Report

Development of a portfolio of tools for the creation and documentation of reproducible simulation workflows – Tools for creating reproducible scientific workflows

1. General information

Applicant	Dr. Jörg F. Unger
Affiliation	Bundesanstalt für Materialforschung und -prüfung (BAM) Abteilung 7.7 Modellierung und Simulation
Project title	Development of a portfolio of tools for the creation and documentation of reproducible simulation workflows
Funding period	01.07.2021 - 31.12.2022

1.1 List of publications

Diercks, P., Gläser, D., Lünsdorf, O., Selzer, M., Flemisch, B., & Unger, J. F. (2022). Evaluation of tools for describing, reproducing and reusing scientific workflows. *ing.grid*. Retrieved from https://preprints.inggrid.org/repository/view/5/

2. Executive Summary

In the field of computational science and engineering, workflows often entail the application of various software, for instance, for simulation or pre- and postprocessing. Typically, these components are combined in arbitrarily complex workflows to address a specific research question. For peer researchers to understand, reproduce and (re)use the findings of a scientific publication, several challenges must be addressed. For instance, the employed workflow has to be automated and information on all used software must be available for a reproduction of the results. Moreover, the results must be traceable, and the workflow documented and readable to allow for external verification and greater trust. The goal of the project is the development of a NFDI4Ing-portfolio of tools for the creation and documentation of reproducible simulation workflows, to support researchers in overcoming the challenges above.

In this project, existing workflow management systems (WfMSs) are discussed regarding their suitability for describing, reproducing, and reusing scientific workflows. To this end, a set of general requirements for WfMSs were deduced from user stories that we deem relevant in the domain of computational science and engineering. Based on an exemplary workflow implementation and available documentation of each individual tool, a selection of different WfMSs is compared with respect to these requirements, to support fellow scientists in identifying the WfMSs that best suit their requirements.

3. Project report

3.1 Introduction

With increasing volume, complexity and creation speed of scholarly data, humans rely more and more on computational support in processing this data. The "FAIR guiding principles for scientific data management and stewardship" (Wilkinson, 2016) were introduced to improve the ability of machines to automatically find and use that data. FAIR comprises the four foundational principles "that all research objects should be Findable, Accessible, Interoperable and Reusable (FAIR) both for machines and for people". Data processing is usually not a single task, but in general relies on a chain of tools. Thus, to achieve transparency, adaptability, and reproducibility of (computational) research, the FAIR principles must be applied to all components of the research process. This includes the tools (i. e. any research software) used to analyze the data, but also the scientific workflow itself which describes how the various processes depend on each other.

In addition, in recent years there has been a tremendous development of different tools (see e. g. awesome-pipeline) that aid the definition and automation of computational workflows. These WfMSs have great potential in contributing to the transparency, adaptability, and reproducibility of computational research. Therefore, the main goal of the project is the development of a NFDI4Ing-portfolio of tools for the creation and documentation of reproducible simulation workflows.

The following measures are defined to reach this goal. Based on the authors' experience, user stories that are relevant in the domain of computational science and engineering are defined. These user stories are then used to extract a set of general requirements for WfMSs. Several different tools are compared with respect to these requirements to support fellow scientists in identifying the tools that best suit their requirements. Moreover, a GitHub repository (Diercks, Gläser, Unger, & Flemisch, 2022), providing an implementation of an exemplary workflow for all tools and a short documentation with a link to further information was created. By demonstrating how the different tools could be used, people are encouraged to use WfMSs in their daily work and a basis for getting started is provided.

Initially, in addition to the user stories above, the implementation of two complex use cases were planned. However, as we identified the definition of process interfaces and the portability of the compute environment in the context of high-performance-computing (HPC) as two major challenges, it was decided to address these issues in a single demonstrator workflow with increased complexity instead. First, the definition of process interfaces plays an important role in joint research and workflow development and is challenged by constant changes of the individual components of the workflow. Second, achieving portability of (parts of) the workflow, such that it can be executed seamlessly in an HPC environment poses several issues. Many WfMSs make use of package management systems or container technology to provide the compute environment. However, without access to the internet this is not feasible. Moreover, successfully using container technology as an MPI-distributed application seems to be a technical challenge.

3.2 Tool comparison

In this section, the results of the tool comparison are given, which was the main objective of the project. First, the user stories used to derive the requirements on the workflow tools are described. Second, the capabilities that the WfMSs should ideally provide are discussed and given in the form of general requirements. More details on the results given here can be found in (Diercks, et al., 2022).

3.2.1 User stories

Starting from user stories that we consider representative for computational science and engineering, a set of requirements is derived that serves as a basis for the comparison of different WfMSs. Reproducibility, which is key to transparent research, is the focus of the first user story. The second user story deals with research groups that develop workflows in a joint effort where subgroups or individuals work on different components of the workflow. Finally, the third user story focuses on computational research that involves generating and processing large amounts of data, which poses special demands on how the workflow tools organize the data that is created upon workflow execution.

3.2.2 Requirements

3.2.2.1 Support for job scheduling system

The main task of a WfMS is to automatically execute the processes of a workflow in the correct order such that the dependencies between them are satisfied. This requirement focuses on the ability of a workflow tool to distribute the computations on available resources. Therefore, it is of great benefit if WfMSs support the integration of widely used job scheduling systems such that users can specify resources (number of nodes, CPUs, memory, etc.) of submitted (either locally or to a remote machine) computations. Ideally, the workflow can be executed anywhere without changing the workflow definition itself, but only the runtime arguments or a configuration file.

3.2.2.2 Monitoring

Depending on the application, the execution of scientific workflows can be very time-consuming. It can be very helpful to be able to query the state of the execution, that is, which processes have been finished, which processes are currently being executed, and which are still pending.

3.2.2.3 Graphical user interface

Independent of a particular execution of the workflow, the workflow system may provide facilities to visualize the graph of the workflow, indicating the mutual dependencies of the individual processes and the direction of the flow of data. Beyond a mere visualization, a GUI may allow for visually connecting different workflows into a new one by means of drag & drop

3.2.2.4 Data provenance

The data provenance graph contains, for a particular execution of the workflow, which data and processes participated in the generation of a particular piece of data. Collection of all relevant information, its storage in machine-readable formats and subsequent publication alongside the data can be very useful for future researchers to understand how exactly the data was produced. Ideally, the workflow system has the means to automatically collect this information upon workflow execution

3.2.2.5 Compute environment

The workflows need to be executable by others, to guarantee interoperability and reproducibility of scientific workflows. Here, the re-instantiation of the compute environment (installation of libraries or source code) poses the main challenge. Therefore, it is of great use if the workflow tool can automatically deploy the software stack (on a per workflow or per process basis) by means of a package manager (e. g. conda) or that running processes in a container (e. g. Docker) is integrated in the tool.

3.2.2.6 Hierarchical composition of workflows

A workflow consists of a mapping between a set of inputs and a set of outputs, whereas in between several processes are performed. Each of the processes can also be a workflow itself. Therefore, it is important that processes or entire workflows can be imported/composed within the WfMS. This might also require defining separate compute environments for each sub-workflow or process.

3.2.2.7 Interfaces

In contrast to traditional file-based pipelines, it is often more convenient to pass non-file output (e. g. float or integer values) directly from one process to another without the creation of intermediate files. Here, it is desirable that the workflow tool can check the validity of the data (e.g., the correct data type) to be processed, making it easier to understand how to use, adapt or extend the workflow/process.

3.2.2.8 Up-to-dateness

There are different areas for the application of workflows. On the one hand, people might use a workflow tool to manage computations involving the generation and processing of large amounts of data. If identical runs are detected, a recomputation should be avoided and the original output should be linked in the data provenance graph. Another area of application is the constant development within the workflow. When changing the processes, the workflow system should rather behave like a build system (such as make) - only recomputing the steps that are changed or that depend on these changes.

3.2.2.9 Ease of first use

Although this is not a requirement per-se, it is beneficial if the workflow system has an intuitive syntax/interface and little work is required for a new user to define a first workflow. The complexity added by the WfMS should be as small as possible.

3.2.2.10 Manually editable workflow definition

It is important that the workflow description is given in a human-readable format, to facilitate version-controlling of the workflow and to not force users and/or developers to rely on the GUI.

3.3 Reproducible workflows in the context of HPC environments

To achieve reproducibility of scientific workflows besides automation and scalability, portability plays an important role. A workflow needs to be portable in the sense that all software dependencies can be automatically installed. Existing workflow management systems support the deployment of the software stack by integration of container technology (docker) or platform independent package management systems (conda). However, (based on our current experience) several limitations to the use of such technology on a traditional HPC cluster exist:

- the HPC user is only allowed to build applications to be run from source,
- without access to the internet installing isolated conda environments or downloading container images is not possible,
- use of docker in HPC environments is usually discouraged due to access rights (security concerns),
- successfully using container technology as an MPI-distributed application seems to be a technical challenge.

Regarding the latter point, great care must be taken to build a container that is compatible (e.g., MPI implementation, drivers, ...) with the host system. With no access to the internet, the best option might be to pursue a containerized (multi-stage build) solution. First, one would need to define a base layer, such that the container is compatible with the host system. This may be provided by the system administrators since the base layer is specific to the host system. The user is then able to build his own application (on a local machine) on top of the base layer. As part of the workflow, the container image then needs to be transferred to the HPC system prior to the execution of the application. Another option is conda-pack which is a command line tool for creating relocatable conda environments. The creation and relocation of the conda environment, as well as the transfer of inputs and outputs associated with the process to be executed on the HPC system, can be integrated into the workflow. This helps to achieve portability and reproducibility of HPC processes within scientific workflows but may not be an unconditionally stable solution since the conda environment is not completely isolated from the host.

3.4 Integration in the NFDI4Ing context

The data generated over the course of the project is publicly available and hosted on GitHub. The GitHub repository (Diercks, Gläser, Unger, & Flemisch, 2022) that

contains the WfMS implementations of the exemplary workflow was created with the aim to continuously add more tools in the future, and to extend the documentation accordingly. Furthermore, the GitHub repository (Diercks, Gläser, Unger, & Flemisch, NFDI4Ing HPC Workflows, 2022) was created to document different approaches that address how to achieve portable workflow implementations in the context of HPC computing.

As part of the project, a special interest group was formed within NFDI4Ing to report findings to, and to include feedback from interested members of NFDI4Ing. The presentations (held by P. Diercks, D. Gläser, B. Flemisch and J. F. Unger) for the SIG meetings can be found on the NFDI4Ing share point.

It is noted that Michael Selzer (Task Area *CADEN*) and Ontje Lünsdorf over the course of the project frequently joined and contributed to the discussions in the bi-weekly meeting used to steer the project. Furthermore, about reproducible workflows in the context of HPC environments and the use of container technology we have collaborated with members of the *DORIS*-Team (Stephan Hachinger (Leibniz Supercomputing Centre, LRZ), Marian Albers (RWTH Aachen)) and Jan Linxweiler (TU Braunschweig, project SURESOFT) with the aim of creating synergies between the different projects.

Early in the project, contributions by the community were made possible and encouraged through the publicly accessible repository on GitHub. Moreover, the developers of the investigated WfMSs were contacted to give feedback and enable exchange. The project was also mentioned in the Guix-HPC Activity Report 2022, to increase visibility.

3.5 References

Diercks, P., Gläser, D., Lünsdorf, O., Selzer, M., Flemisch, B., & Unger, J. F. (2022). Evaluation of tools for describing, reproducing and reusing scientific workflows. *ing.grid*. Retrieved from https://preprints.inggrid.org/repository/view/5/

Diercks, P., Gläser, D., Unger, J. F., & Flemisch, B. (2022). *NFDI4Ing HPC Workflows*. Von https://github.com/BAMresearch/NFDI4IngHPCWorkflows abgerufen

Diercks, P., Gläser, D., Unger, J. F., & Flemisch, B. (2022). *NFDI4Ing Scientific Workflow Requirements*. Von https://github.com/BAMresearch/NFDI4IngScientificWorkflowRequirements abgerufen

Wilkinson, M. D. (2016). The FAIR Guiding Principles for scientific data management and stewardship. *Sci Data, 3*(1). doi:10.1038/sdata.2016.18

3.6 Acknowledgements

The authors (P. Diercks, D. Gläser, O. Lünsdorf, M. Selzer, B. Flemisch and J. F. Unger) would like to thank the Federal Government and the Heads of Government of the Länder, as well as the Joint Science Conference (GWK), for their funding and support within the framework of the NFDI4Ing consortium. Funded by the German Research Foundation (DFG) - project number 442146713.