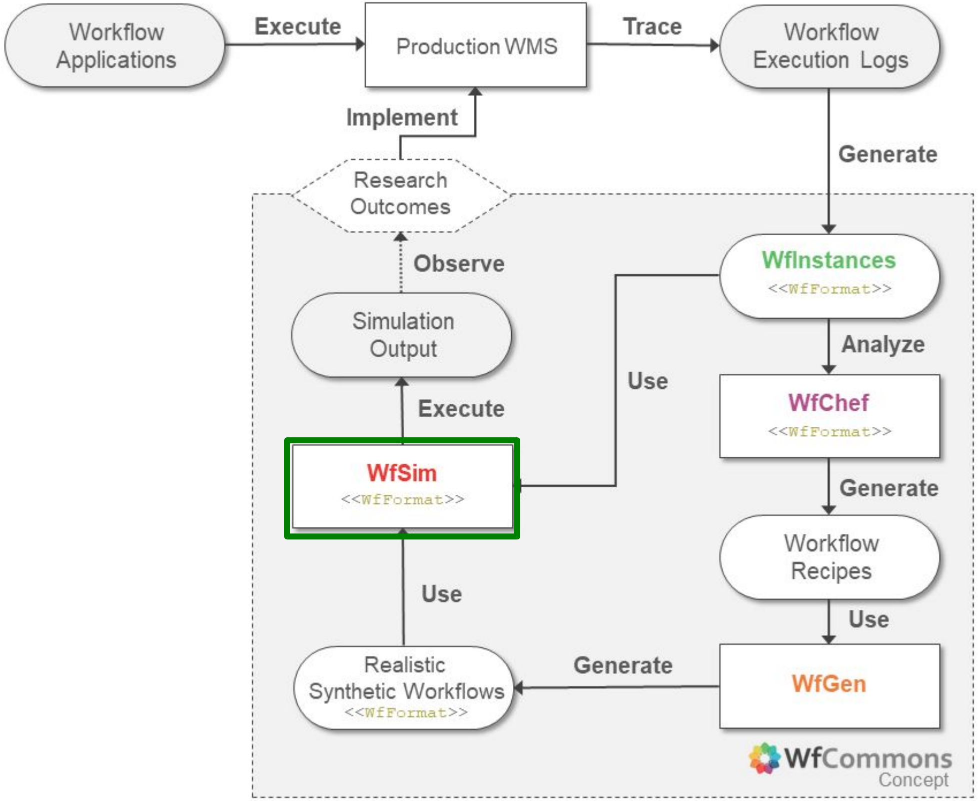
WfGen: Realistic workflow generator



WfCommons

https://wfcommons.org

WfSim: fosters the use of simulation for the development, evaluation, and verification of scheduling and resource provisioning algorithms



WfCommons Goal

https://wfcommons.org

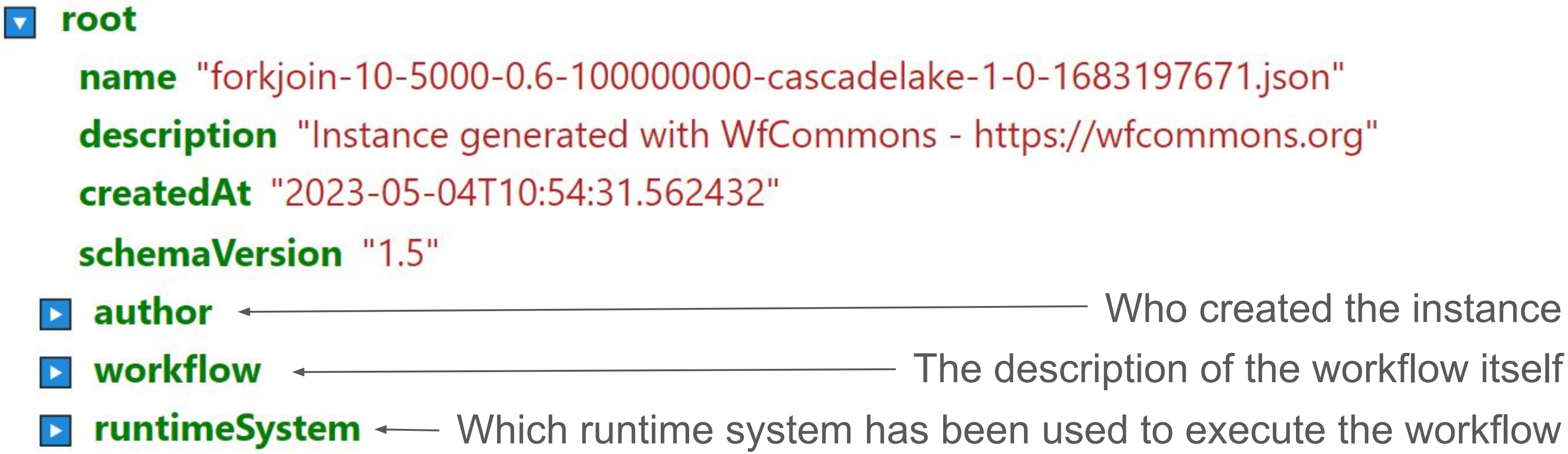
Enable and simplify the **exploratory research** and **benchmarking** of workflow applications

WfFormat
and
WfInstances

Actual workflow execution instances

WfFormat The WfCommons JSON Schema

- **Objective:** Ensure a seamless integration across frameworks and simulators
- **Proposition:** A universal JSON schema
  - https://github.com/wfcommons/WfFormat/blob/main/wfcommons-schema.json
  - Current version: 1.5 (since June 2023)
- **Overall structure**



▼ root

name "forkjoin-10-5000-0.6-100000000-cascadelake-1-0-1683197671.json"

description "Instance generated with WfCommons - https://wfcommons.org"

createdAt "2023-05-04T10:54:31.562432"

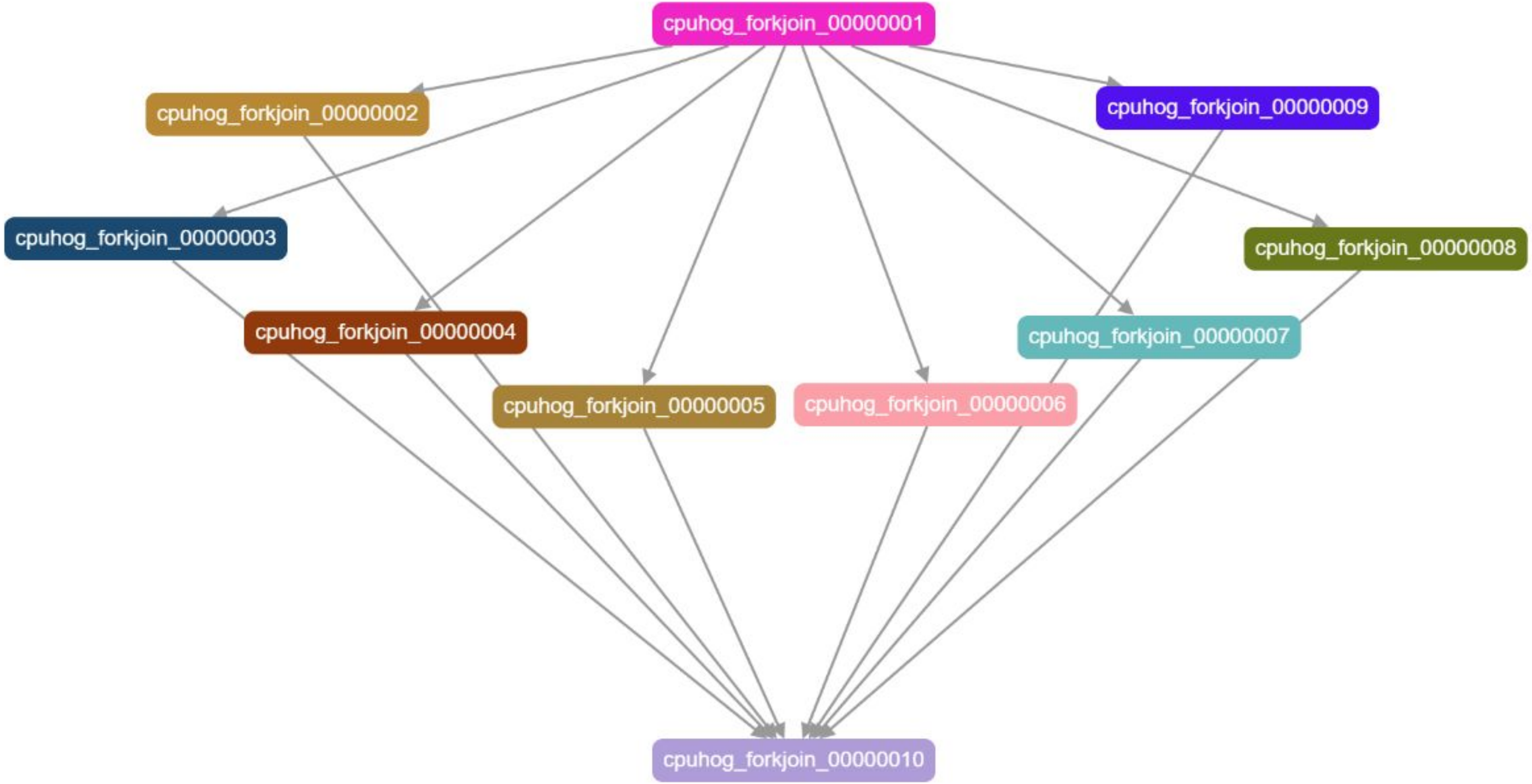
schemaVersion "1.5"

▶ author ← Who created the instance

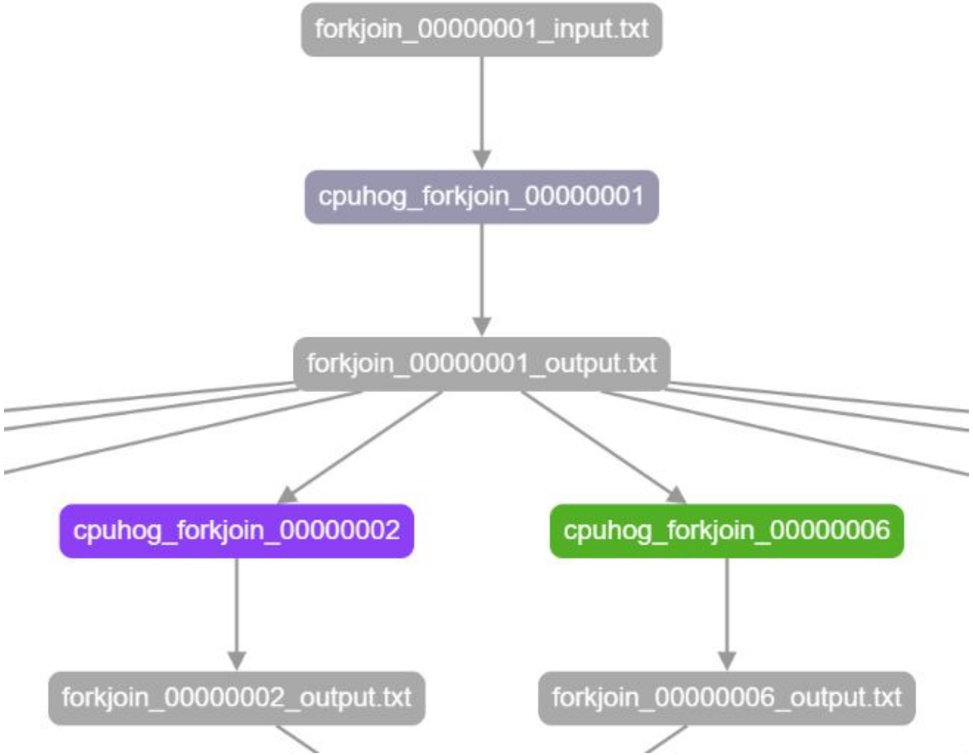
▶ workflow ← The description of the workflow itself

▶ runtimeSystem ← Which runtime system has been used to execute the workflow

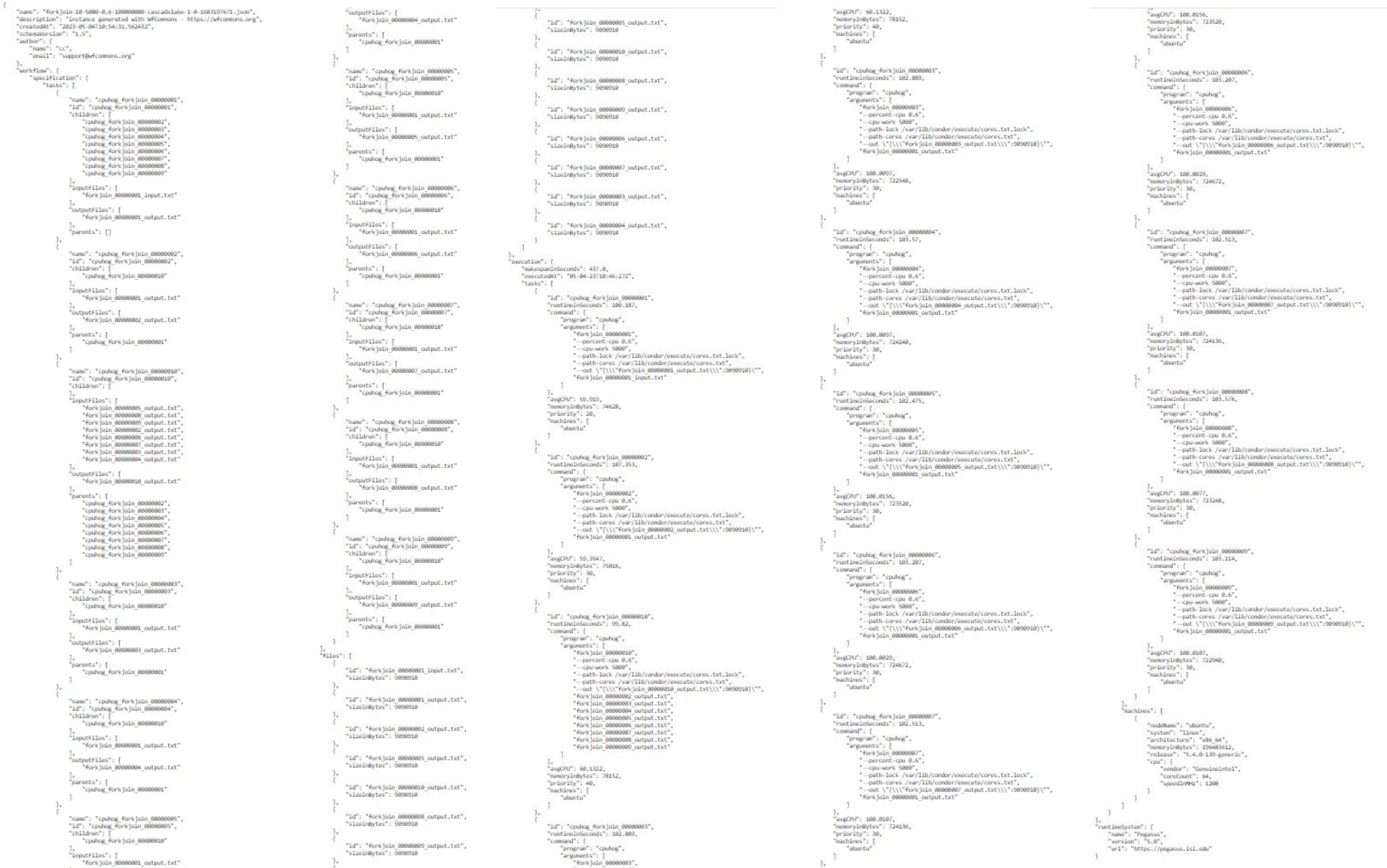
Simple Forkjoin Example Workflow



Simple Forkjoin Example Workflow


forkjoin_00000001_input.txt
cpuhog_forkjoin_00000001
forkjoin_00000001_output.txt
cpuhog_forkjoin_00000002
cpuhog_forkjoin_00000006
forkjoin_00000002_output.txt
forkjoin_00000006_output.txt

And its WfFormat Description …



```
{
    "name": "forkjoin-10-5000-0.6-100000000-cascadelake-1-0-1683197671.json",
    "description": "Instance generated with WfCommons - https://wfcommons.org",
    "createdAt": "2023-05-04T10:54:31.562432",
    "schemaVersion": "1.5",
    "author": {
        "name": "cc",
        "email": "support@wfcommons.org"
    },
    "workflow": {
        "specification": {
            "tasks": [
                {
                    "name": "cpuhog_forkjoin_00000001",
                    "id": "cpuhog_forkjoin_00000001",
                    "children": [
                        "cpuhog_forkjoin_00000002",
                        "cpuhog_forkjoin_00000003",
                        "cpuhog_forkjoin_00000004",
                        "cpuhog_forkjoin_00000005",
                        "cpuhog_forkjoin_00000006",
                        "cpuhog_forkjoin_00000007",
                        "cpuhog_forkjoin_00000008",
                        "cpuhog_forkjoin_00000009"
                    ],
                    "inputFiles": [
                        "forkjoin_00000001_input.txt"
                    ],
                    "outputFiles": [
                        "forkjoin_00000001_output.txt"
                    ],
                    "parents": []
                },
                {
                    "name": "cpuhog_forkjoin_00000002",
                    "id": "cpuhog_forkjoin_00000002",
                    "children": [
                        "cpuhog_forkjoin_00000010"
                    ],
                    "inputFiles": [
                        "forkjoin_00000001_output.txt"
                    ],
                    "outputFiles": [
                        "forkjoin_00000002_output.txt"
                    ],
                    "parents": [
                        "cpuhog_forkjoin_00000001"
                    ]
                },
                {
                    "name": "cpuhog_forkjoin_00000010",
                    "id": "cpuhog_forkjoin_00000010",
                    "children": [
                    ],
                    "inputFiles": [
                        "forkjoin_00000005_output.txt",
                        "forkjoin_00000008_output.txt",
                        "forkjoin_00000009_output.txt",
                        "forkjoin_00000002_output.txt",
                        "forkjoin_00000006_output.txt",
                        "forkjoin_00000007_output.txt",
                        "forkjoin_00000003_output.txt",
                        "forkjoin_00000004_output.txt"
                    ],
                    "outputFiles": [
                        "forkjoin_00000010_output.txt"
                    ],
                    "parents": [
                        "cpuhog_forkjoin_00000002",
                        "cpuhog_forkjoin_00000003",
                        "cpuhog_forkjoin_00000004",
                        "cpuhog_forkjoin_00000005",
                        "cpuhog_forkjoin_00000006",
                        "cpuhog_forkjoin_00000007",
                        "cpuhog_forkjoin_00000008",
                        "cpuhog_forkjoin_00000009"
                    ]
                },
                {
                    "name": "cpuhog_forkjoin_00000003",
                    "id": "cpuhog_forkjoin_00000003",
                    "children": [
                        "cpuhog_forkjoin_00000010"
                    ],
                    "inputFiles": [
                        "forkjoin_00000001_output.txt"
                    ],
                    "outputFiles": [
                        "forkjoin_00000003_output.txt"
                    ],
                    "parents": [
                        "cpuhog_forkjoin_00000001"
                    ]
                },
                {
                    "name": "cpuhog_forkjoin_00000004",
                    "id": "cpuhog_forkjoin_00000004",
                    "children": [
                        "cpuhog_forkjoin_00000010"
                    ],
                    "inputFiles": [
                        "forkjoin_00000001_output.txt"
                    ],
                    "outputFiles": [
                        "forkjoin_00000004_output.txt"
                    ],
                    "parents": [
                        "cpuhog_forkjoin_00000001"
                    ]
                },
                {
                    "name": "cpuhog_forkjoin_00000005",
                    "id": "cpuhog_forkjoin_00000005",
                    "children": [
                        "cpuhog_forkjoin_00000010"
                    ],
                    "inputFiles": [
                        "forkjoin_00000001_output.txt"
```

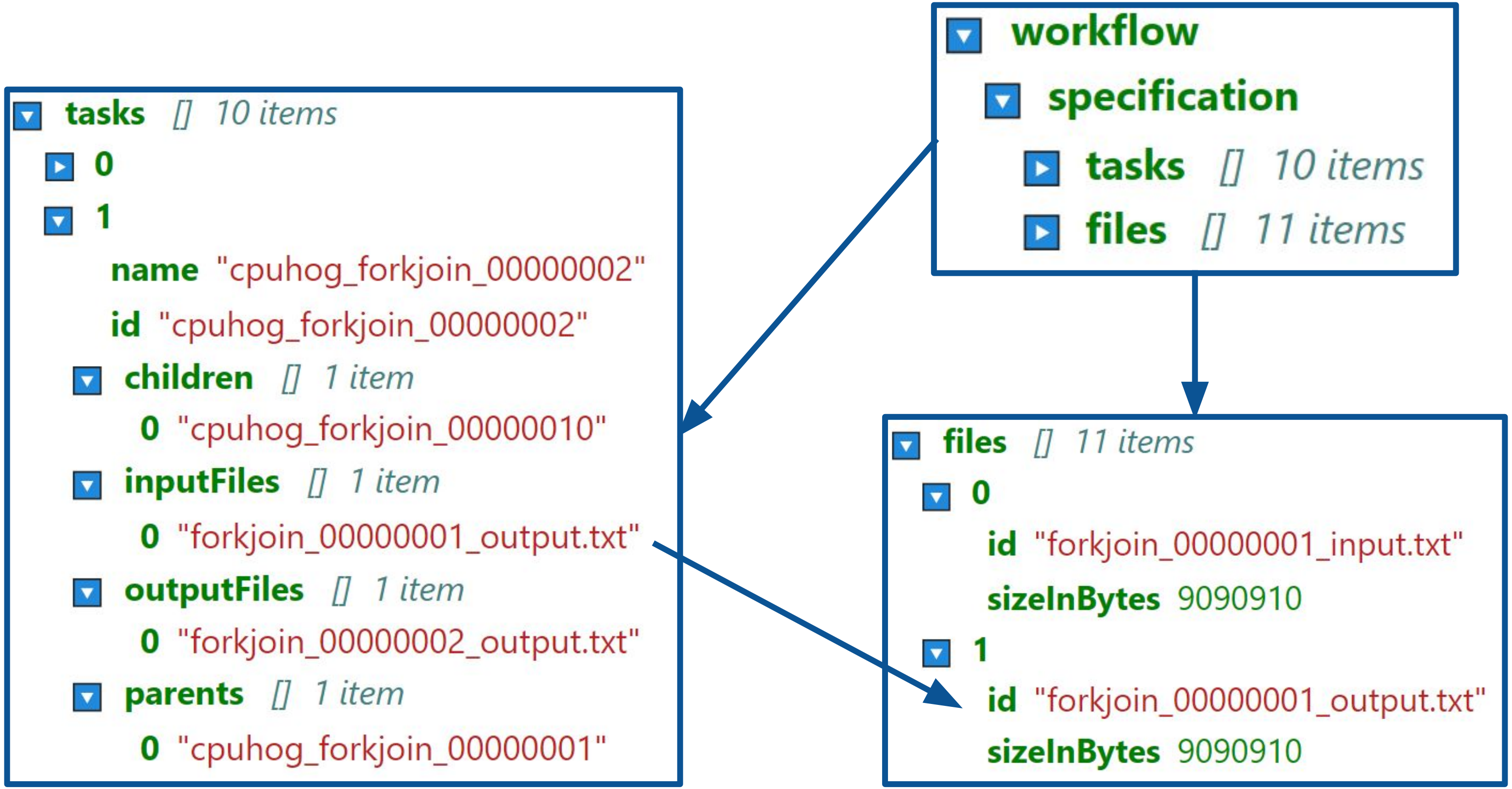
```
                    ],
                    "outputFiles": [
                        "forkjoin_00000004_output.txt"
                    ],
                    "parents": [
                        "cpuhog_forkjoin_00000001"
                    ]
                },
                {
                    "name": "cpuhog_forkjoin_00000005",
                    "id": "cpuhog_forkjoin_00000005",
                    "children": [
                        "cpuhog_forkjoin_00000010"
                    ],
                    "inputFiles": [
                        "forkjoin_00000001_output.txt"
                    ],
                    "outputFiles": [
                        "forkjoin_00000005_output.txt"
                    ],
                    "parents": [
                        "cpuhog_forkjoin_00000001"
                    ]
                },
                {
                    "name": "cpuhog_forkjoin_00000006",
                    "id": "cpuhog_forkjoin_00000006",
                    "children": [
                        "cpuhog_forkjoin_00000010"
                    ],
                    "inputFiles": [
                        "forkjoin_00000001_output.txt"
                    ],
                    "outputFiles": [
                        "forkjoin_00000006_output.txt"
                    ],
                    "parents": [
                        "cpuhog_forkjoin_00000001"
                    ]
                },
                {
                    "name": "cpuhog_forkjoin_00000007",
                    "id": "cpuhog_forkjoin_00000007",
                    "children": [
                        "cpuhog_forkjoin_00000010"
                    ],
                    "inputFiles": [
                        "forkjoin_00000001_output.txt"
                    ],
                    "outputFiles": [
                        "forkjoin_00000007_output.txt"
                    ],
                    "parents": [
                        "cpuhog_forkjoin_00000001"
                    ]
                },
                {
                    "name": "cpuhog_forkjoin_00000008",
                    "id": "cpuhog_forkjoin_00000008",
                    "children": [
                        "cpuhog_forkjoin_00000010"
                    ],
                    "inputFiles": [
                        "forkjoin_00000001_output.txt"
                    ],
                    "outputFiles": [
                        "forkjoin_00000008_output.txt"
                    ],
                    "parents": [
                        "cpuhog_forkjoin_00000001"
                    ]
                },
                {
                    "name": "cpuhog_forkjoin_00000009",
                    "id": "cpuhog_forkjoin_00000009",
                    "children": [
                        "cpuhog_forkjoin_00000010"
                    ],
                    "inputFiles": [
                        "forkjoin_00000001_output.txt"
                    ],
                    "outputFiles": [
                        "forkjoin_00000009_output.txt"
                    ],
                    "parents": [
                        "cpuhog_forkjoin_00000001"
                    ]
                }
            ],
            "files": [
                {
                    "id": "forkjoin_00000001_input.txt",
                    "sizeInBytes": 9090910
                },
                {
                    "id": "forkjoin_00000001_output.txt",
                    "sizeInBytes": 9090910
                },
                {
                    "id": "forkjoin_00000002_output.txt",
                    "sizeInBytes": 9090910
                },
                {
                    "id": "forkjoin_00000005_output.txt",
                    "sizeInBytes": 9090910
                },
                {
                    "id": "forkjoin_00000010_output.txt",
                    "sizeInBytes": 9090910
                },
                {
                    "id": "forkjoin_00000008_output.txt",
                    "sizeInBytes": 9090910
                },
                {
                    "id": "forkjoin_00000009_output.txt",
                    "sizeInBytes": 9090910
                },
                {
```

```
                },
                {
                    "id": "forkjoin_00000005_output.txt",
                    "sizeInBytes": 9090910
                },
                {
                    "id": "forkjoin_00000010_output.txt",
                    "sizeInBytes": 9090910
                },
                {
                    "id": "forkjoin_00000008_output.txt",
                    "sizeInBytes": 9090910
                },
                {
                    "id": "forkjoin_00000009_output.txt",
                    "sizeInBytes": 9090910
                },
                {
                    "id": "forkjoin_00000006_output.txt",
                    "sizeInBytes": 9090910
                },
                {
                    "id": "forkjoin_00000007_output.txt",
                    "sizeInBytes": 9090910
                },
                {
                    "id": "forkjoin_00000003_output.txt",
                    "sizeInBytes": 9090910
                },
                {
                    "id": "forkjoin_00000004_output.txt",
                    "sizeInBytes": 9090910
                }
            ]
        },
        "execution": {
            "makespanInSeconds": 437.0,
            "executedAt": "05-04-23T10:46:272",
            "tasks": [
                {
                    "id": "cpuhog_forkjoin_00000001",
                    "runtimeInSeconds": 100.187,
                    "command": {
                        "program": "cpuhog",
                        "arguments": [
                            "forkjoin_00000001",
                            "--percent-cpu 0.6",
                            "--cpu-work 5000",
                            "--path-lock /var/lib/condor/execute/cores.txt.lock",
                            "--path-cores /var/lib/condor/execute/cores.txt",
                            "--out \"{\\\"forkjoin_00000001_output.txt\\\":9090910}\"",
                            "forkjoin_00000001_input.txt"
                        ]
                    },
                    "avgCPU": 59.919,
                    "memoryInBytes": 74628,
                    "priority": 20,
                    "machines": [
                        "ubuntu"
                    ]
                },
                {
                    "id": "cpuhog_forkjoin_00000002",
                    "runtimeInSeconds": 107.353,
                    "command": {
                        "program": "cpuhog",
                        "arguments": [
                            "forkjoin_00000002",
                            "--percent-cpu 0.6",
                            "--cpu-work 5000",
                            "--path-lock /var/lib/condor/execute/cores.txt.lock",
                            "--path-cores /var/lib/condor/execute/cores.txt",
                            "--out \"{\\\"forkjoin_00000002_output.txt\\\":9090910}\"",
                            "forkjoin_00000001_output.txt"
                        ]
                    },
                    "avgCPU": 59.3547,
                    "memoryInBytes": 75016,
                    "priority": 30,
                    "machines": [
                        "ubuntu"
                    ]
                },
                {
                    "id": "cpuhog_forkjoin_00000010",
                    "runtimeInSeconds": 99.82,
                    "command": {
                        "program": "cpuhog",
                        "arguments": [
                            "forkjoin_00000010",
                            "--percent-cpu 0.6",
                            "--cpu-work 5000",
                            "--path-lock /var/lib/condor/execute/cores.txt.lock",
                            "--path-cores /var/lib/condor/execute/cores.txt",
                            "--out \"{\\\"forkjoin_00000010_output.txt\\\":9090910}\"",
                            "forkjoin_00000002_output.txt",
                            "forkjoin_00000003_output.txt",
                            "forkjoin_00000004_output.txt",
                            "forkjoin_00000005_output.txt",
                            "forkjoin_00000006_output.txt",
                            "forkjoin_00000007_output.txt",
                            "forkjoin_00000008_output.txt",
                            "forkjoin_00000009_output.txt"
                        ]
                    },
                    "avgCPU": 60.1322,
                    "memoryInBytes": 78152,
                    "priority": 40,
                    "machines": [
                        "ubuntu"
                    ]
                },
                {
                    "id": "cpuhog_forkjoin_00000003",
                    "runtimeInSeconds": 102.889,
                    "command": {
                        "program": "cpuhog",
                        "arguments": [
                            "forkjoin_00000003",
```

```
                    "avgCPU": 60.1322,
                    "memoryInBytes": 78152,
                    "priority": 40,
                    "machines": [
                        "ubuntu"
                    ]
                },
                {
                    "id": "cpuhog_forkjoin_00000003",
                    "runtimeInSeconds": 102.889,
                    "command": {
                        "program": "cpuhog",
                        "arguments": [
                            "forkjoin_00000003",
                            "--percent-cpu 0.6",
                            "--cpu-work 5000",
                            "--path-lock /var/lib/condor/execute/cores.txt.lock",
                            "--path-cores /var/lib/condor/execute/cores.txt",
                            "--out \"{\\\"forkjoin_00000003_output.txt\\\":9090910}\"",
                            "forkjoin_00000001_output.txt"
                        ]
                    },
                    "avgCPU": 100.0097,
                    "memoryInBytes": 722548,
                    "priority": 30,
                    "machines": [
                        "ubuntu"
                    ]
                },
                {
                    "id": "cpuhog_forkjoin_00000004",
                    "runtimeInSeconds": 103.57,
                    "command": {
                        "program": "cpuhog",
                        "arguments": [
                            "forkjoin_00000004",
                            "--percent-cpu 0.6",
                            "--cpu-work 5000",
                            "--path-lock /var/lib/condor/execute/cores.txt.lock",
                            "--path-cores /var/lib/condor/execute/cores.txt",
                            "--out \"{\\\"forkjoin_00000004_output.txt\\\":9090910}\"",
                            "forkjoin_00000001_output.txt"
                        ]
                    },
                    "avgCPU": 100.0097,
                    "memoryInBytes": 724240,
                    "priority": 30,
                    "machines": [
                        "ubuntu"
                    ]
                },
                {
                    "id": "cpuhog_forkjoin_00000005",
                    "runtimeInSeconds": 102.475,
                    "command": {
                        "program": "cpuhog",
                        "arguments": [
                            "forkjoin_00000005",
                            "--percent-cpu 0.6",
                            "--cpu-work 5000",
                            "--path-lock /var/lib/condor/execute/cores.txt.lock",
                            "--path-cores /var/lib/condor/execute/cores.txt",
                            "--out \"{\\\"forkjoin_00000005_output.txt\\\":9090910}\"",
                            "forkjoin_00000001_output.txt"
                        ]
                    },
                    "avgCPU": 100.0156,
                    "memoryInBytes": 723520,
                    "priority": 30,
                    "machines": [
                        "ubuntu"
                    ]
                },
                {
                    "id": "cpuhog_forkjoin_00000006",
                    "runtimeInSeconds": 103.207,
                    "command": {
                        "program": "cpuhog",
                        "arguments": [
                            "forkjoin_00000006",
                            "--percent-cpu 0.6",
                            "--cpu-work 5000",
                            "--path-lock /var/lib/condor/execute/cores.txt.lock",
                            "--path-cores /var/lib/condor/execute/cores.txt",
                            "--out \"{\\\"forkjoin_00000006_output.txt\\\":9090910}\"",
                            "forkjoin_00000001_output.txt"
                        ]
                    },
                    "avgCPU": 100.0029,
                    "memoryInBytes": 724672,
                    "priority": 30,
                    "machines": [
                        "ubuntu"
                    ]
                },
                {
                    "id": "cpuhog_forkjoin_00000007",
                    "runtimeInSeconds": 102.513,
                    "command": {
                        "program": "cpuhog",
                        "arguments": [
                            "forkjoin_00000007",
                            "--percent-cpu 0.6",
                            "--cpu-work 5000",
                            "--path-lock /var/lib/condor/execute/cores.txt.lock",
                            "--path-cores /var/lib/condor/execute/cores.txt",
                            "--out \"{\\\"forkjoin_00000007_output.txt\\\":9090910}\"",
                            "forkjoin_00000001_output.txt"
                        ]
                    },
                    "avgCPU": 100.0107,
                    "memoryInBytes": 724136,
                    "priority": 30,
                    "machines": [
                        "ubuntu"
                    ]
                },
```

```
                    },
                    "avgCPU": 100.0156,
                    "memoryInBytes": 723520,
                    "priority": 30,
                    "machines": [
                        "ubuntu"
                    ]
                },
                {
                    "id": "cpuhog_forkjoin_00000006",
                    "runtimeInSeconds": 103.207,
                    "command": {
                        "program": "cpuhog",
                        "arguments": [
                            "forkjoin_00000006",
                            "--percent-cpu 0.6",
                            "--cpu-work 5000",
                            "--path-lock /var/lib/condor/execute/cores.txt.lock",
                            "--path-cores /var/lib/condor/execute/cores.txt",
                            "--out \"{\\\"forkjoin_00000006_output.txt\\\":9090910}\"",
                            "forkjoin_00000001_output.txt"
                        ]
                    },
                    "avgCPU": 100.0029,
                    "memoryInBytes": 724672,
                    "priority": 30,
                    "machines": [
                        "ubuntu"
                    ]
                },
                {
                    "id": "cpuhog_forkjoin_00000007",
                    "runtimeInSeconds": 102.513,
                    "command": {
                        "program": "cpuhog",
                        "arguments": [
                            "forkjoin_00000007",
                            "--percent-cpu 0.6",
                            "--cpu-work 5000",
                            "--path-lock /var/lib/condor/execute/cores.txt.lock",
                            "--path-cores /var/lib/condor/execute/cores.txt",
                            "--out \"{\\\"forkjoin_00000007_output.txt\\\":9090910}\"",
                            "forkjoin_00000001_output.txt"
                        ]
                    },
                    "avgCPU": 100.0107,
                    "memoryInBytes": 724136,
                    "priority": 30,
                    "machines": [
                        "ubuntu"
                    ]
                },
                {
                    "id": "cpuhog_forkjoin_00000008",
                    "runtimeInSeconds": 103.576,
                    "command": {
                        "program": "cpuhog",
                        "arguments": [
                            "forkjoin_00000008",
                            "--percent-cpu 0.6",
                            "--cpu-work 5000",
                            "--path-lock /var/lib/condor/execute/cores.txt.lock",
                            "--path-cores /var/lib/condor/execute/cores.txt",
                            "--out \"{\\\"forkjoin_00000008_output.txt\\\":9090910}\"",
                            "forkjoin_00000001_output.txt"
                        ]
                    },
                    "avgCPU": 100.0077,
                    "memoryInBytes": 723248,
                    "priority": 30,
                    "machines": [
                        "ubuntu"
                    ]
                },
                {
                    "id": "cpuhog_forkjoin_00000009",
                    "runtimeInSeconds": 103.114,
                    "command": {
                        "program": "cpuhog",
                        "arguments": [
                            "forkjoin_00000009",
                            "--percent-cpu 0.6",
                            "--cpu-work 5000",
                            "--path-lock /var/lib/condor/execute/cores.txt.lock",
                            "--path-cores /var/lib/condor/execute/cores.txt",
                            "--out \"{\\\"forkjoin_00000009_output.txt\\\":9090910}\"",
                            "forkjoin_00000001_output.txt"
                        ]
                    },
                    "avgCPU": 100.0107,
                    "memoryInBytes": 722940,
                    "priority": 30,
                    "machines": [
                        "ubuntu"
                    ]
                }
            ],
            "machines": [
                {
                    "nodeName": "ubuntu",
                    "system": "linux",
                    "architecture": "x86_64",
                    "memoryInBytes": 196483612,
                    "release": "5.4.0-139-generic",
                    "cpu": {
                        "vendor": "GenuineIntel",
                        "coreCount": 64,
                        "speedInMHz": 1200
                    }
                }
            ]
        }
    },
    "runtimeSystem": {
        "name": "Pegasus",
        "version": "5.0",
        "url": "https://pegasus.isi.edu"
    }
```

WfFormat Workflow Specification

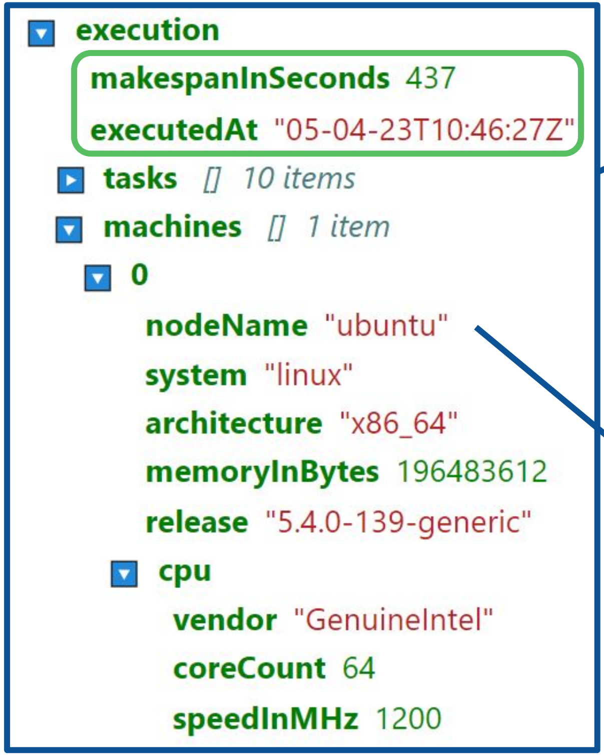


```
▼ workflow
  ▼ specification
    ▶ tasks   []  10 items
    ▶ files   []  11 items
```

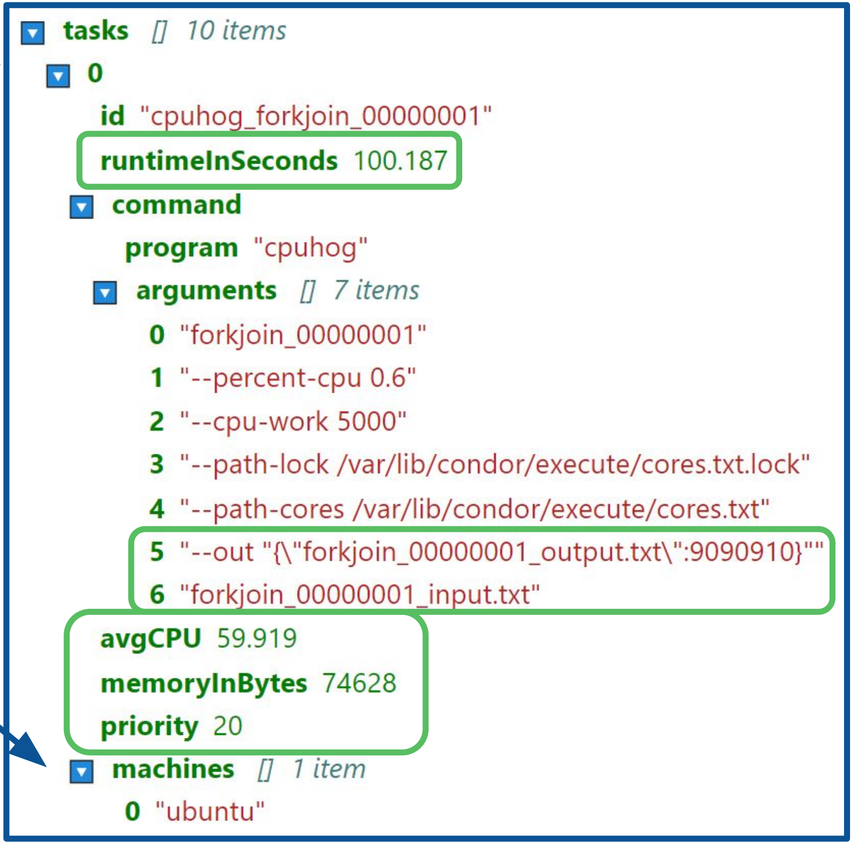
```
▼ tasks  []  10 items
  ▶ 0
  ▼ 1
      name  "cpuhog_forkjoin_00000002"
      id  "cpuhog_forkjoin_00000002"
    ▼ children  []  1 item
        0  "cpuhog_forkjoin_00000010"
    ▼ inputFiles  []  1 item
        0  "forkjoin_00000001_output.txt"
    ▼ outputFiles  []  1 item
        0  "forkjoin_00000002_output.txt"
    ▼ parents  []  1 item
        0  "cpuhog_forkjoin_00000001"
```

```
▼ files  []  11 items
  ▼ 0
      id  "forkjoin_00000001_input.txt"
      sizeInBytes  9090910
  ▼ 1
      id  "forkjoin_00000001_output.txt"
      sizeInBytes  9090910
```

WfFormat Workflow Execution



```
▼ execution
    makespanInSeconds 437
    executedAt "05-04-23T10:46:27Z"
  ▶ tasks [] 10 items
  ▼ machines [] 1 item
    ▼ 0
        nodeName "ubuntu"
        system "linux"
        architecture "x86_64"
        memoryInBytes 196483612
        release "5.4.0-139-generic"
      ▼ cpu
          vendor "GenuineIntel"
          coreCount 64
          speedInMHz 1200
```

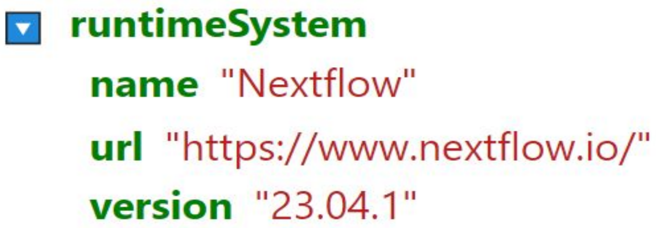


```
▼ tasks [] 10 items
  ▼ 0
      id "cpuhog_forkjoin_00000001"
      runtimeInSeconds 100.187
    ▼ command
        program "cpuhog"
      ▼ arguments [] 7 items
          0 "forkjoin_00000001"
          1 "--percent-cpu 0.6"
          2 "--cpu-work 5000"
          3 "--path-lock /var/lib/condor/execute/cores.txt.lock"
          4 "--path-cores /var/lib/condor/execute/cores.txt"
          5 "--out "{\"forkjoin_00000001_output.txt\":9090910}""
          6 "forkjoin_00000001_input.txt"
      avgCPU 59.919
      memoryInBytes 74628
      priority 20
    ▼ machines [] 1 item
        0 "ubuntu"
```

WfFormat Runtime System

➢ Several runtime systems can export execution traces in WfFormat

nextflow

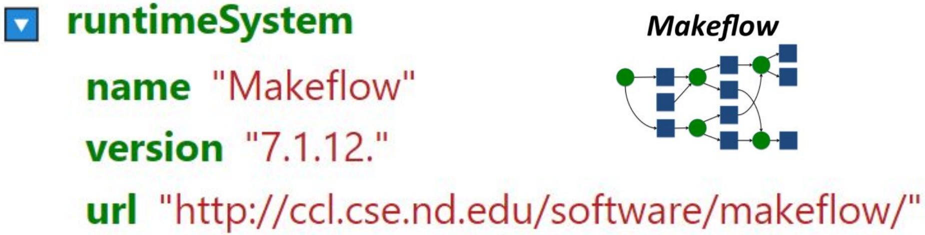


▾ runtimeSystem

name "Nextflow"

url "https://www.nextflow.io/"

version "23.04.1"



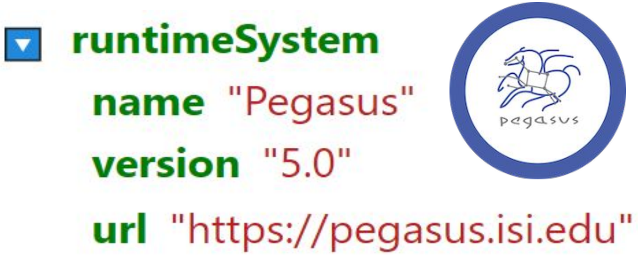
▾ runtimeSystem

name "Makeflow"

version "7.1.12."

url "http://ccl.cse.nd.edu/software/makeflow/"

Makeflow



▾ runtimeSystem

name "Pegasus"

version "5.0"

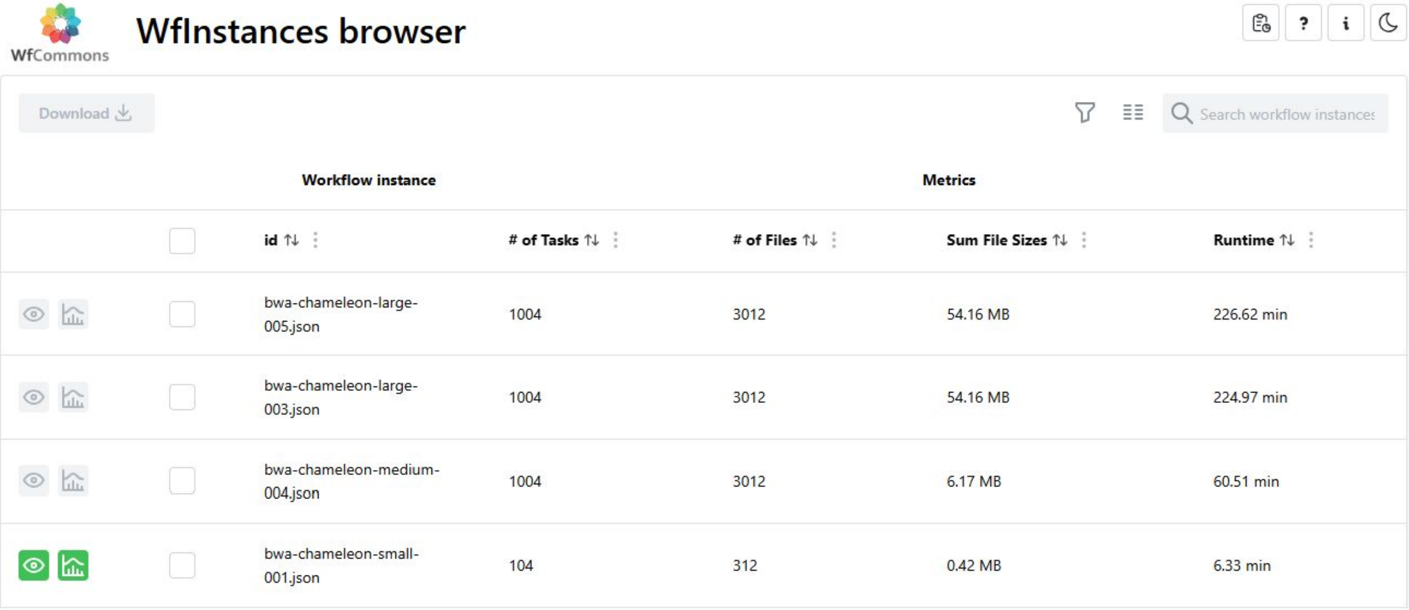
url "https://pegasus.isi.edu"

WfInstances

- A Collection of **curated** open access **production workflow executions** from **various scientific applications**.
  - https://github.com/wfcommons/WfInstances/
- Obtained with **three workflow runtime systems**
  - Makeflow, nextflow, and Pegasus
- **24 scientific workflows** from different scientific domains
  - Astronomy, AgroEconomics, Bioinformatics, and Seismology
- A total of **180 instances**
  - Composed of from **11 to 9,805 tasks**
  - Running in from **1.2 minutes to 42.9 hours**
  - Handling up to nearly **52k files**
  - **Compute-** or **Data-intensive**

WfInstances Browser Web Application

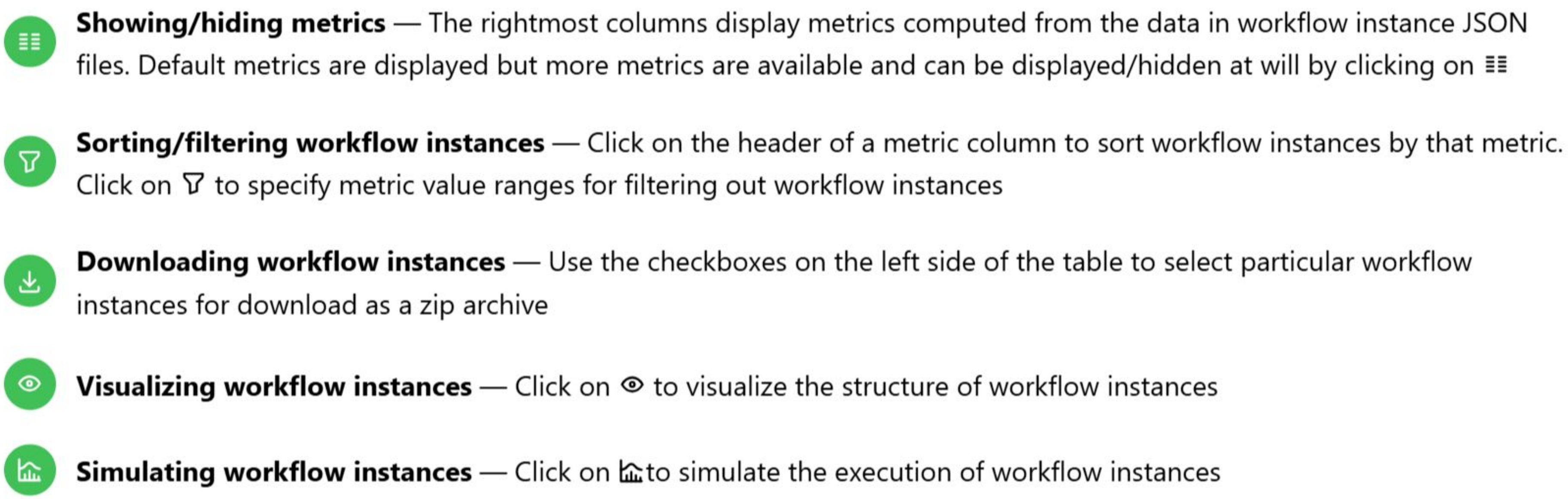
- ➢ Develop at the University of Hawaii at Manoa
  - ○ WfInstances browser
- ➢ Periodically synchronized with the git repository



WfInstances browser

	Workflow instance		Metrics		
□	id	# of Tasks	# of Files	Sum File Sizes	Runtime
□	bwa-chameleon-large-005.json	1004	3012	54.16 MB	226.62 min
□	bwa-chameleon-large-003.json	1004	3012	54.16 MB	224.97 min
□	bwa-chameleon-medium-004.json	1004	3012	6.17 MB	60.51 min
□	bwa-chameleon-small-001.json	104	312	0.42 MB	6.33 min

WfInstances Browser Help



- **Showing/hiding metrics** — The rightmost columns display metrics computed from the data in workflow instance JSON files. Default metrics are displayed but more metrics are available and can be displayed/hidden at will by clicking on ≣

- **Sorting/filtering workflow instances** — Click on the header of a metric column to sort workflow instances by that metric. Click on ▽ to specify metric value ranges for filtering out workflow instances

- **Downloading workflow instances** — Use the checkboxes on the left side of the table to select particular workflow instances for download as a zip archive

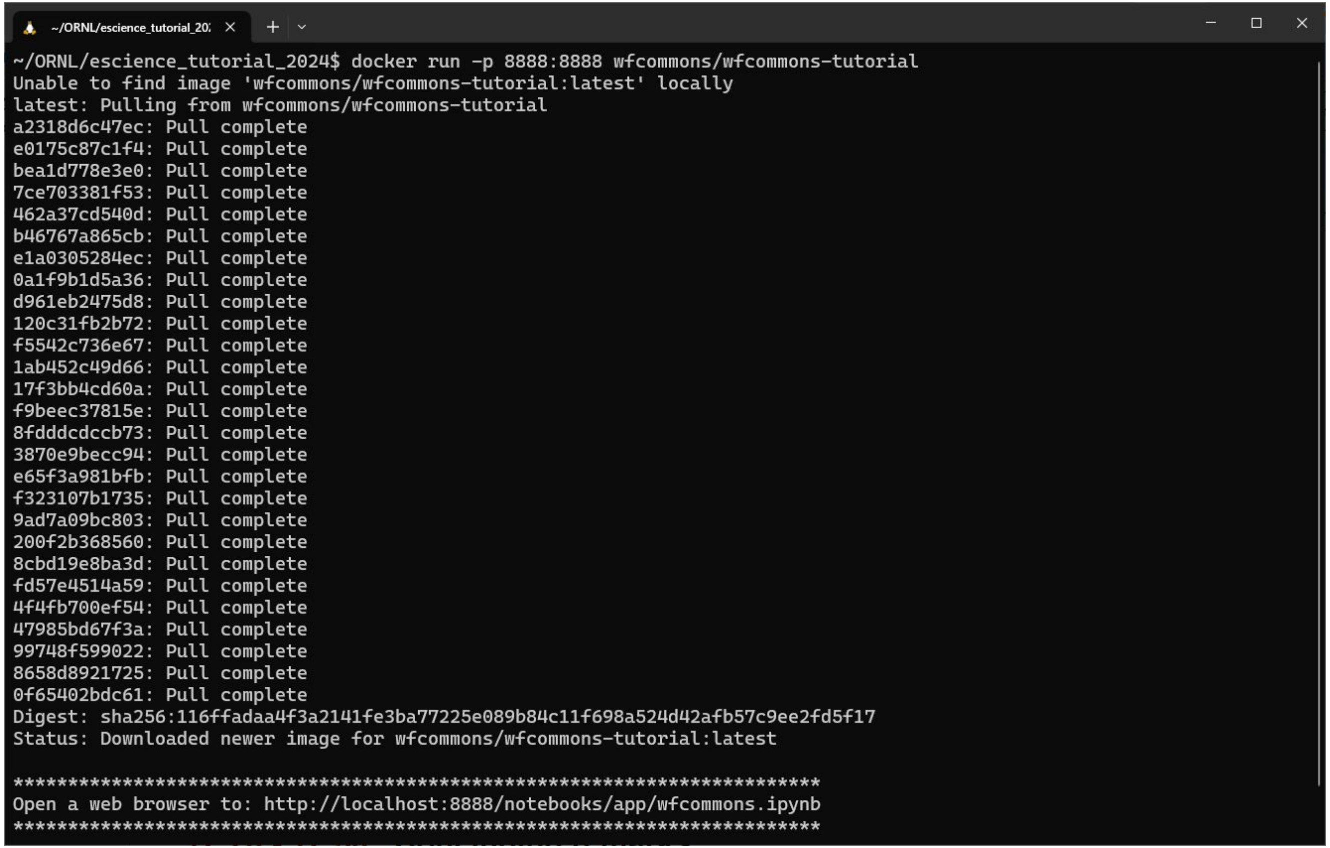
- **Visualizing workflow instances** — Click on 👁 to visualize the structure of workflow instances

- **Simulating workflow instances** — Click on 📈to simulate the execution of workflow instances

WfFormat and WfInstances

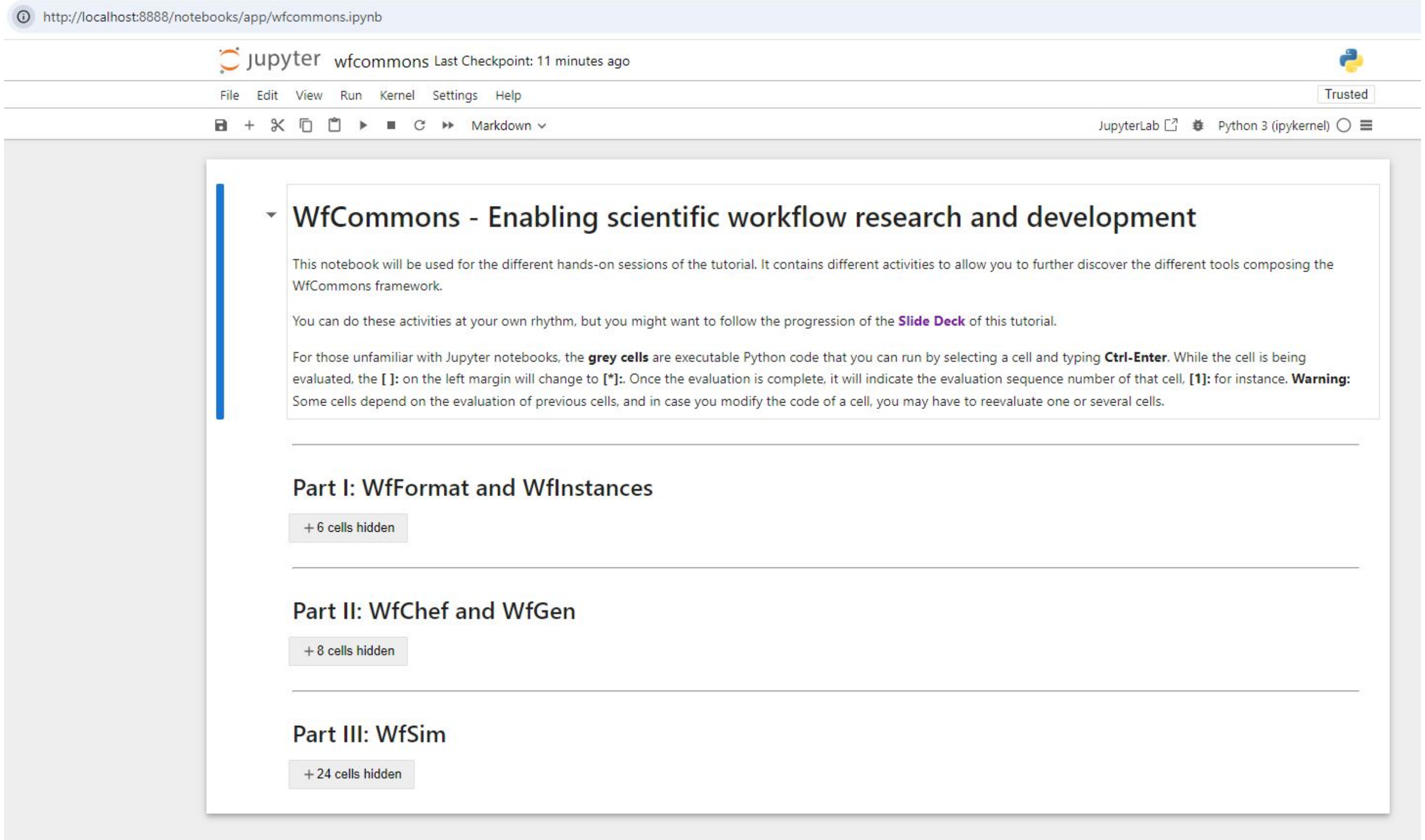
Hands on

Docker: docker run -p 8888:8888 wfcommons/wfcommons-tutorial



```
~/ORNL/escience_tutorial_2024$ docker run -p 8888:8888 wfcommons/wfcommons-tutorial
Unable to find image 'wfcommons/wfcommons-tutorial:latest' locally
latest: Pulling from wfcommons/wfcommons-tutorial
a2318d6c47ec: Pull complete
e0175c87c1f4: Pull complete
bea1d778e3e0: Pull complete
7ce703381f53: Pull complete
462a37cd540d: Pull complete
b46767a865cb: Pull complete
e1a0305284ec: Pull complete
0a1f9b1d5a36: Pull complete
d961eb2475d8: Pull complete
120c31fb2b72: Pull complete
f5542c736e67: Pull complete
1ab452c49d66: Pull complete
17f3bb4cd60a: Pull complete
f9beec37815e: Pull complete
8fdddcdccb73: Pull complete
3870e9becc94: Pull complete
e65f3a981bfb: Pull complete
f323107b1735: Pull complete
9ad7a09bc803: Pull complete
200f2b368560: Pull complete
8cbd19e8ba3d: Pull complete
fd57e4514a59: Pull complete
4f4fb700ef54: Pull complete
47985bd67f3a: Pull complete
99748f599022: Pull complete
8658d8921725: Pull complete
0f65402bdc61: Pull complete
Digest: sha256:116ffadaa4f3a2141fe3ba77225e089b84c11f698a524d42afb57c9ee2fd5f17
Status: Downloaded newer image for wfcommons/wfcommons-tutorial:latest

**************************************************************************
Open a web browser to: http://localhost:8888/notebooks/app/wfcommons.ipynb
**************************************************************************
```

Jupyter Notebook



http://localhost:8888/notebooks/app/wfcommons.ipynb
jupyter wfcommons Last Checkpoint: 11 minutes ago
File Edit View Run Kernel Settings Help
Trusted
Markdown
JupyterLab
Python 3 (ipykernel)

WfCommons - Enabling scientific workflow research and development

This notebook will be used for the different hands-on sessions of the tutorial. It contains different activities to allow you to further discover the different tools composing the WfCommons framework.

You can do these activities at your own rhythm, but you might want to follow the progression of the **Slide Deck** of this tutorial.

For those unfamiliar with Jupyter notebooks, the **grey cells** are executable Python code that you can run by selecting a cell and typing **Ctrl-Enter**. While the cell is being evaluated, the **[ ]:** on the left margin will change to **[*]:**. Once the evaluation is complete, it will indicate the evaluation sequence number of that cell, **[1]:** for instance. **Warning:** Some cells depend on the evaluation of previous cells, and in case you modify the code of a cell, you may have to reevaluate one or several cells.

Part I: WfFormat and WfInstances

+ 6 cells hidden

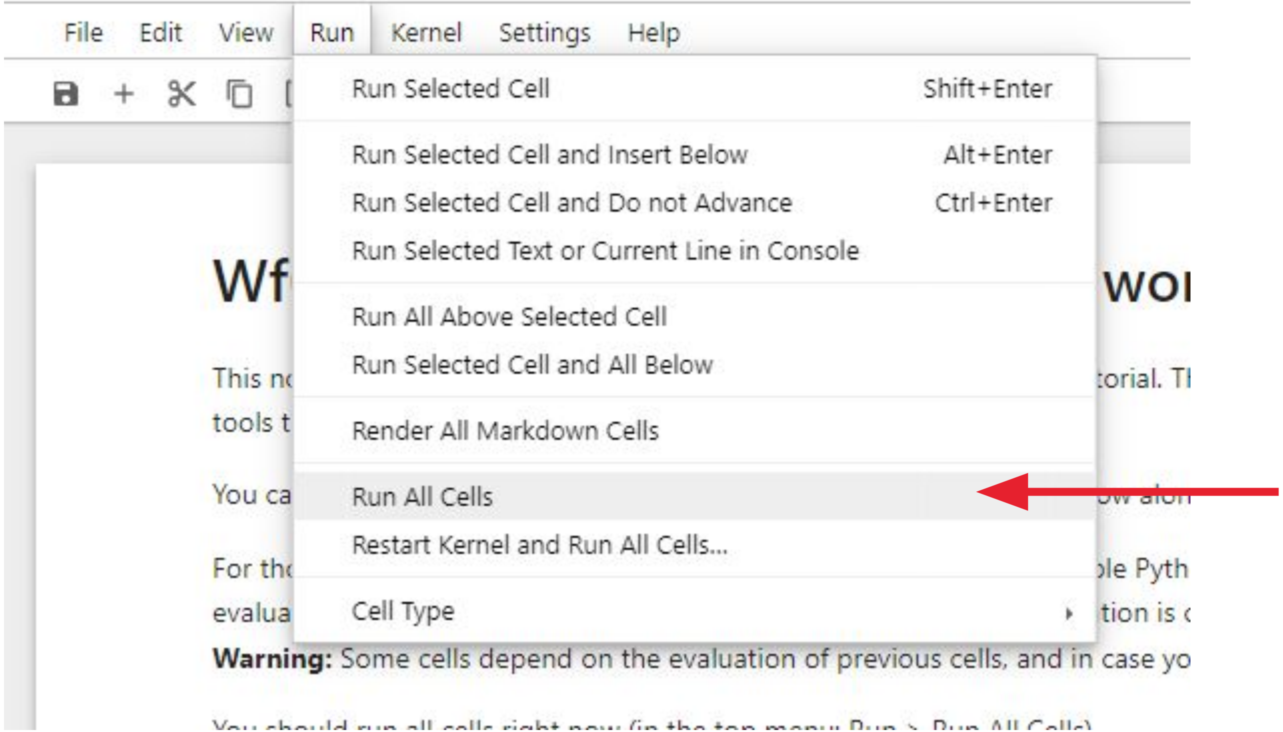
Part II: WfChef and WfGen

+ 8 cells hidden

Part III: WfSim

+ 24 cells hidden

Run all cells now


File
Edit
View
Run
Kernel
Settings
Help
Run Selected Cell
Shift+Enter
Run Selected Cell and Insert Below
Alt+Enter
Run Selected Cell and Do not Advance
Ctrl+Enter
Run Selected Text or Current Line in Console
Run All Above Selected Cell
Run Selected Cell and All Below
Render All Markdown Cells
Run All Cells
Restart Kernel and Run All Cells...
Cell Type

Wf wo

This n torial. Th

tools t

You ca ow alon

For th ole Pyth

evalua tion is

Warning: Some cells depend on the evaluation of previous cells, and in case yo

You should run all cells right now (in the top menu: Run > Run All Cells)

WfChef
and
WfGen

Synthetic and realistic workflow instances

WfChef

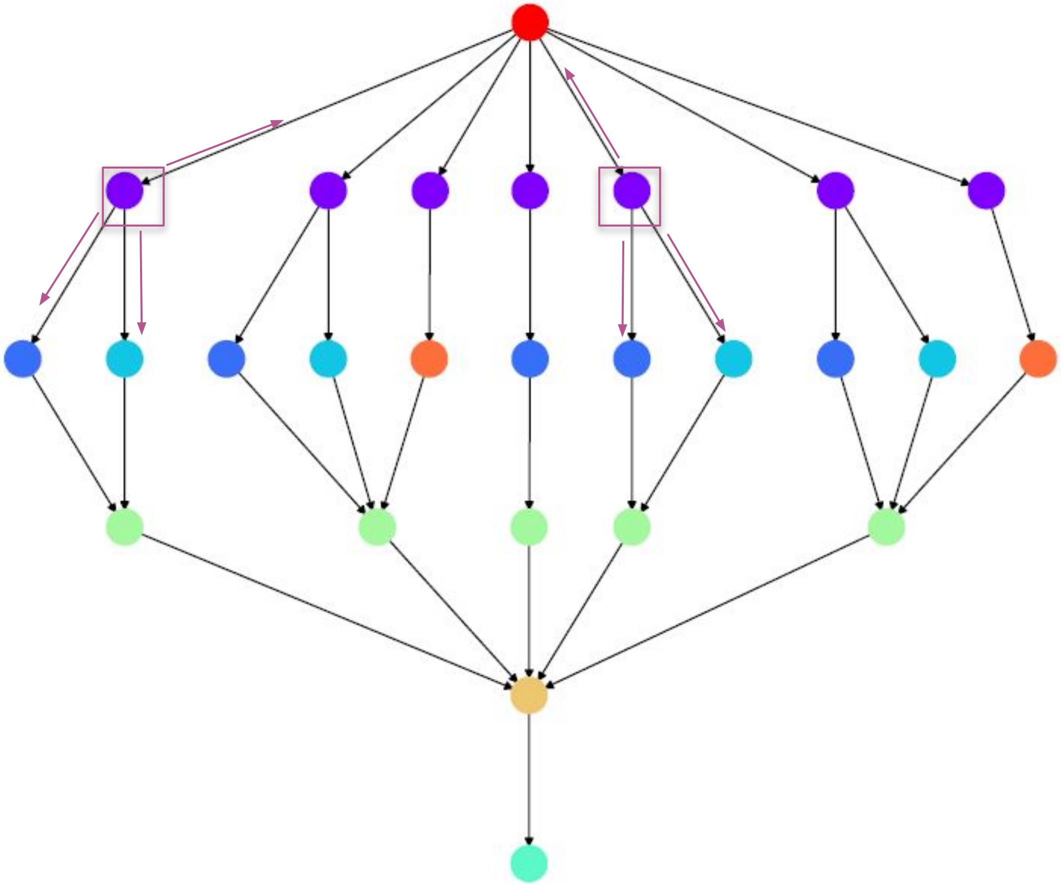
Wfchef: Automated generation of accurate scientific workflow generators.
Coleman, T., Casanova, H., & da Silva, R. F.
2021 IEEE 17th International Conference on eScience

- Automates the construction of synthetic workflow generators
- Inputs
  - Set of real world workflow instances
  - Desired instance size (number of tasks)
- Analyzes the instances
- Finds common patterns
- Duplicates patterns to produce new graphs with desired size

```python
from wfcommons.wfchef.recipes import SeismologyRecipe

# creating a Seismology workflow recipe with increased/decreased runtime and file sizes
  recipe = SeismologyRecipe.from_num_tasks(num_tasks=100,
                                           runtime_factor=1.1,
                                           input_file_size_factor=1.5,
                                           output_file_size_factor=0.8)
```

WfChef

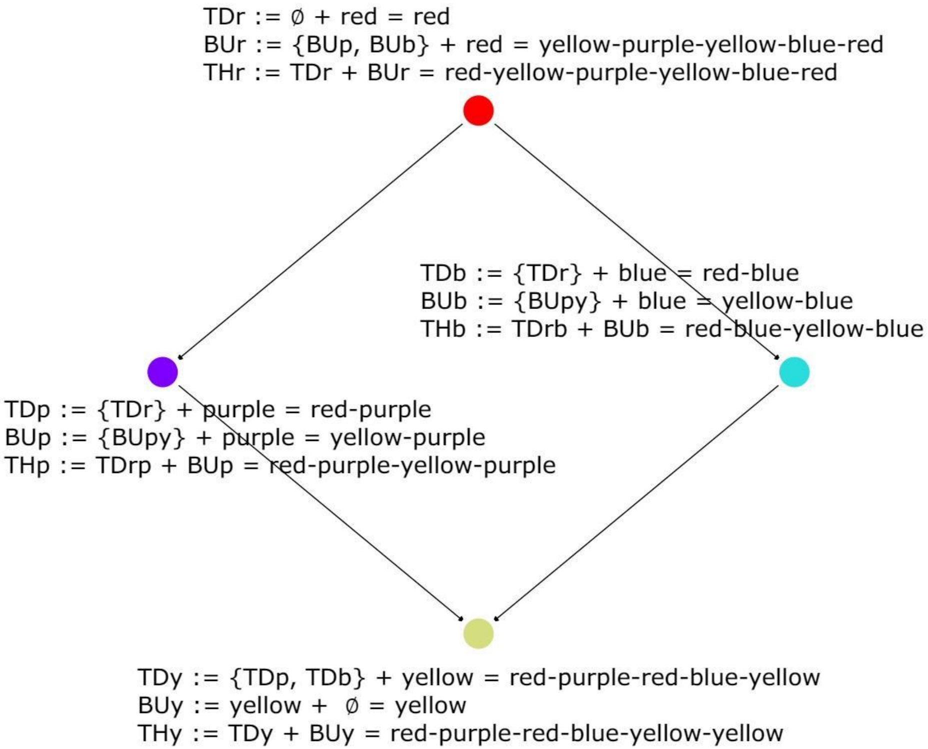
Task Types



WfChef

Type Hash

- ○ Top Down (TD)
- ○ Bottom UP (BU)
- ○ Taks Type
- ○ Type Hash = TD + BU



TDr := ∅ + red = red
BUr := {BUp, BUb} + red = yellow-purple-yellow-blue-red
THr := TDr + BUr = red-yellow-purple-yellow-blue-red

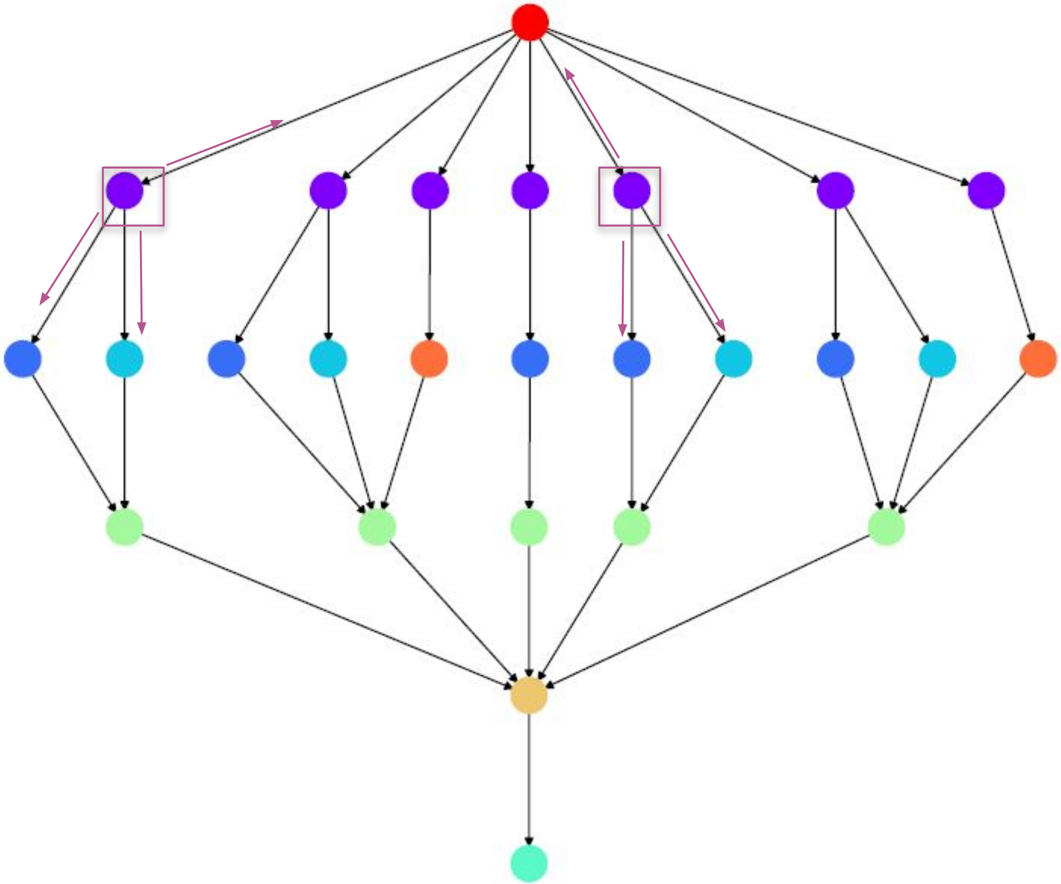
TDb := {TDr} + blue = red-blue
BUb := {BUpy} + blue = yellow-blue
THb := TDrb + BUb = red-blue-yellow-blue

TDp := {TDr} + purple = red-purple
BUp := {BUpy} + purple = yellow-purple
THp := TDrp + BUp = red-purple-yellow-purple

TDy := {TDp, TDb} + yellow = red-purple-red-blue-yellow
BUy := yellow + ∅ = yellow
THy := TDy + BUy = red-purple-red-blue-yellow-yellow

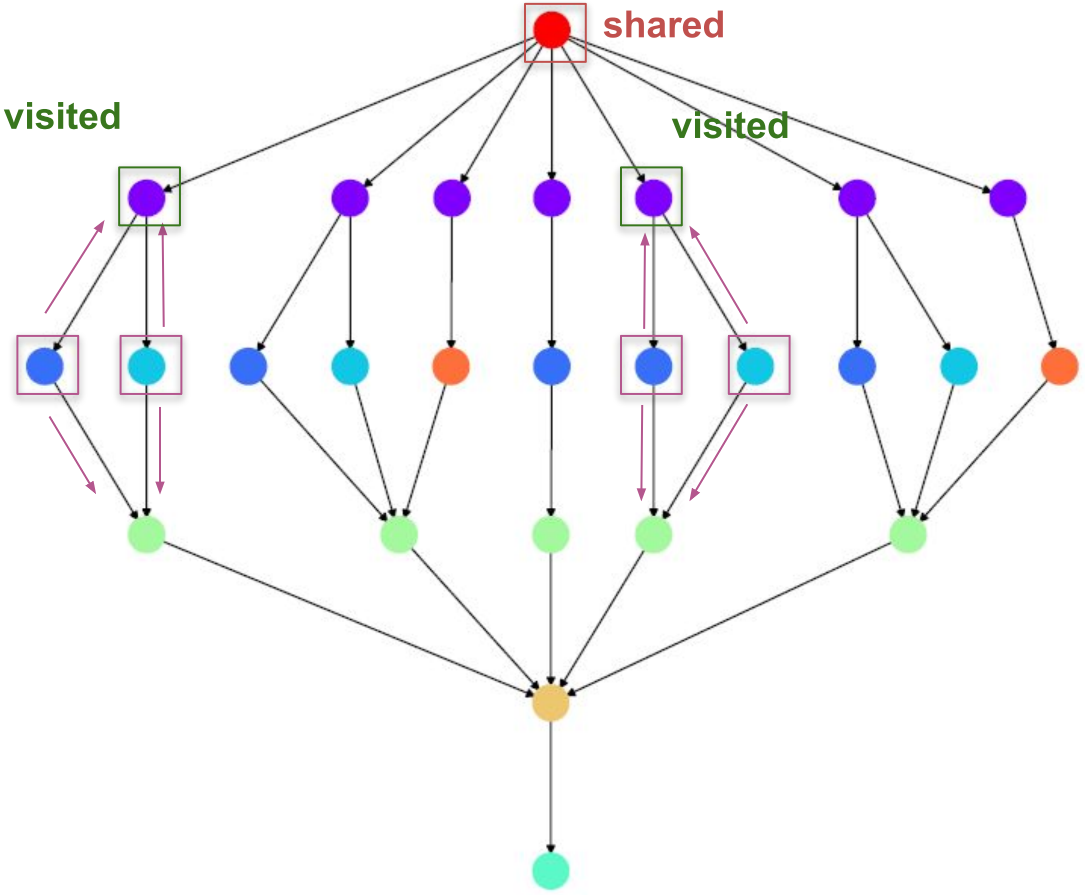
WfChef

Find Pattern Round 1



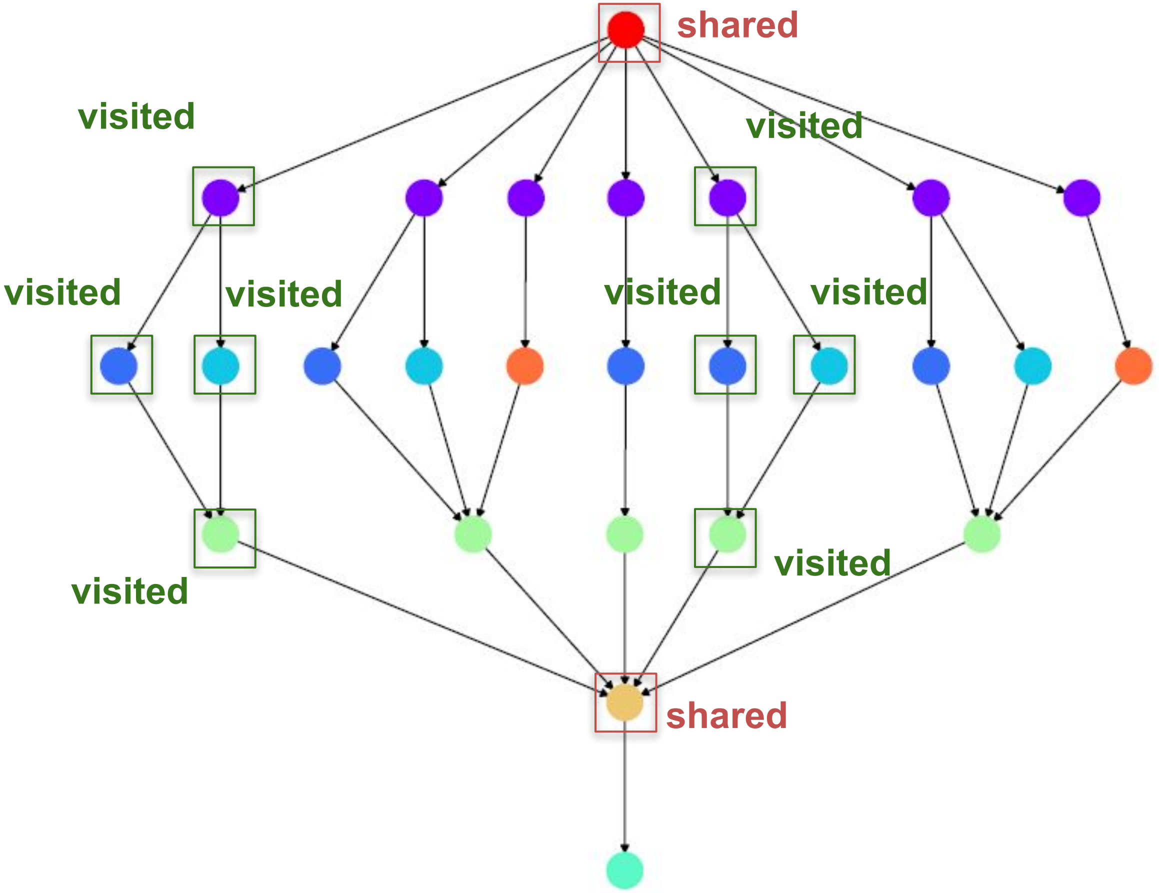
WfChef

Find Pattern Round 2



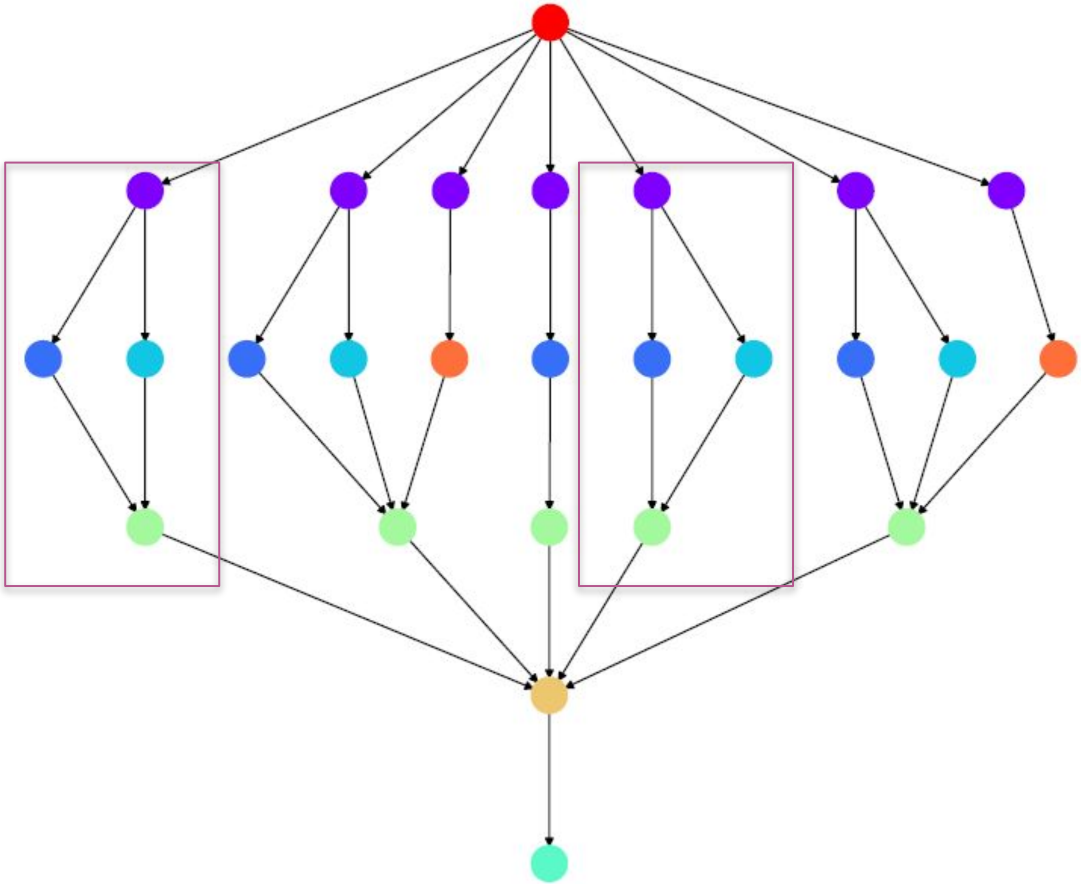
WfChef

Find Pattern Round 3



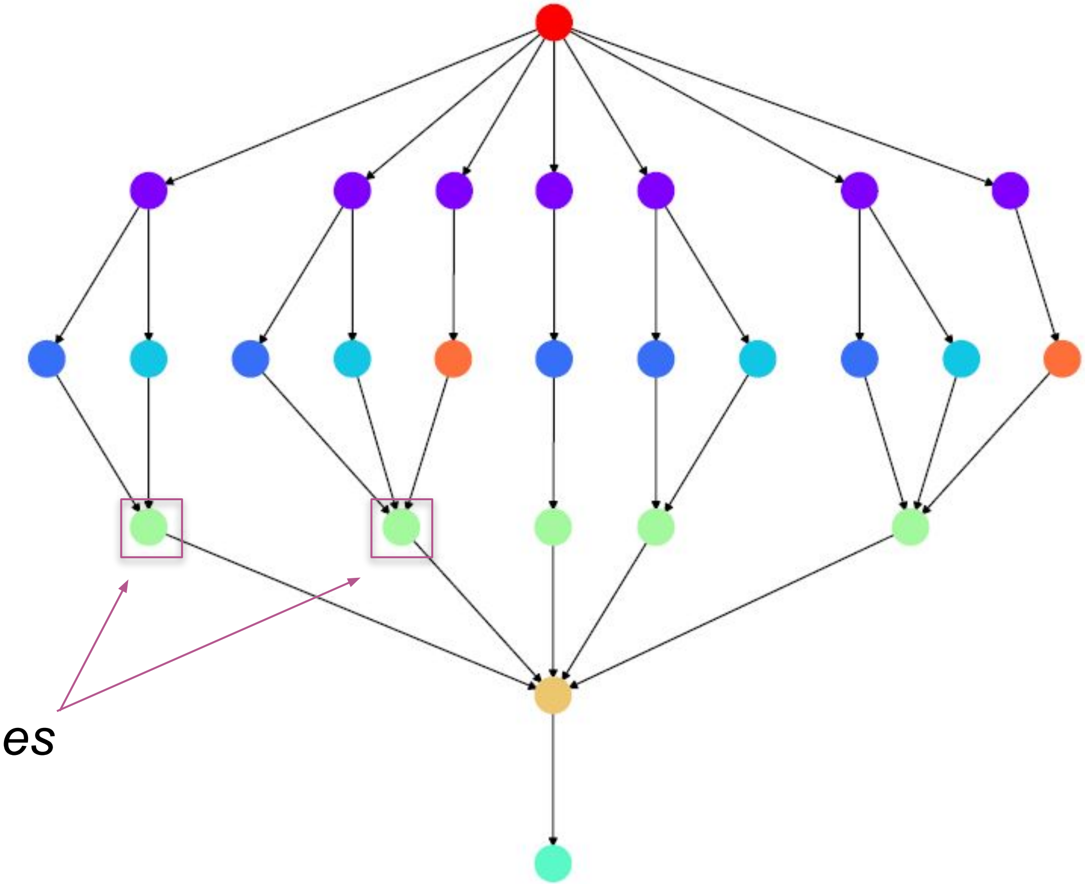
WfChef

Find Patterns



WfChef

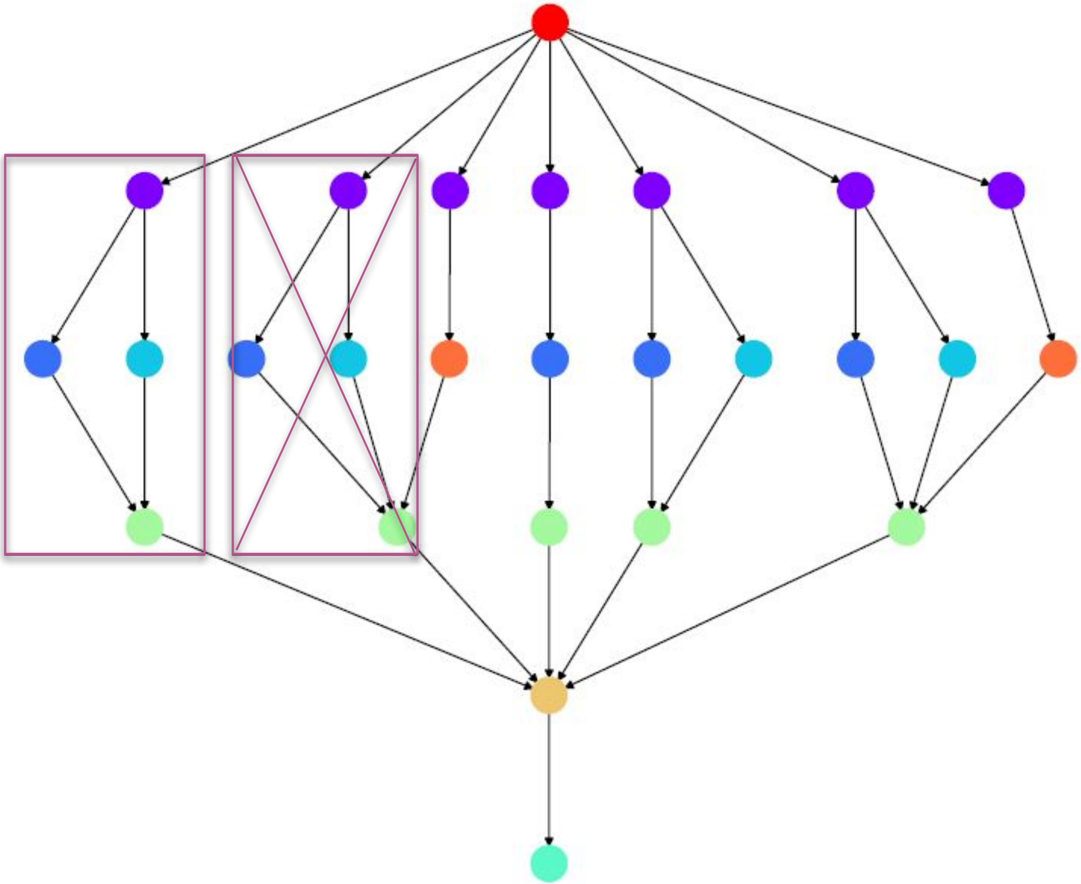
Find Patterns



Different Type Hashes

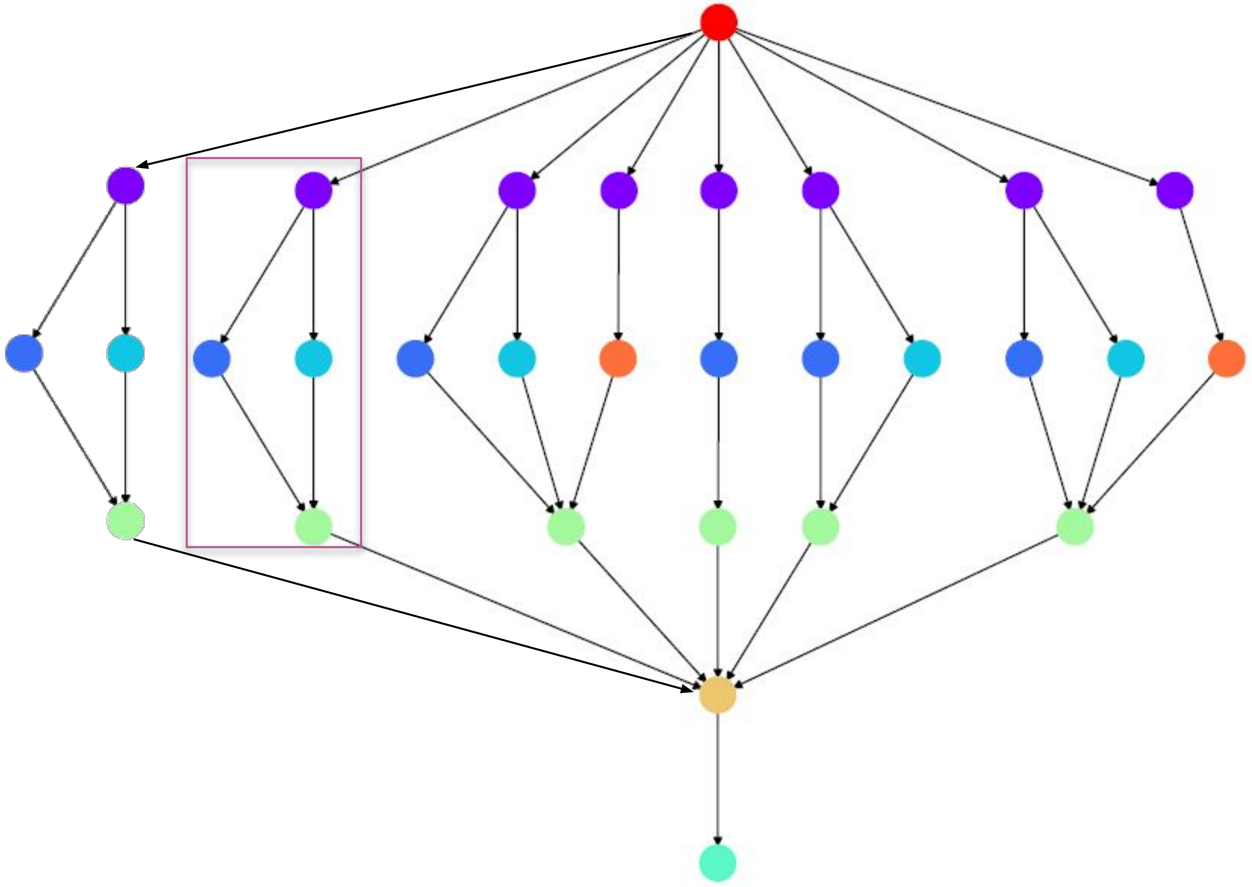
WfChef

Find Patterns



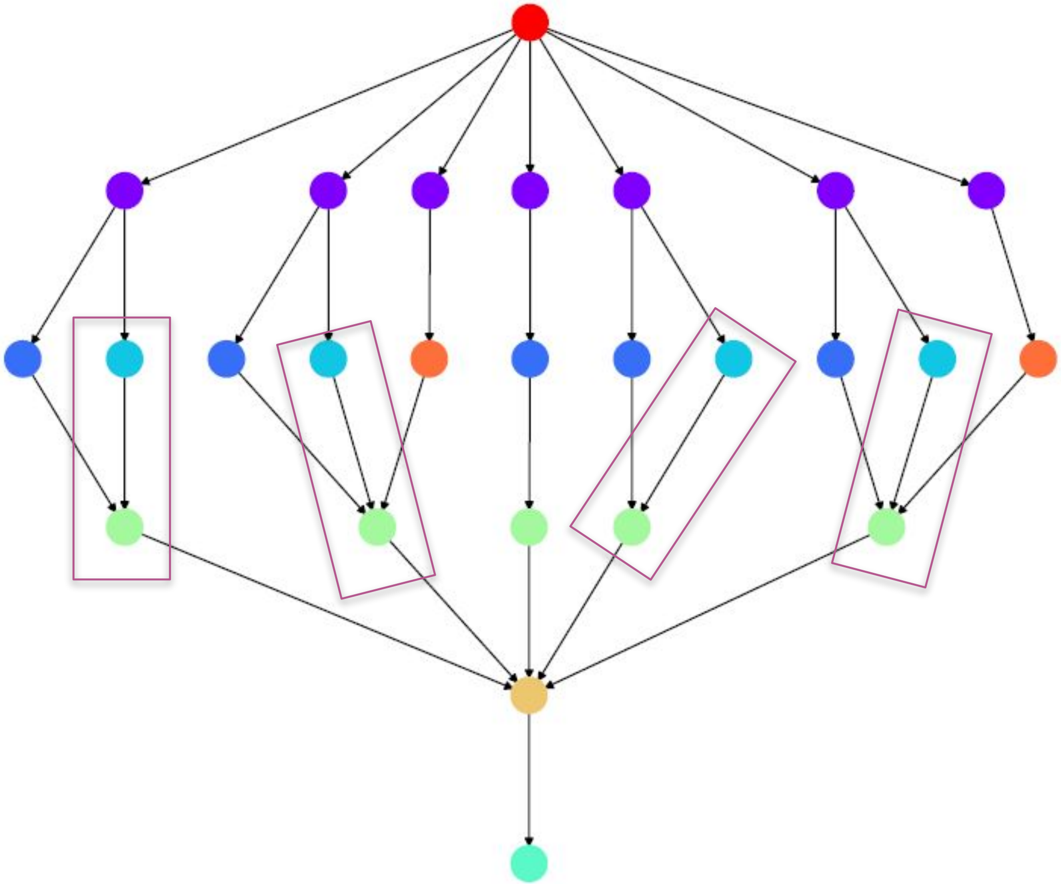
WfChef

Replicate & Generate



WfChef

- ➢ Not a pattern?
- ➢ Cannot be duplicated by itself



WfChef and WfGen

Hands on

WfSim

Simulation of workflows and cyberinfrastructures

Why Simulation?

- Real-world experiments in the field of parallel and distributed computing are not easy to conduct
    - **Labor-intensive:** need fully deployed software/hardware stacks
    - **Time- and energy-intensive:** experiments are often long-running
    - **Limited in scope:** can only experiments with software/hardware stacks available to the researcher
    - **Non-perfectly reproducible**: platform noise, background loads, updates/upgrades
    - **Non-perfect observable:** Full logging is expensive and typically not available

- Simulation (implementation of a software artifact that mimics the real world) can alleviate all these difficulties!

What simulator to use?

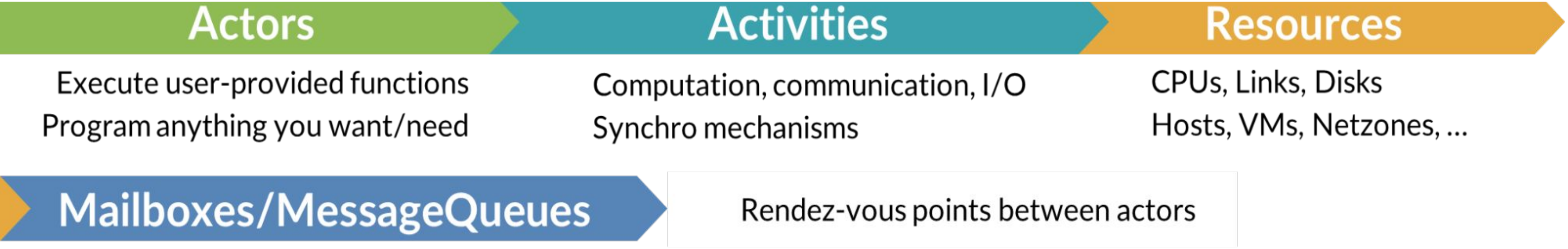
- Any simulator able to parse WfFormat would do the job, **but …**

- Performance evaluation results should enable transfer of to production
- Need to reflect behavior of all the components of actual cyberinfrastructures
  - Compute, network, storage, software stack, …
- Better to build on existing frameworks
  - Validated simulation models, well defined API, reusability, maintenance, …

- Additional desired feature: keep development of simulator simple!

SimGrid *A scientific instrument on your laptop*

- ➢ Open Project since 1998
  - ○ 2,300+ citations and 640+ usages
  - ○ Version 3.36 https://simgrid.org
- ➢ Key strengths
  - ○ **Usability:** Fast, Reliable, User-oriented APIs
  - ○ **Validated performance models:** Open Science à Predictive Power
  - ○ **Versatility:** Grid, P2P, HPC, Cloud, Fog, …
- ➢ SimGrid's fundamental concepts (the **S4U** API)



Actors

Execute user-provided functions
Program anything you want/need

Activities

Computation, communication, I/O
Synchro mechanisms

Resources

CPUs, Links, Disks
Hosts, VMs, Netzones, ...

Mailboxes/MessageQueues

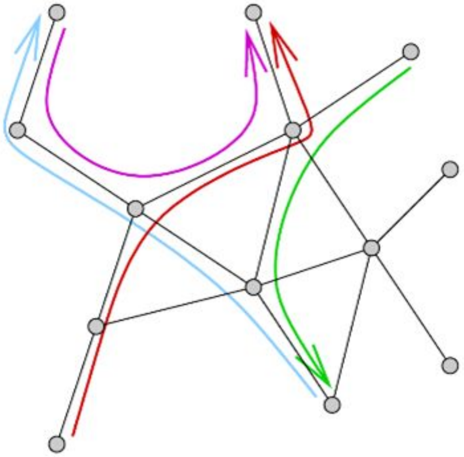
Rendez-vous points between actors

SimGrid Models in a Nutshell

- ➢ Discrete Event Simulator (sequential, but fast)

- ➢ Simulation kernel main loop
  a. Some **activities** are created (by **actors**) and assigned to **resources**
  b. Compute **share** of everyone (resource sharing algorithms)
  c. Compute the **earliest finishing** activity, **advance** simulated time
  d. Remove finished activity
  e. Loop back to b

- ➢ **Flow-level** models
  - ○ Boils down to solve a **linear max min** problem
  - ○ Applied to computing, network, and I/Os
  - ○ Good tradeoff between **speed and accuracy**
  - ○ Multiple optimization techniques and specializations

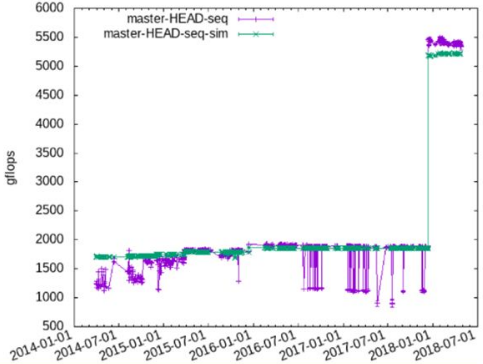


Some SimGrid highlights

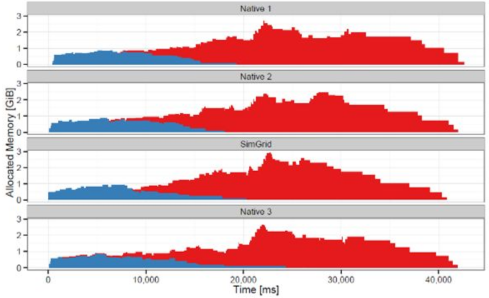
HPL at scale

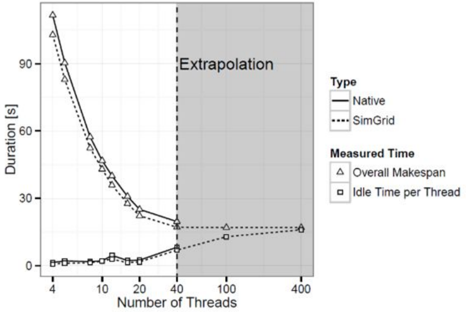
- Qualification run
  - Matrix rank: 3,875,000
  - 6,006 MPI ranks (stampede)
  - Duration: 2 hours
- Simulated on **1 core**
  - In 47 hours
  - With 16GB of memory

Performance regression detector

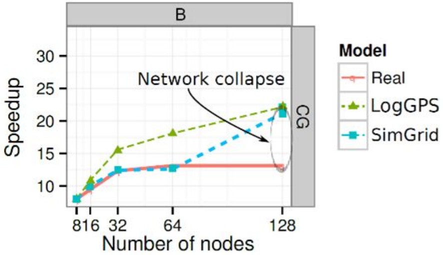


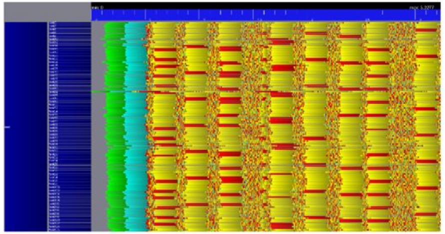
Memory peak estimation & performance extrapolation for Sparse QR factorization



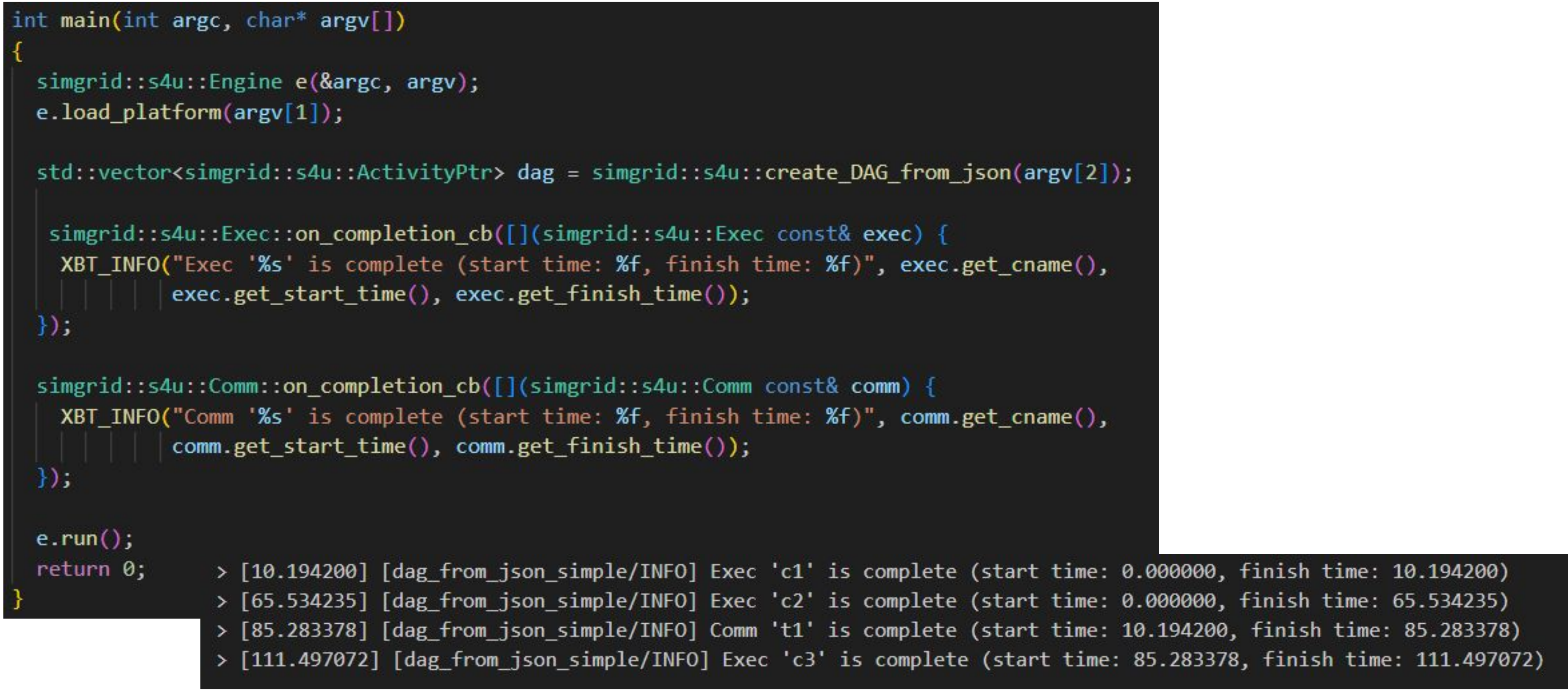


Blame the system not the model





Workflow Simulation with SimGrid

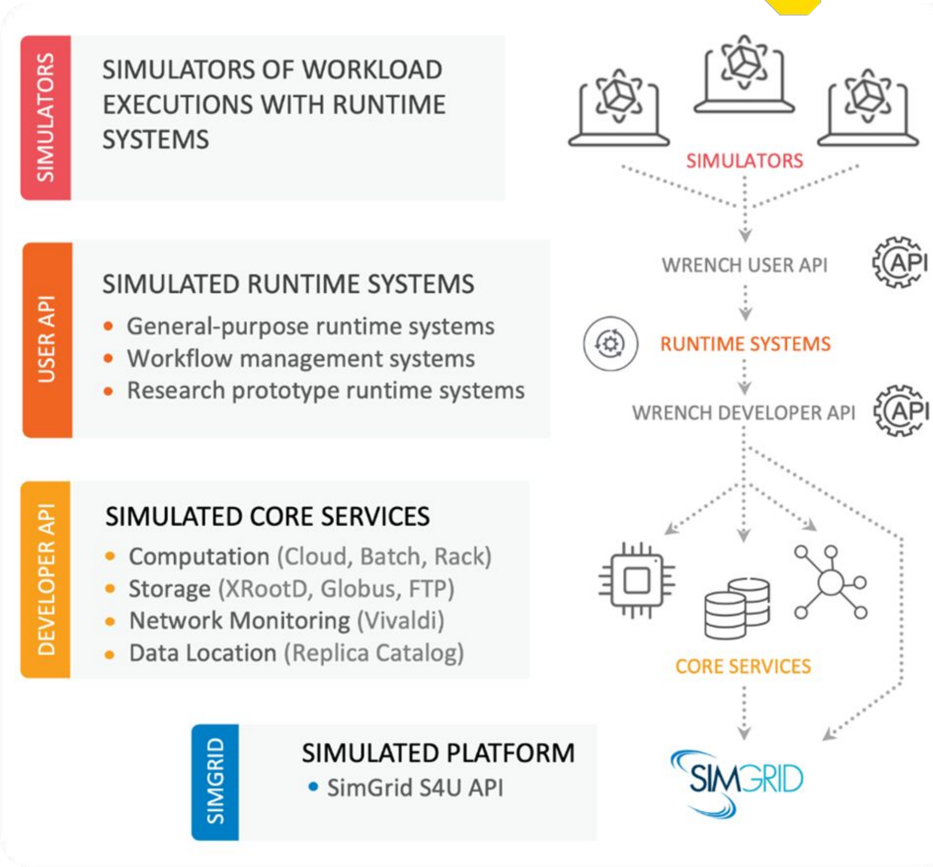


```cpp
int main(int argc, char* argv[])
{
  simgrid::s4u::Engine e(&argc, argv);
  e.load_platform(argv[1]);

  std::vector<simgrid::s4u::ActivityPtr> dag = simgrid::s4u::create_DAG_from_json(argv[2]);

   simgrid::s4u::Exec::on_completion_cb([](simgrid::s4u::Exec const& exec) {
    XBT_INFO("Exec '%s' is complete (start time: %f, finish time: %f)", exec.get_cname(),
             exec.get_start_time(), exec.get_finish_time());
  });

  simgrid::s4u::Comm::on_completion_cb([](simgrid::s4u::Comm const& comm) {
    XBT_INFO("Comm '%s' is complete (start time: %f, finish time: %f)", comm.get_cname(),
             comm.get_start_time(), comm.get_finish_time());
  });

  e.run();
  return 0;
}
```

```
> [10.194200] [dag_from_json_simple/INFO] Exec 'c1' is complete (start time: 0.000000, finish time: 10.194200)
> [65.534235] [dag_from_json_simple/INFO] Exec 'c2' is complete (start time: 0.000000, finish time: 65.534235)
> [85.283378] [dag_from_json_simple/INFO] Comm 't1' is complete (start time: 10.194200, finish time: 85.283378)
> [111.497072] [dag_from_json_simple/INFO] Exec 'c3' is complete (start time: 85.283378, finish time: 111.497072)
```

WRENCH

- ➢ Project initiated in 2016
- ➢ **Objectives**
  - ○ A virtual lab to study WMS
  - ○ Improve SimGrid expressiveness
- ➢ **DSL**-like approach:
  - ○ High level concepts
  - ○ Composable modules
  - ○ Different levels of APIs
- ➢ Version 2.3
- ➢ https://wrench-project.org



SIMULATORS

SIMULATORS OF WORKLOAD EXECUTIONS WITH RUNTIME SYSTEMS

USER API

SIMULATED RUNTIME SYSTEMS

- General-purpose runtime systems
- Workflow management systems
- Research prototype runtime systems

DEVELOPER API

SIMULATED CORE SERVICES

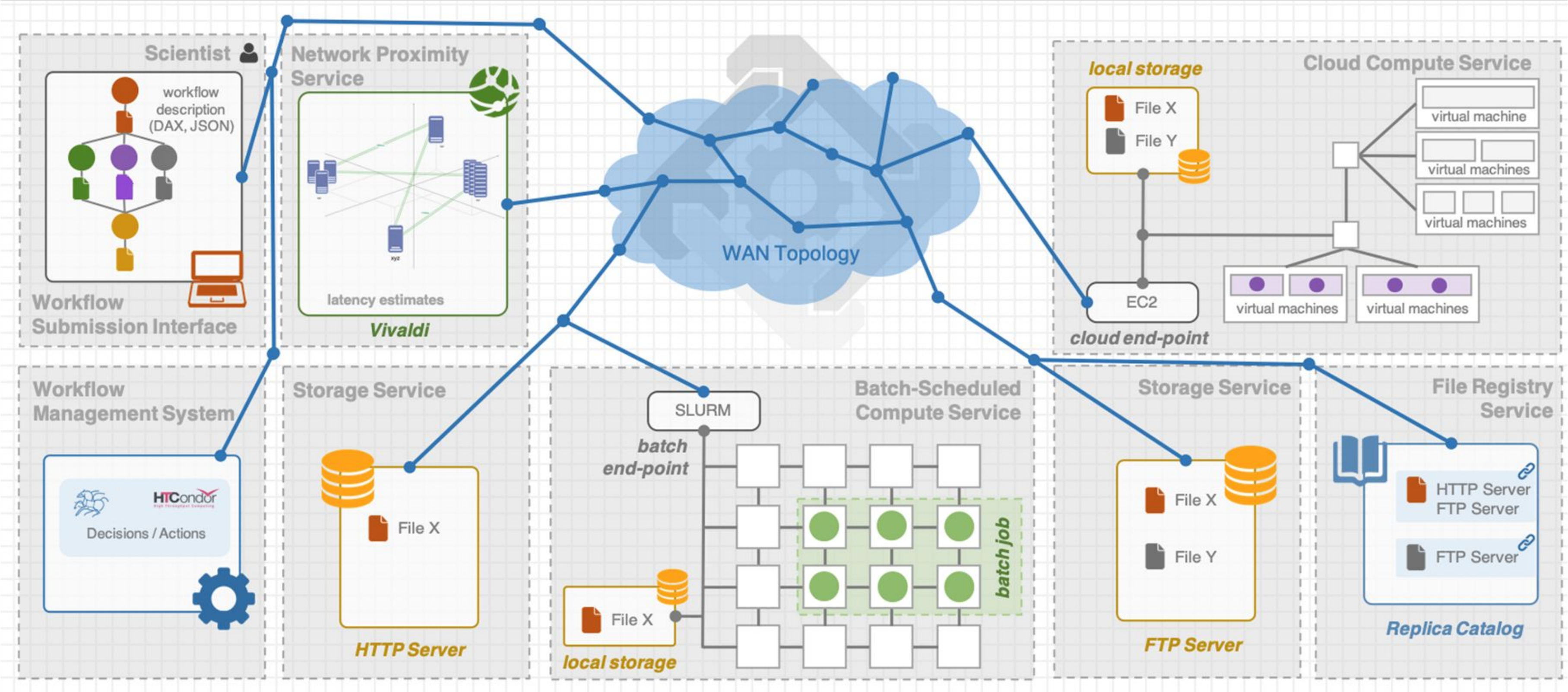
- Computation (Cloud, Batch, Rack)
- Storage (XRootD, Globus, FTP)
- Network Monitoring (Vivaldi)
- Data Location (Replica Catalog)

SIMGRID

SIMULATED PLATFORM

- SimGrid S4U API

Simulation with WRENCH



WfSim

Hands on

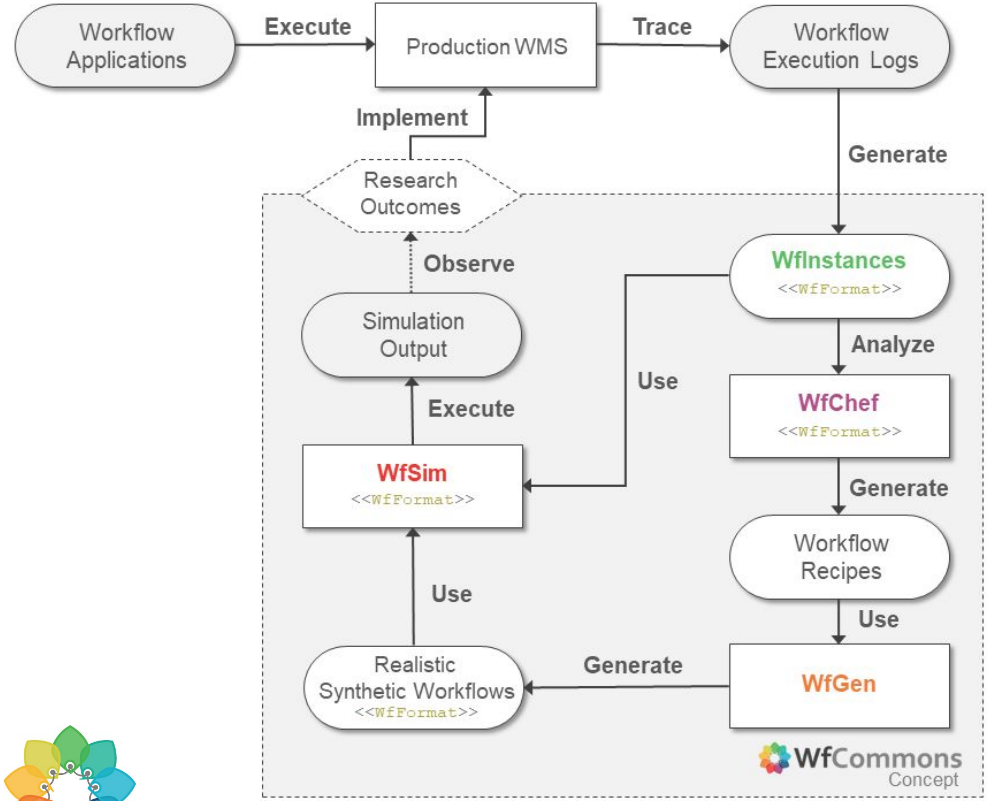
Concluding Remarks

WfCommons is a framework that provides a collection of tools for analyzing **real workflow execution traces**, producing realistic **synthetic workflow execution traces**, and **benchmarking** / **simulating** workflow executions.

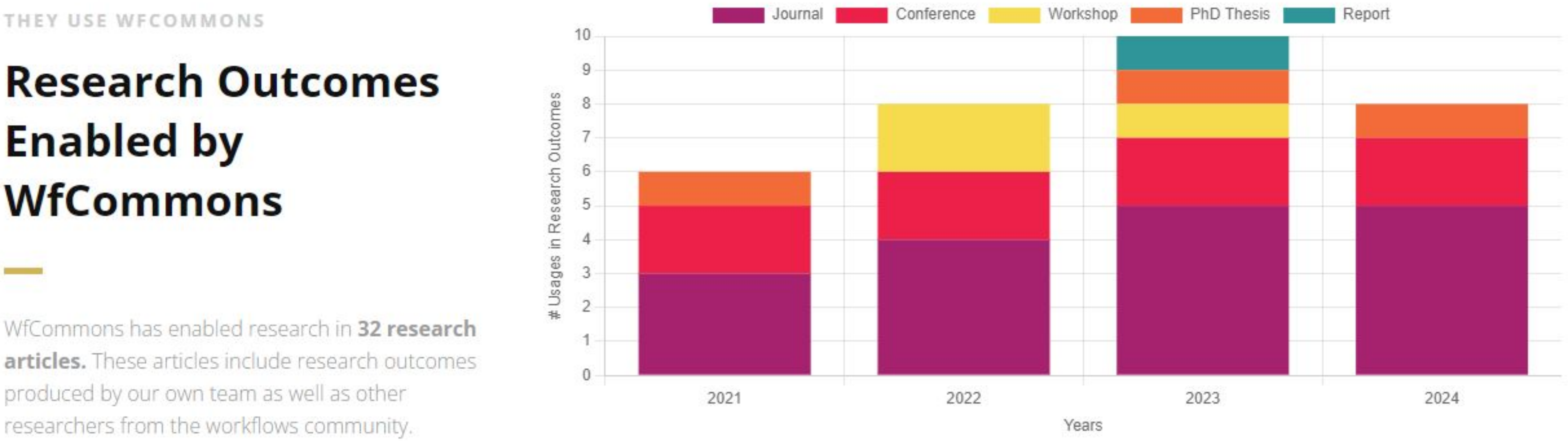
WfCommons: A framework for enabling scientific workflow research and development

Coleman, T., Casanova, H., Pottier, L., Kaushik, M., Deelman, E., & da Silva, R. F. Future generation computer systems 128 (2022).

https://wfcommons.org



WfCommons is Used in Research Studies



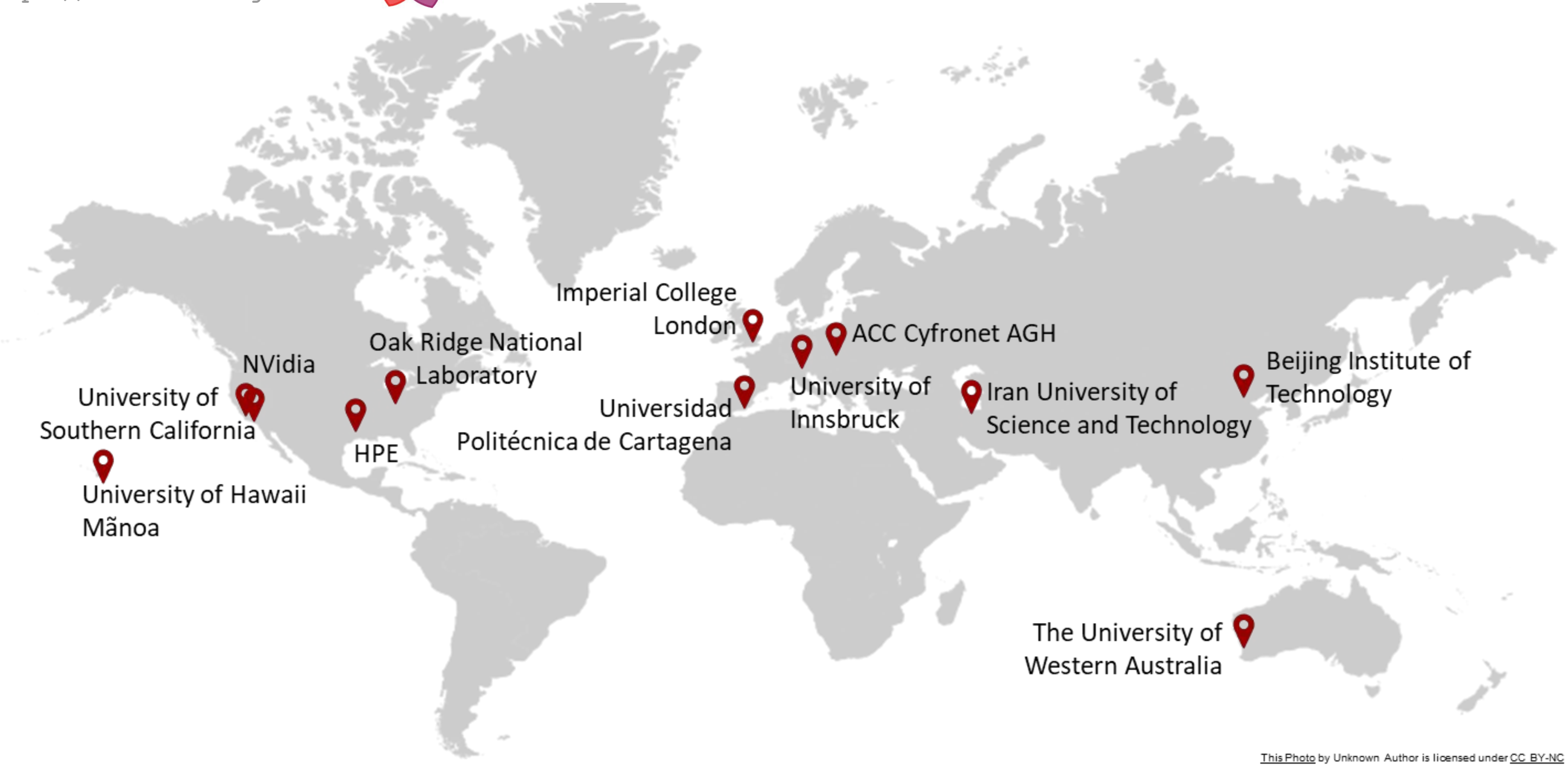
THEY USE WFCOMMONS

Research Outcomes Enabled by WfCommons

WfCommons has enabled research in **32 research articles.** These articles include research outcomes produced by our own team as well as other researchers from the workflows community.

WfCommons

https://wfcommons.org


Imperial College London
ACC Cyfronet AGH
Oak Ridge National Laboratory
NVidia
Beijing Institute of Technology
University of Southern California
Universidad Politécnica de Cartagena
University of Innsbruck
Iran University of Science and Technology
HPE
University of Hawaii Mãnoa
The University of Western Australia

Useful Links

- Website
  - https://wfcommons.org/
  - https://docs.wfcommons.org/en/latest/
- WfInstances browser
  - https://wfinstances.ics.hawaii.edu/
- Github repositories
  - https://github.com/wfcommons
  - https://github.com/wfcommons/WfFormat
  - https://github.com/wfcommons/wfinstances
- Simulation tools
  - https://simgrid.org
  - https://wrench-project.org
  - https://eduwrench.ics.hawaii.edu/

Contributing Back to WfCommons

https://wfcommons.org

- ➢ Adopt **WfFormat** when tracing workflow executions
- ➢ Add new instances to the **WfInstances** repositories

- ➢ Let us know of:
  - ○ New **recipes**
  - ○ New **simulators**
  - ○ New **simulation toolkits** supporting **WfFormat**
  - ○ New **publications** using **WfCommons**

Thank you for your participation on this tutorial!