Exposé for a Studienprojekt Scientific Workflow Partitioning for Federated Execution

Felix Kummer, Supervised by Fabian Lehmann

Humboldt-Universität zu Berlin

1 Introduction

Scientific workflows have become an important part of different areas of research [1]. These workflows consist of tasks and dependencies. In particular, tasks may require inputs from other tasks, external data sources, or both. Scientific workflows are often depicted as directed acyclic graphs (DAGs). In these DAGs, nodes and edges represent tasks and dependencies, respectively. Building on this DAG representation, scientific workflow management systems (SWMSs) - such as Nextflow [2] - enable the specification and execution of scientific workflows. SWMSs instantiate abstract tasks with concrete tool invocations to derive physical DAGs. These physical DAGs may be static (i.e. fully know a priori) or dynamic (i.e. created at runtime).
When a workflow task finishes, the SWMS may schedule the execution of dependent tasks by traversing the DAG. However, the physical execution of these tasks is conducted by an executor. Executors include local execution, cloud execution, provisioned virtual machines at remote sites, and sophisticated resource managers like SLURM [3] or Kubernetes [4].
Advances in storage capacities enable the provision of large datasets. Modern scientific workflows leverage these input datasets to increase their scopes [5]. Large datasets are particularly common in remote sensing workflows (e.g. the Sentinel mission publishes multiple TiBs per day [6]). Remote sensing datasets are partially or fully hosted by a variety of heterogeneous institutional and commercial providers [7]. The execution of large scale scientific workflows will, therefore, involve the integration of multiple remotely hosted datasets.
State-of-the-art SWMSs and resource managers are limited to operate on a single cluster of compute resources [2,3,4]. Thus, remotely hosted input data for scientific workflows must be transferred to a single site. This introduces a plethora of problems. Specifically, transferring data from multiple remote sources to a local site imposes long wait times and large network loads. Moreover, complete datasets might not fit into the locally available storage capacities or become outdated.
These observations imply the need for novel federated scientific workflow systems (FSWSs). FSWSs have to deal with (1) gathering information on available datasets and compute capabilities at certain sites, (2) partitioning scientific workflows to multiple sites, and (3) orchestrating data transfers and distributed

workflow task execution.
In this study project, we will develop a prototype for federated workflow execution. Specifically, we focus our efforts on the workflow partitioning problem for static DAGs. Workflow partitioning can be viewed as finding a task-to-site mapping that optimises a set of goals (e.g. minimizing total execution time, monetary costs or network load). During the design and development process, we seek to establish new requirements and open challenges that need to be tackled in order to arrive at ready-to-use FSWSs. The prototype is not conceived as an optimal solution to the diverse challenges of federated workflow execution, but should rather establish a first baseline for future work. Additionally, we will develop an evaluation model for workflow partitioning strategies.

2 Related Work

Scientific workflows and different aspects of designing, executing, and monitoring them have been subject to numerous studies [5]. The need for the federated execution of scientific workflows has recently emerged as a consequence of increased data availability. Results of prior studies on scientific workflows will inevitably influence the research in this novel field. Nevertheless, the federate execution of scientific workflows requires the adaption to new challenges and requirements. There is little prior research on federated workflow execution. However, other areas of research have faced similar challenges and thus, might facilitate the development and study of FSWSs.
These related areas of research include distributed and federated query processing [8,9,10], cloud computing [11,12] and service-oriented architectures [13,14] [15,16]. We note that these areas of research share the challenges of processing data on distributed sites. However, federated scientific workflow execution entails a unique and novel combination of challenges such as large and distributed datasets, non-standard data models, heterogeneous sites, heterogeneous workflow tasks, complex dependencies, orchestration of concurrent and distributed executions, and the potential need of workflow partitioning.
Some of these challenges have been targeted individually in the past (e.g. [17,18] [19,20]). Nevertheless, a complete solution for federated scientific workflow execution is not existent as of writing this report.

3 Prototyping a Federated Scientific Workflow System

In this work, we plan to develop a prototype for FSWSs' workflow partitioning. The purpose of this prototype is neither to be an optimal solution nor to provide the functionality of a fully developed FSWS. We aim to investigate the possibilities of implementing and evaluating workflow partitioning strategies. This study should provide a staging ground for future work on workflow partitioning alongside a baseline for comparison with sophisticated partitioning approaches. We will focus our efforts on workflow partitioning and replace other FSWS functionality with placeholders.

In the initial phase, we plan to identify a framework to integrate our prototype into. The preliminary options are:

- Build the system from scratch without third-party software. This scenario would incorporate many placeholders and have limited usability beyond the scope of this study.
- Integrate the functionality into Nextflow [2]. Nextflow is currently not capable of supporting multiple compute sites. Hence, major changes to its source code would be necessary. However, we note that the previously developed Common Workflow Scheduler [21] could be leveraged to ease this task.
- Build the prototype on top of the ExaWorks software stack [20]. ExaWorks provides functionality for the distributed execution of workflow tasks and can be extended with workflow partitioning.
- Integrate workflow partitioning into the Workflow Description Language[1][2] (WDL). WDL is designed for readability and simplicity but lacks some features of Nextflow [22]. However, its simplicity will reduce the effort for integrating workflow partitioning.

To enable workflow partitioning, a set of sites that host certain datasets and provide compute capabilities is required. As part of this work, we aim to populate such a set with multiple virtual machines and compute clusters. Additionally, we will implement placeholder functionality that distributes site-wise information on hosted datasets and compute capabilities. Together, these components will form our testbed.
Partitioning of workflows tasks will require a cost model for executing a given task on a given site. Costs can incorporate a variety of factors. For this study project, we will develop a cost model that, at a minimum, incorporates the sizes of datasets that would need to be transferred to the site.
In the final phase of this study, we plan to implement a baseline workflow partitioning strategy. The baseline will likely be a random task distribution that exploits site-wise information on hosted datasets and the cost model.
Sophisticated workflow partitioning strategies will likely incorporate additional information such as compute capabilities and bandwidth between sites. Therefore, we will investigate the possibility of providing this information for future work.

4 Evaluation

Research Questions

This study has an exploratory character. Thus, we plan to conduct a qualitative evaluation. We address the following research questions:

RQ1 How can we develop a prototype for workflow partitioning in FSWSs and which challenges exist?
RQ2 How can we evaluate the performance of an FSWS?

[1] `https://github.com/openwdl/wdl/tree/main`
[2] `https://openwdl.org/`

Setup

We will evaluate the partitioning and execution of our prototype for a selected set of scientific workflows. The workflows will be synthetically assembled to enable systematic evaluation of workflow partitioning. Workflow tasks will involve the connection to the site, potential data transfers, and initiating the execution. In order to evaluate the performance of our prototype and future partitioning strategies, we will develop performance metrics that incorporate time measures for on-site execution and inter-site data transfers.
Compute sites will be simulated through a set of virtual machines with access to storage capacities. Additionally, we will investigate the integration of institutional clusters under Kubernetes management into our testbed.
We plan to simulate distributed data sources by deploying different data distribution strategies to host datasets on certain sites.

References

1. R. F. da Silva, H. Casanova, K. Chard, D. Laney, D. Ahn, S. Jha, C. Goble, L. Ramakrishnan, L. Peterson, B. Enders, *et al.*, "Workflows community summit: Bringing the scientific workflows community together," *arXiv preprint arXiv:2103.09181*, 2021.
2. P. Di Tommaso, M. Chatzou, E. W. Floden, P. P. Barja, E. Palumbo, and C. Notredame, "Nextflow enables reproducible computational workflows," *Nature biotechnology*, vol. 35, no. 4, pp. 316–319, 2017.
3. A. B. Yoo, M. A. Jette, and M. Grondona, "Slurm: Simple linux utility for resource management," in *Workshop on job scheduling strategies for parallel processing*, pp. 44–60, Springer, 2003.
4. B. Burns, B. Grant, D. Oppenheimer, E. Brewer, and J. Wilkes, "Borg, omega, and kubernetes," *Communications of the ACM*, vol. 59, no. 5, pp. 50–57, 2016.
5. J. Liu, E. Pacitti, P. Valduriez, and M. Mattoso, "A survey of data-intensive scientific workflow management," *Journal of Grid Computing*, vol. 13, pp. 457–493, 2015.
6. A. G. Castriotta, "Copernicus sentinel data access annual report," 2022. URL: `https://scihub.copernicus.eu/twiki/pub/SciHubWebPortal/AnnualReport2021/COPE-SERCO-RP-22-1312_-_Sentinel_Data_Access_Annual_Report_Y2021_merged_v1.1.pdf`, Acessed: 2023-11-05.
7. V. C. Gomes, G. R. Queiroz, and K. R. Ferreira, "An overview of platforms for big earth observation data management and analysis," *Remote Sensing*, vol. 12, no. 8, p. 1253, 2020.
8. C. T. Yu and C. Chang, "Distributed query processing," *ACM computing surveys (CSUR)*, vol. 16, no. 4, pp. 399–433, 1984.
9. D. Kossmann, "The state of the art in distributed query processing," *ACM Computing Surveys (CSUR)*, vol. 32, no. 4, pp. 422–469, 2000.
10. A. Schwarte, P. Haase, K. Hose, R. Schenkel, and M. Schmidt, "Fedx: Optimization techniques for federated query processing on linked data," in *The Semantic Web–ISWC 2011: 10th International Semantic Web Conference, Bonn, Germany, October 23-27, 2011, Proceedings, Part I 10*, pp. 601–616, Springer, 2011.

11. M. D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis, and A. Vakali, "Cloud computing: Distributed internet computing for it and scientific research," *IEEE Internet computing*, vol. 13, no. 5, pp. 10–13, 2009.
12. J. Wang, P. Korambath, I. Altintas, J. Davis, and D. Crawl, "Workflow as a service in the cloud: architecture and scheduling algorithms," *Procedia computer science*, vol. 29, pp. 546–556, 2014.
13. M. P. Papazoglou and W.-J. Van Den Heuvel, "Service oriented architectures: approaches, technologies and research issues," *The VLDB journal*, vol. 16, pp. 389–415, 2007.
14. N. Mahmoudi, C. Lin, H. Khazaei, and M. Litoiu, "Optimizing serverless computing: Introducing an adaptive function placement algorithm," in *Proceedings of the 29th Annual International Conference on Computer Science and Software Engineering*, pp. 203–213, 2019.
15. D. Agrawal, S. Chawla, B. Contreras-Rojas, A. Elmagarmid, Y. Idris, Z. Kaoudi, S. Kruse, J. Lucas, E. Mansour, M. Ouzzani, *et al.*, "Rheem: enabling cross-platform data processing: may the big data be with you!," *Proceedings of the VLDB Endowment*, vol. 11, no. 11, pp. 1414–1427, 2018.
16. S. Kruse, Z. Kaoudi, J.-A. Quiané-Ruiz, S. Chawla, F. Naumann, and B. Contreras-Rojas, "Optimizing cross-platform data movement," in *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pp. 1642–1645, IEEE, 2019.
17. M. J. Rosa, C. G. Ralha, M. Holanda, and A. P. Araujo, "Computational resource and cost prediction service for scientific workflows in federated clouds," *Future Generation Computer Systems*, vol. 125, pp. 844–858, 2021.
18. Z. Wen, J. Cala, and P. Watson, "A scalable method for partitioning workflows with security requirements over federated clouds," in *2014 IEEE 6th International Conference on Cloud Computing Technology and Science*, pp. 122–129, IEEE, 2014.
19. S. Abdi, L. PourKarimi, M. Ahmadi, and F. Zargari, "Cost minimization for bag-of-tasks workflows in a federation of clouds," *The Journal of Supercomputing*, vol. 74, pp. 2801–2822, 2018.
20. A. Al-Saadi, D. H. Ahn, Y. Babuji, K. Chard, J. Corbett, M. Hategan, S. Herbein, S. Jha, D. Laney, A. Merzky, *et al.*, "Exaworks: Workflows for exascale," in *2021 IEEE Workshop on Workflows in Support of Large-Scale Science (WORKS)*, pp. 50–57, IEEE, 2021.
21. F. Lehmann, J. Bader, F. Tschirpke, L. Thamsen, and U. Leser, "How workflow engines should talk to resource managers: A proposal for a common workflow scheduling interface," *arXiv preprint arXiv:2302.07652*, 2023.
22. A. E. Ahmed, J. M. Allen, T. Bhat, P. Burra, C. E. Fliege, S. N. Hart, J. R. Heldenbrand, M. E. Hudson, D. D. Istanto, M. T. Kalmbach, *et al.*, "Design considerations for workflow management systems use in production genomics research and the clinic," *Scientific reports*, vol. 11, no. 1, p. 21680, 2021.