Exposé for a Master Thesis
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 [1,2]. Consuming input data, scientific workflows execute a interdependent set of data processing tasks.
Advances in storage capacities enable scientific workflows to leverage increasingly large input datasets [3]. Large datasets are particularly common in remote sensing workflows (e.g. the Sentinel mission publishes multiple TiBs per day [4]). Remote sensing datasets are partially or fully hosted by a variety of heterogeneous institutional and commercial providers [5]. The execution of data-intensive scientific workflows can, therefore, involve the integration of multiple remotely hosted datasets.
State-of-the-art scientific workflow management systems (SWMSs) and resource managers are limited to operate on a single cluster of compute resources [6,7,8]. Thus, remotely hosted input data need to be transferred to a single site to start the execution of a scientific workflow. 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 Moreover, complete datasets might not fit into the locally available storage capacities or become outdated.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) , managing 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 execution.
One of the major shortcomings of previous research on scientific workflows is the assumptions of static workflows (i.e. all task instances are known a priori). This may be a desirable property for certain types of theoretical workflow analysis but does not reflect the reality in modern SWMSs [9].
In this master's thesis, as a first step towards FSWSs, we aim to design and evaluate a dynamic partitioning strategy for scientific workflows. To this end, we will establish a cost model with an emphasis on the costs of transferring data. Additional problems we need to solve include determining when to recompute the current partitioning and how to apply a new partitioning to a running workflow (e.g. how do deal with running tasks.)

2 Related Work

Scientific workflows and different aspects of designing, executing, and monitoring them have been subject to numerous studies [2,3]. The need for the federated execution of scientific workflows has recently emerged as a consequence of increased data sizes. 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.

One such area of research is the distributed or federated processing of database queries [10,11,12]. A shared challenge with federated workflow execution is the processing data in a distributed setup and the problem of deciding if, when and where to transfer data. However, the data model in query processing is fully known a priori, enabling a simple record-based partitioning. In contrast, scientific workflows incorporate an arbitrary number of potentially non-standard data models. Another difference is that query processing relies on a well-known set of relational operators, while workflow tasks can range from simple bash scripts to complex scientific tools and are thus treated as black boxes.

The partitioning of scientific workflows has previously received attention [13,14]. A key driver for this research was the consideration of security aspects. Specifically, prior work aims to partition scientific workflows with respect to different security levels of data and compute sites [15,16,17,18,19,20]. In contrast, our work focuses on data transfer reduction and workflows without security critical input data. This is supported by open data policies in certain targeted fields of science (e.g. open data policies in remote sensing [21]). Other previous work on workflow partitioning assumes static parameters for sites and workflows [15], optimizes for makespan time [22] or assumes tree-shaped DAGs [23].

Dynamic partitioning of scientific workflows has received little prior attention. Wen et al. proposes a dynamic re-partitioning approach [19]. However, here, the dynamicity does not refer to the workflow but the cloud environment. In particular, their approach reacts to cloud failures or changes in the pricing models whereas we target dynamic workflows.

Other related areas of research include cost models for multisite execution [24,25,26], multi-cloud research [26,27,28,29,16,24,25,30], cloud federation [29,24], scheduling of scientific workflows [31,32,33,34], scheduling that aims to reduce data transfers [35,36,37,38], and data placement [39,40].

In the preceding study project [41], we developed a parser application that derives the DAG from a scientific workflow defined in the Workflow Description Language[1] (WDL). Additionally, we manually partitioned a static scientific workflow, and prototyped an evaluation environment for the partitioning and federated execution of scientific workflows.

[1] `https://github.com/openwdl`

3 Dynamic Partitioning

Cost Model

Comparing different partitionings relative to a given set of task characteristics, task dependencies, available sites, and their resources requires a cost model. In this work, we aim to establish a cost model that incorporates the costs for computations, storing data, and data transfers. We will focus on the cost of data transfers incurred by handling large and distributed datasets.
A key challenge in our work will be the dynamicity of the workflow. Specifically, at the start of the execution, it may be unknown how many physical tasks will be instantiated from a single abstract task and which data will be required for a single task. Therefore, we will design heuristics to estimate the expected costs of task execution on a given site. Aligning with the dynamicity of the workflow, we will continuously improve the cost model when new information becomes available about physical tasks.

Partitioning

We define a static partitioning of a scientific workflow as the mapping of all physical tasks to compute sites. Static partitionings require knowledge of all physical tasks. However, due to the dynamicity in our scenario, we can not assume to have that knowledge a priori. To solve this problem, we propose dynamic partitioning of scientific workflows. Here, knowledge about physical tasks only becomes available gradually. A dynamic workflow partitioning incorporates new knowledge of physical tasks to update the task assignments. This update process will involve the computation of the total cost of the current partitioning and the search for cheaper alternative partitionings.
Our approach will start with the computation of an initial partitioning of the scientific workflow. Initial partitionings will be solely based on the abstract DAG and locations of the input data. This reveals another challenge for our approach. Namely, we partition physical tasks of a scientific workflow without any knowledge about the number and concrete data dependencies of these tasks. To solve this, our partitioning will contain a combination of physical tasks assignments and conditional assignments. Conditional assignments are evaluated when a related physical task becomes available and return a concrete compute site.
Based on the initial partitioning, our approach will continuously evaluate new information on physical tasks and their dependencies to update the cost model and reevaluate the partitioning. The core questions in this process will be when to reevaluate the partitioning and how to incorporate new information into the cost model and the partitioning.

Implementation Aspects

Based on the parser application developed in previous work, we will develop a workflow partitioner. The partitioner reevaluates the current partitioning and

potentially computes a new partitioning.
The workflow rewriter reads the WDL file that describes the original workflow and gets a workflow partitioning. Based on these inputs, the rewriter creates a new WDL file that materializes the partitioning in the original workflow. Concretely, all workflow tasks are rewritten so that they connect to a remote site and execute the original tasks on that site. Moreover, we introduce dedicated tasks to organize data transfers. This setup enables us to use a local executor that orchestrates the federated execution of scientific workflows.

4 Evaluation

In the preceding study project, we created an evaluation environment for the partitioning of scientific workflows. The environment consists of 5 virtual machines with modifiable network properties between the sites. In this work, we will reuse and extend the environment if needed.
We will extend WfCommons'[2] WfBench[3] to generate synthetic WDL workflows for our evaluation.
In addition, we will investigate the feasibility of switching our prototype to a simulation mode to accelerate the evaluation process.

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. E. Deelman, T. Peterka, I. Altintas, C. D. Carothers, K. K. Van Dam, K. Moreland, M. Parashar, L. Ramakrishnan, M. Taufer, and J. Vetter, "The future of scientific workflows," *The International Journal of High Performance Computing Applications*, vol. 32, no. 1, pp. 159–175, 2018.
3. 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.
4. 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.
5. 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.
6. 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.

[2] `https://wfcommons.org/`
[3] `https://github.com/wfcommons/WfCommons/tree/main/wfcommons/wfbench`

7. 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.
8. 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.
9. R. Ferreira da Silva, R. Filgueira, I. Pietri, M. Jiang, R. Sakellariou, and E. Deelman, "A characterization of workflow management systems for extreme-scale applications," *Future Generation Computer Systems*, vol. 75, pp. 228–238, Oct. 2017.
10. C. T. Yu and C. Chang, "Distributed query processing," *ACM computing surveys (CSUR)*, vol. 16, no. 4, pp. 399–433, 1984.
11. D. Kossmann, "The state of the art in distributed query processing," *ACM Computing Surveys (CSUR)*, vol. 32, no. 4, pp. 422–469, 2000.
12. 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.
13. W. Chen and E. Deelman, "Partitioning and scheduling workflows across multiple sites with storage constraints," in *Parallel Processing and Applied Mathematics: 9th International Conference, PPAM 2011, Torun, Poland, September 11-14, 2011. Revised Selected Papers, Part II 9*, pp. 11–20, Springer, 2012.
14. J. Liu, V. Silva, E. Pacitti, P. Valduriez, and M. Mattoso, "Scientific workflow partitioning in multisite cloud," in *Euro-Par 2014: Parallel Processing Workshops: Euro-Par 2014 International Workshops, Porto, Portugal, August 25-26, 2014, Revised Selected Papers, Part I 20*, pp. 105–116, Springer, 2014.
15. 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.
16. E. Goettelmann, W. Fdhila, and C. Godart, "Partitioning and cloud deployment of composite web services under security constraints," in *2013 IEEE International Conference on Cloud Engineering (IC2E)*, pp. 193–200, IEEE, 2013.
17. P. Watson, "A multi-level security model for partitioning workflows over federated clouds," *Journal of Cloud Computing: Advances, Systems and Applications*, vol. 1, pp. 1–15, 2012.
18. Z. Wen, J. Cała, P. Watson, and A. Romanovsky, "Cost effective, reliable and secure workflow deployment over federated clouds," *IEEE Transactions on Services Computing*, vol. 10, no. 6, pp. 929–941, 2016.
19. Z. Wen, R. Qasha, Z. Li, R. Ranjan, P. Watson, and A. Romanovsky, "Dynamically partitioning workflow over federated clouds for optimising the monetary cost and handling run-time failures," *IEEE Transactions on Cloud Computing*, vol. 8, no. 4, pp. 1093–1107, 2016.
20. Z. Wen and P. Watson, "Dynamic exception handling for partitioned workflow on federated clouds," in *2013 IEEE 5th International Conference on Cloud Computing Technology and Science*, vol. 1, pp. 198–205, IEEE, 2013.
21. C. E. Woodcock, R. Allen, M. Anderson, A. Belward, R. Bindschadler, W. Cohen, F. Gao, S. N. Goward, D. Helder, E. Helmer, *et al.*, "Free access to landsat imagery.," *SCIENCE VOL 320: 1011*, 2008.
22. W. Chen and E. Deelman, "Integration of workflow partitioning and resource provisioning," in *2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*, pp. 764–768, IEEE, 2012.

23. S. Kulagina, H. Meyerhenke, and A. Benoit, "Mapping tree-shaped workflows on memory-heterogeneous architectures," in *European Conference on Parallel Processing*, pp. 158–170, Springer, 2022.
24. 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.
25. E. N. Alkhanak, S. P. Lee, R. Rezaei, and R. M. Parizi, "Cost optimization approaches for scientific workflow scheduling in cloud and grid computing: A review, classifications, and open issues," *Journal of Systems and Software*, vol. 113, pp. 1–26, 2016.
26. H. M. Fard, R. Prodan, and T. Fahringer, "A truthful dynamic workflow scheduling mechanism for commercial multicloud environments," *IEEE Transactions on Parallel and Distributed systems*, vol. 24, no. 6, pp. 1203–1212, 2012.
27. M. Masdari and M. Zangakani, "Efficient task and workflow scheduling in inter-cloud environments: challenges and opportunities," *The Journal of Supercomputing*, vol. 76, no. 1, pp. 499–535, 2020.
28. J. Diaz-Montes, M. Diaz-Granados, M. Zou, S. Tao, and M. Parashar, "Supporting data-intensive workflows in software-defined federated multi-clouds," *IEEE Transactions on Cloud Computing*, vol. 6, no. 1, pp. 250–263, 2015.
29. R. d. C. Coutinho, L. M. Drummond, Y. Frota, and D. de Oliveira, "Optimizing virtual machine allocation for parallel scientific workflows in federated clouds," *Future Generation Computer Systems*, vol. 46, pp. 51–68, 2015.
30. J. Liu, E. Pacitti, P. Valduriez, D. De Oliveira, and M. Mattoso, "Multi-objective scheduling of scientific workflows in multisite clouds," *Future Generation Computer Systems*, vol. 63, pp. 76–95, 2016.
31. I. Ahmad and Y.-K. K. Y.-K. Kwok, "A new approach to scheduling parallel programs using task duplication," in *1994 Internatonal Conference on Parallel Processing Vol. 2*, vol. 2, pp. 47–51, IEEE, 1994.
32. B. Kruatrachue and T. Lewis, "Grain size determination for parallel processing," *IEEE software*, vol. 5, no. 1, pp. 23–32, 1988.
33. H. Topcuoglu, S. Hariri, and M.-Y. Wu, "Performance-effective and low-complexity task scheduling for heterogeneous computing," *IEEE transactions on parallel and distributed systems*, vol. 13, no. 3, pp. 260–274, 2002.
34. G. P. Rodrigo, E. Elmroth, P.-O. Östberg, and L. Ramakrishnan, "Enabling workflow-aware scheduling on hpc systems," in *Proceedings of the 26th International Symposium on High-Performance Parallel and Distributed Computing*, pp. 3–14, 2017.
35. I. Pietri and R. Sakellariou, "Scheduling data-intensive scientific workflows with reduced communication," in *Proceedings of the 30th International Conference on Scientific and Statistical Database Management*, pp. 1–4, 2018.
36. P. Bryk, M. Malawski, G. Juve, and E. Deelman, "Storage-aware algorithms for scheduling of workflow ensembles in clouds," *Journal of Grid Computing*, vol. 14, pp. 359–378, 2016.
37. P. Donnelly, N. Hazekamp, and D. Thain, "Confuga: scalable data intensive computing for posix workflows," in *2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 392–401, IEEE, 2015.
38. M. Tanaka and O. Tatebe, "Workflow scheduling to minimize data movement using multi-constraint graph partitioning," in *2012 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012)*, pp. 65–72, IEEE, 2012.

39. Q. Zhao, C. Xiong, X. Zhao, C. Yu, and J. Xiao, "A data placement strategy for data-intensive scientific workflows in cloud," in *2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 928–934, IEEE, 2015.
40. Ü. V. Çatalyürek, K. Kaya, and B. Uçar, "Integrated data placement and task assignment for scientific workflows in clouds," in *Proceedings of the fourth international workshop on Data-intensive distributed computing*, pp. 45–54, 2011.
41. F. Kummer, "Scientific workflow partitioning for federated execution," 2024. Study project, Humboldt-Universität zu Berlin.