A Framework for Scientific Workflow Reproducibility in the Cloud

Rawaa Qasha, Jacek Cała, Paul Watson
Newcastle University, Newcastle upon Tyne, UK
Email: {r.qasha, jacek.cala, paul.watson}@newcastle.ac.uk
In this paper

• A new framework for repeatability and reproducibility of scientific workflow

• Integrating logical and physical preservation approaches

• Offering Workflow/tasks repositories with version control

• Supporting automatic deployment and image capture of workflows and tasks
Outline

• Background
• Challenges for workflow reproducibility
• Our solution for logical and physical preservations
• Overview of reproducibility framework
• Experiments and results
• Conclusions
Workflows & Reproducibility

- **total no. of workflows**
- **Workflows can be re-executed**

---

**Number of workflows**

<table>
<thead>
<tr>
<th>Study</th>
<th>Total Workflows</th>
<th>Re-executable Workflows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Study 1</td>
<td>92</td>
<td>18 (~20%)</td>
</tr>
<tr>
<td>Study 2</td>
<td>1443</td>
<td>341 (~24%)</td>
</tr>
</tbody>
</table>

---

Challenges for workflow reproducibility

- Insufficiently detailed workflow description
- Insufficient description of the execution environment
- Unavailable execution environments
- Absence of & changes in the external dependencies
- Missing input data
Common reproducibility approaches

Logical preservation

Physical preservation
Using TOSCA as a logical preservation

Workflow and execution environment description
Using Docker for physical preservation

(a) Initial task deployment & execution

(b) Task deployment & execution with task image

Preserving execution environment and dependencies, tracking changes
Workflow Deployment & Enactment Engine
(TOSCA Runtime Environment: Cloudify)

Target Execution Environment
(Docker over local VM, AWS, Azure, GCE, …)
Multi-container deployment
Single container deployment
Time line of workflow devOps

Developer

- Develop WF tasks, Node/Relationship Types
- Implementing Lifecycle Scripts
- Building WF Blueprints

Researcher/User

- Deploying WF and automatic producing Docker images
- Creating WF repository

WF Available

- Clone specific WF repository
- Reproduce the cloned WF
Preserving description, input data, tracking changes and deployment instructions
Experiments and Results
1 - Repeatability of a workflow on different clouds
### 2- Automatic image capture for improved performance

<table>
<thead>
<tr>
<th></th>
<th>NJ</th>
<th>Picard</th>
<th>Col-Invert</th>
<th>FileZip</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Deployment Time [Mins]</strong></td>
<td><strong>Base Image</strong></td>
<td><strong>Base+dependencies</strong></td>
<td><strong>Tasks Images</strong></td>
<td><strong>Base Image</strong></td>
</tr>
</tbody>
</table>
3- Reproducibility in the face of development changes
Conclusions

• Full workflow reproducibility is a long-standing issue

• TOSCA description is used for logical preservation

• Docker images for tasks/workflows support physical preservation

• Changes tracking and automatic deployment also contribute to a comprehensive solution of the problem

• Integration of these techniques addresses majority of the issues related to workflow decay
THANK YOU