A Conceptual framework supporting pattern design selection for scientific workflow applications in cloud computing

Al-Khannaq, Ehab Nabiel; Khan, Saif Ur Rehman; Verbraeck, Alexander; Van Lint, Hans

Publication date
2020

Document Version
Final published version

Published in
CLOSER 2020 - Proceedings of the 10th International Conference on Cloud Computing and Services Science

Citation (APA)
Al-Khannaq, E. N., Khan, S. U. R., Verbraeck, A., & Van Lint, H. (2020). A Conceptual framework supporting pattern design selection for scientific workflow applications in cloud computing. In D. Ferguson, M. Helfert, & C. Pahl (Eds.), *CLOSER 2020 - Proceedings of the 10th International Conference on Cloud Computing and Services Science* (pp. 229-236). (CLOSER 2020 - Proceedings of the 10th International Conference on Cloud Computing and Services Science). SciTePress.

A Conceptual Framework Supporting Pattern Design Selection for Scientific Workflow Applications in Cloud Computing

Ehab Nabiel Alkhanak[1][a], Saif Ur Rehman Khan[2][b], Alexander Verbraeck[3][c] and Hans van Lint[1][d]

[1]*Transport and Planning Department, Faculty of Civil Engineering and Geosciences (CiTG), Delft University of Technology (TU Delft), Delft, The Netherlands*

[2]*Department of Computer Science, COMSATS University Islmabad (CUI), Islamabad, Pakistan*

[3]*Department Multi-actor Systems, Faculty of Technology, Policy and Management (TPM), Delft University of Technology (TU Delft), Delft, The Netherlands*

Keywords: Scientific Workflow Application, Workflow Management System, Design Patterns, Cloud Computing Environment.

Abstract: Scientific Workflow Applications (SWFA) play a vital role for both service consumers and service providers in designing and implementing large and complex scientific processes. Previously, researchers used parallel and distributed computing technologies, such as utility and grid computing to execute the SWFAs, these technologies provide limited utilization for the shared resources. In contrast, the scalability and flexibility challenges are better handled by using cloud-computing technologies for SWFA. Since cloud computing offers a technology that can significantly utilize the amounts of storage space and computing resources necessary for processing large-size and complex SWFAs. The workflow pattern design has provided the facility of re-using previously developed workflow solutions that enable the developers to adopt them for the considered SWFA. Inspired by this, the researchers have adopted several patterns of design to better design the SWFA. Effective pattern design that can consider challenges that may not become visible only in the implementation stage of a SWFA. However, the selection of the most effective pattern design in accordance with an execution method, data size, and problem complexity of a SWFA remains a challenging task. Motivated by this, we have proposed a conceptual framework that facilitates in recommending a suitable pattern design based on the quality requirements and capabilities are given and advertised by cloud consumers and providers, respectively. Finally, guidelines to assist in a smooth migrating of SWFA from other computation paradigms to cloud computing.

1 INTRODUCTION

The workflow has been defined by Workflow Management Coalition (WfMC) as "The automation of a business process, in whole or part, during which documents, information or tasks are passed from one participant to another for action, according to a set of procedural rules" (Hollingsworth and Hampshire, 1993). In this research context, a workflow can be defined as a series of processes and these processes represent one or several computational tasks based on their dependencies. These computational tasks can be any executable instances (e.g., load sets, report sets, programs, and data) with different structures (e.g., jobs, pipeline, data distribution, data aggregation, and data redistribution). Generally speaking, a Workflow Application (WFA) acts like software that automatically handles scientific or business-related jobs by creating and initiating these instances while the Workflow Management Service offers a certain API to facilitate the management of these processes. Historically, researchers have emphasized workflow applications for a business domain (e.g., banking systems, transaction systems). Developers need to consider large-scale, fault-tolerant, complex, and maintainable scientific processes when they need to deal with scientific workflows. Some of the main application areas of Scientific Workflow Applications (SWFA) are Bioinformatics, Geoinformatics, Cheminformatics, Biomedical Informatics, and Astrophysics (Pan et al., 2019).

[a] https://orcid.org/0000-0002-9880-6365
[b] https://orcid.org/0000-0002-9643-6858
[c] https://orcid.org/0000-0002-1572-0997
[d] https://orcid.org/0000-0003-1493-6750

Alkhanak, E., Khan, S., Verbraeck, A. and van Lint, H.
A Conceptual Framework Supporting Pattern Design Selection for Scientific Workflow Applications in Cloud Computing.
DOI: 10.5220/0008916102290236
In *Proceedings of the 10th International Conference on Cloud Computing and Services Science (CLOSER 2020)*, pages 229-236
ISBN: 978-989-758-424-4

Scientific Workflow Management Systems (SWFMS) like Pegasus SWFMS (Deelman et al., 2015) comprise a set of components (including Mapper, Local execution engine, Job scheduler, Monitoring component) that focus on automating the processes of large-size data. SWFMS provides a graphical environment that supports the reuse and integration of domain-related workflows. Ultimately, SWFMS facilitate scientists in performing their domain-specific workflows by analysing given dataset(s) and visualizing the result of the processing of the dataset(s).

Previously, researchers have used parallel and grid computing resources to execute SWFAs (Dong, 2009; Juve et al., 2013). Today's SWFAs generate a large amount of data and due to this huge increase in size and complexity, parallel and grid computing technologies are unable to fulfil the associated computational demands (i.e. vast amount of storage space, strict completion deadlines, etc) because these technologies are not scalable and unable to fully utilize the available recourse.

Consequently, researchers aim on providing methods for the design and realisation of scientific workflows and also setting up computational environments that provide significant storage space and powerful computational resources and most importantly the researchers aim to map the designed workflow components to existing computation environments. Cloud computing resources provide optimal resource utilization that best matches the demands of scientists by providing on-demand, scalable, and flexible solutions for considered SWFA (Juve and Deelman, 2011).

The workflow pattern design in cloud computing environments provides the facility of re-using previously developed workflow templates, which consequently enables the developer to use them to ensure better design of SWFA. In this manuscript, the authors focus on type level of workflow patterns and not on complete workflows which include the mapping of their task/components to actual scripts and services. Fig. 1 depicts three main stages of the SWFA pattern design lifecycle: (i) workflow design, (ii) designer interpretation, and (iii) cloud execution environment. In the workflow design stage, the scientists use previous workflows patters and thus effectively systematically reuse knowledge for controlled experimentation (Pan et al., 2019). After that, in the design interpretation stage, the parameters and data resources are determined based on the experimentation requirements. As tasks could be realised by different services/scripts which could operate on equivalent-typed but different in format and structure data sets. Thus, the interpretation stage simplified these complexities between the tasks by selecting the appropriate services/scripts which are depend on the selection of actual data resources. Notice that the determined selections will subsequently be used for the execution stage in the pattern design lifecycle. Finally, the main purpose of cloud execution environment stage is to first deploy the cloud computing environment that is able to execute the workflow. And based on the complete specification of the workflow that has been collected from previous two stages (workflow pattern selection, workflow authoring, development/selection of workflow components, etc.) the cloud resources consume and create the data of SWFA for the desired execution. Due to parameters dependences (that have been determined in designer interpretation stage) effect the task executions, thus all these tasks cannot be executed concurrently in the cloud execution environment. The runtime executions required a runtime monitoring component that could continuously monitors and informs the scientists about the current status of the SWFA execution. This monitoring functionality plays a crucial role about successful execution of SWFA. As overall, the pattern design stages are consisting of design, deployment and provisioning where the latter includes execution, monitoring and adaptation.

In the literature, a number of workflow pattern designs have been proposed to better design the SWFA (Ramakrishnan et al., 2011; Genez et al., 2012; Wu et al., 2013b). However, the selection of most efficient workflow pattern design remains a challenging task as it required to consider several aspects. The user requirements are the main aspects which include the make-span, reliability, deadline, and budget, while the other aspects are related to the particular execution method, data size, and problem complexity of a SWFA. In this paper, we propose a conceptual framework that suggest suitable workflow pattern designs based on user functional and non-functional requirements service consumers and quality constraints of service providers. The migration of SWFA from other computation paradigms to cloud computing requires considering several aspects and constraints. Thus, based on the literature we provide a set of guidelines to assist this transformation to fully utilize the strengths of cloud computing. Furthermore, a classification of potential workflow pattern design is proposed to guide the framework in selecting a most appropriate workflow pattern design for the considered SWFA.

The remainder of this paper is organized as follows: Section 2 presents our proposed conceptual framework. Section 3 provides a classification of reported pattern design for SWFAs. A set of guidelines to shift SWFA from different computing envi-

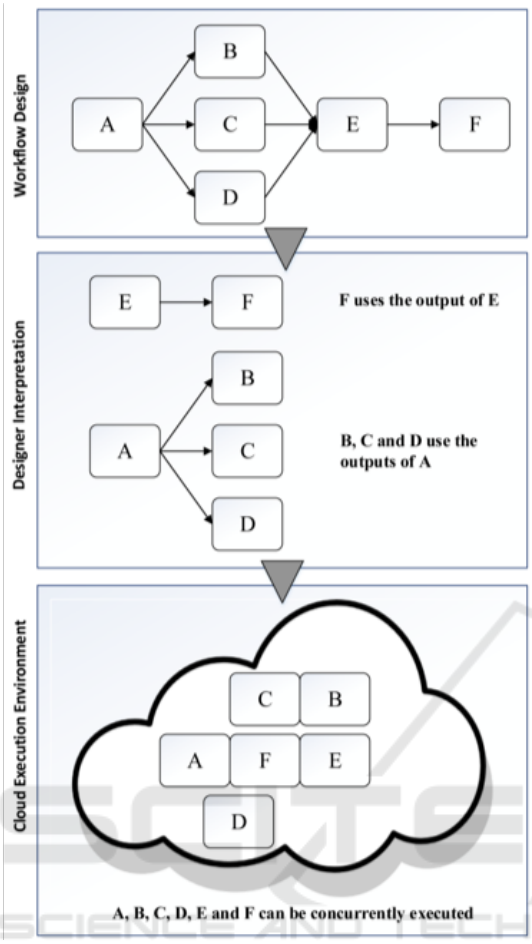

Figure 1: The Pattern Design Stages of Scientific Workflow Applications Lifecycle.

ronments to the cloud computing is provided in Section 4. Section 5 discusses previously conducted relevant works. Finally, Section VI concludes this paper and outlines a certain number of future research directions.

2 THE PROPOSED CONCEPTUAL FRAMEWORK

The utilization of cloud computing resources is depending on the respective application requirements and also relates to the resources being consumed by the WFM. Thus, there is a need for a lightweight WFMS that still offers the right functionality to its users and this can be achieved by selecting appropriate workflow pattern design. Motivated by this, we present our proposed conceptual framework for selecting workflow pattern design for SFWA in cloud computing. Figure 2 depicts the high-level layered architecture of our proposed conceptual framework. The proposed framework consists of four main layers, including cloud actor, cloud scientific workflow application, cloud use cases, and pattern design. The first layer represents the various cloud actors, which can be categorized based on the specified rules of the SWFA users, tasks and access rights. The main actors are: (i) Scientist: SWFA provides interactive tools to help scientists in better executing their own workflows and visualise the results in a real-time manner. (ii) System developer: or system administrator responsible for designing the WfMS based on the requirements of service consumers. (iii) Service vendor, and (iv) Resource owner: they are service providers (i.e. cloud service providers) that offer the SWFS system with virtualised computational resources (i.e. private, public or hybrid).

There are two types of interactions handled in the cloud environment using Application Programming Interfaces (APIs): (i) inter-interaction (i.e. the interaction between cloud actor and cloud SWFA), and (ii) intra-interaction (i.e. interaction between one cloud actor with another cloud actor). An example of interaction is inter-interaction between scientists and developer. The developers design a SWFA that could simplify the process for scientists to reuse the same workflows and provides an easy-to-use environment to track and share output results virtually.

The second layer manages the cloud scientific workflow application, and we distinguish two main types: (i) scientific applications, and (ii) management applications. The scientific application provides a user-friendly interface that enables cloud actors to interact with a scientific workflow. The management application handles all management related issues that may occur while executing the SWFAs. E.g., one application for the design of an SWFA and another for the execution of an SWFA. Note that developers continuously monitor the performance of the management application and perform maintenance related tasks to avoid any service degradation as well as the system could also perform some adaptations to the SWFAs. The third layer represents cloud use-cases. The use-cases depict different scenarios that may occur while the developer interacts with the cloud resources. Therefore, in the context of cloud use-cases, cloud computing offers different types of application paradigms. In the literature, cloud use cases are categorized into four classes, including application use cases, delivery of services use cases, deployment of application use cases, and interaction between the application and cloud services use cases based on user profiles (Petcu, 2010).

The fourth layer then encompasses pattern design.

The circles denote the cloud use-cases (with different colours and each colour maps to a different kind of cloud use case) and the arrows represent their dependencies. Thus, we could define workflow pattern design as a selection of the right use cases in order to instantiate a specific workflow management scenario. In other words, workflow pattern design results in a cluster of use cases on which specific dependencies are introduced. We consider that different use case kinds are covered and we try to select them in order to support the whole management of an SWFA.

The following are the constraints that need to be considered in order to determine suitable workflow pattern design for SWFA in a cloud computing environment:

- Identify the interaction between cloud actors and SWFA. This mapping is based on the requirements and preferences such an actor would have. These requirements and preferences are bound to the profile of that actor that is maintained by the proposed framework.
- Identify the real mapping between cloud actors and cloud use-cases.
- Identify the required computational requirement based on the nature and type of SWFA.
- Identify the values for the parameters and controlling mechanism based on the type of scheduling or optimization methods.
- Select the most suitable workflow pattern design from the classification list, which has been presented in the next section.

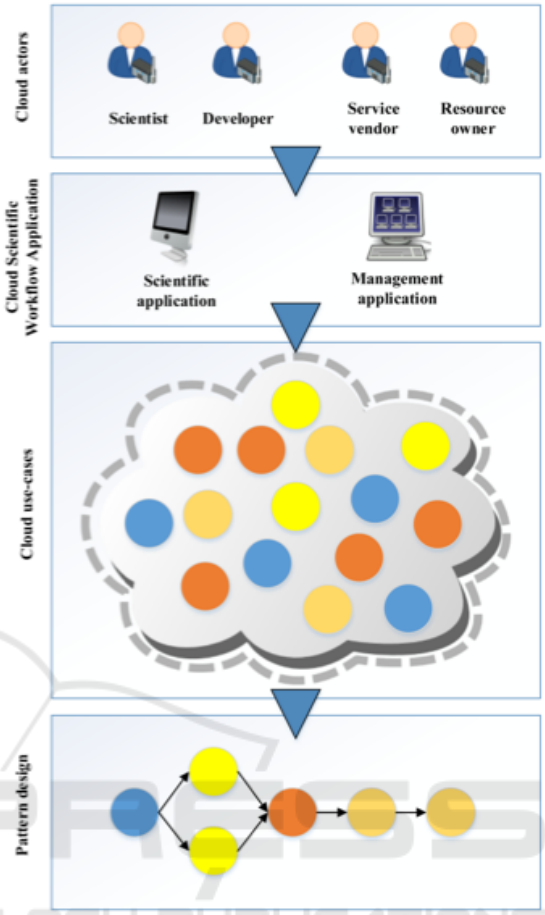

Figure 2: High-Level Layered Architecture of Proposed Framework.

3 CLASSIFICATION OF SCIENTIFIC WORKFLOW APPLICATIONS (SWFA) PATTERN DESIGN

The current state-of-the-art lacks in using different types of pattern designs for SWFA. The majority of the work focuses on control flow and workflow data pattern design only. However, (Kiepuszewski et al., 2003) reports that both control flow and workflow data pattern design is difficult to adopt and follow for SWFA due to implicitly defined control flow relations. In contrast, there are several useful but ignored pattern designs, which are effective to model and implement existing SWFAs. We have analysed and stated possible classification SWFA pattern designs based on the reviewed papers. Fig. 3 presents this categorization by presenting two levels of categories of SWFA design patterns. Our classification identifies four main categories. These main categories are: (i) control flow patterns, (ii) workflow data patterns, (iii) workflow resource patterns, and (iv) other scientific patterns. Each aspect is further categorized into several classes based on computational nature and use cases. It can be clearly seen that control flow pattern design has a greater number of sub-categories than any other aspect.

4 GUIDELINES TO ASSIST THE MIGRATION OF SWFA FROM OTHER COMPUTATION PARADIGMS TO CLOUD COMPUTING

Migration from one computational paradigm to another is never an easy task. In order to fully utilize the strengths of cloud computing, developers need to consider the following quality constraints (Figure 4)

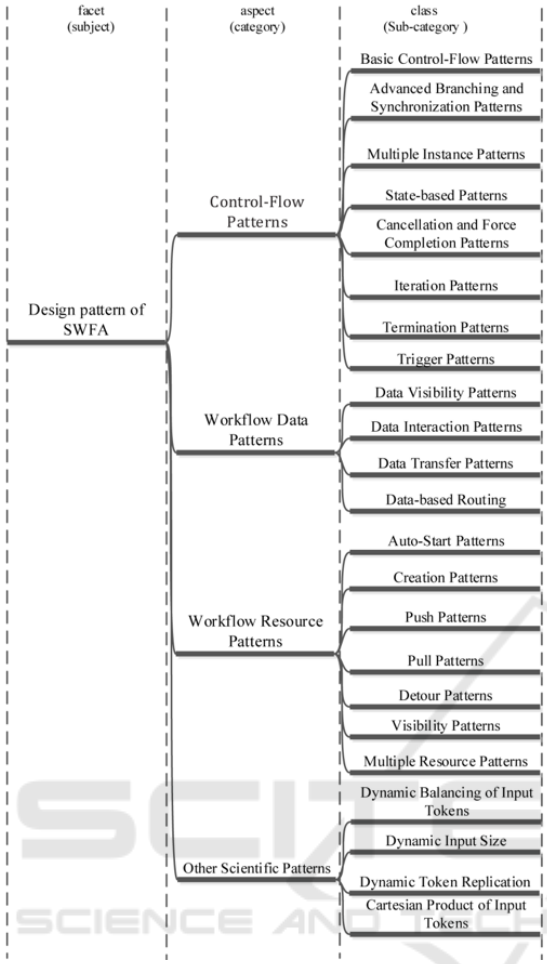

Figure 3: Classification of SWFA Pattern Design.

for successful transformation from other computing paradigms to cloud computing:

4.1 Demand Cloud Resource

Cloud actors can request and acquire cloud resources at any time according to their requirements to execute SWFA. This quality constraint provides flexibility in terms of problem size, budget, and Make-span. Consequently, it improves resource utilization by providing instant responsiveness to cloud user requests.

4.2 Elasticity

This refers to the on-demand response from cloud to actors. This quality constraint facilitates WFMS to manage the available resources according to SWFA demands. The use of popular workflow patterns (e.g., data distribution, data aggregation) enables it to fully utilize the available resources.

4.3 Legacy Applications

It contains a variety of software components for different types of cloud actors. The WFMS integrates these heterogeneous types of components into a single application. The legacy code can easily be executed using virtualization technology together with clouds.

4.4 Provenance and Reproducibility

This constraint refers to the metadata storage for the computation by keeping all traceability information about the origin and type of data using a particular computation.

4.5 Application Deployment

This quality constraint focuses on the usage of virtualization technology in Cloud computing environment dynamically pre-loaded and/or deployed onto a virtual machine (VM) based on the computational requirements.

4.6 Scalability

This is one of the most important quality constraints, especially when dealing with large-size data and resources. To efficiently deal with data-intensive and computer-intensive workflows, a necessary graphical environment is required to fully utilize the capabilities of cloud resources.

4.7 Reliability and Fault-tolerance

The cloud actors (mainly developer) have to explicitly specify the contingency actions (e.g., the strategies to deal with particular failure) to make the workflow more reliable against the unreliable environments.

Figure 4: Quality Constraints for SWFA Migration to Cloud Computing.

5 RELATED WORK

In the literature, several review studies relevant to this area of research have been performed. A decade ago, (Yu and Buyya, 2005) propose taxonomy by considering four aspects: (i) workflow design, (ii) workflow

scheduling, (iii) fault tolerance, and (iv) data movement to evaluate scientific WFMSs in terms of Grid computing, however, this classification is not suitable for selecting a most suitable workflow pattern design.

In contrast, (Petcu, 2010) focus on providing general pattern features, and re-using patterns from related architectures. In terms of application deployment reasoning, the authors provide a mechanism via which the most appropriate patterns are selected by considering a combination of virtualization technology of cloud and self-service facilitated application deployment. The definitions of patterns on different security levels are provided to handle the security of data services in Clouds. On the other hand, a general interaction pattern for service providers is recommended by (Deelman et al., 2015). Other previous work (Alkhanak et al., 2015) focused on cost-aware workflow scheduling challenges in cloud computing. In the previously conducted work, we have reported on various cost optimization aspects and parameters covered by workflow scheduling approaches necessary to understand the body of knowledge in this area of research.

However, to the best of our knowledge, no work has been done related to classifications or guidelines useful in selecting the most suitable workflow pattern design for SWFA in cloud computing. Table 1 illustrates state-of-the art scientific workflow application approaches.

From Table 1, all pattern design is control-flow based, and this shows that there is a need to utilize other types of workflow patterns design such as workflow data patterns, workflow resource patterns, and other scientific patterns. Also, it can be clearly observed from Table 1 that most of the proposed approaches have considered the Cloud/Grid provider as the cloud actor. This could be due to the profit that Cloud/Grid provider could gain from utilizing the pattern designs. Also, this shows there is a need to give more attention to the functional and non-functional requirements of the service consumer (i.e. researcher). In summary, our proposed conceptual framework and guidelines to assist the migration of SWFA from other computation paradigms to cloud computing could help both cloud actors.

Table 1: The State-of-the Art Scientific Workflow Application Approaches.

Reference	Pattern Design	Cloud Actor
(Wu et al., 2013b)	Control flow	Service consumer
(Abrishami and Naghibzadeh, 2012)	Control flow	Service consumer
(Ramakrishnan et al., 2011)	Control flow	Cloud/Grid provider
(Nargunam and Shajin, 2012)	Control flow	Cloud/Grid provider
(Wu et al., 2013a)	Control flow	Cloud/Grid provider
(Ostermann et al., 2010)	Control flow	Cloud/Grid provider
(Liu et al., 2011)	Control flow	Cloud/Grid provider
(Yang et al., 2008)	Control flow	Service consumer
(Genez et al., 2012)	Control flow	Service consumer
(Xu et al., 2009)	Control flow	Cloud/Grid provider
(Saeid Abrishami, 2013)	Control flow	Cloud/Grid provider
(Lin and Lu, 2011)	Control flow	Cloud/Grid provider
(Pandey et al., 2010)	Control flow	Cloud/Grid provider
(Tanaka and Tatebe, 2012)	Control flow	Cloud/Grid provider
(Bittencourt and Madeira, 2011)	Control flow	Cloud/Grid provider

6 CONCLUSION AND FUTURE WORK

Scientific Workflow Applications (SWFA) have significantly attained the researcher's attention since the advent of cloud computing. Cloud computing offers a technology that can significantly utilize the amounts of storage space and computing resources necessary for processing large-size and complex SWFAs. Currently, developers are facing numerous challenges while transforming SWFAs from other computational paradigms to cloud computing.

This paper presented a conceptual framework that can identify the interaction between cloud actors and SWFA. And also identify the real mapping between cloud actors and cloud use-cases which ultimately identify the required computational requirement based on the nature and type of SWFA. Furthermore, we devised a taxonomy for SWFA pattern design, which helps developers in selecting the

most suitable pattern design based on their requirements. Moreover, a set of guidelines is provided to effectively migrate SWFA from other computation paradigms to cloud computing. The pattern design for SWFA in cloud computing can certainly decrease development and maintenance cost and time. In the future, other types of workflow applications (e.g., business workflows) would be considered in a cloud computing environment. An extensive evaluation about the performance of proposed framework using a real-world scientific workflow application in a cloud platform would be conducted. Furthermore, we plan to make a connection between our conceptual framework and a WFMS by identifying the traditional architecture of a WFMS and which conceptual elements are exploited by which WFMS components such showcase can be exploited across the whole scientific workflow (management) lifecycle.

ACKNOWLEDGEMENTS

This work has been sponsored partially by the NWO/TTW project Multi-scale integrated Traffic Observatory for Large Road Networks (MiRRORS) under grant number 16270. This work is related to the PhD research by Dr. Ehab Al-Khannaq, sponsored by RG114-12ICT, supervised by Prof. Dr. Sai Peck Lee, and supported by Ministry of Education, Malaysia.

REFERENCES

Abrishami, S. and Naghibzadeh, M. (2012). Deadline-constrained workflow scheduling in software as a service cloud. *Scientia Iranica*, 19(3):680–689.

Alkhanak, E. N., Lee, S. P., and Khan, S. U. R. (2015). Cost-aware challenges for workflow scheduling approaches in cloud computing environments: Taxonomy and opportunities. *Future Generation Computer Systems*, 50:3–21.

Bittencourt, L. F. and Madeira, E. R. M. (2011). Hcoc: a cost optimization algorithm for workflow scheduling in hybrid clouds. *Journal of Internet Services and Applications*, 2(3):207–227.

Deelman, E., Vahi, K., Juve, G., Rynge, R. F., and Livny, M. (2015). Pegasus, a workflow management system for science automation. *Future Generation Computer Systems*, 46:17–35.

Dong, F. (2009). *Workflow scheduling algorithms in the grid*. PhD thesis.

Genez, T. A., Bittencourt, L. F., and Madeira, E. R. (2012). Workflow scheduling for saas/paas cloud providers considering two sla levels. In *2012 IEEE Network Operations and Management Symposium (NOMS)*, pages 906–912. IEEE.

Hollingsworth, D. and Hampshire, U. (1993). Workflow management coalition the workflow reference model. *Workflow Management Coalition*, 68.

Juve, G. and Deelman, E. (2011). Scientific workflows in the cloud. In *Grids, clouds and virtualization*, pages 71–91. Springer.

Juve, G., Rynge, M., Deelman, E., Vöckler, J.-S., and Berriman, G. B. (2013). Comparing futuregrid, amazon ec2, and open science grid for scientific workflows. *Computing in Science & Engineering*, 15(4):20–29.

Kiepuszewski, B., ter Hofstede, A. H., and van der Aalst, W. M. (2003). Fundamentals of control flow in workflows. *Acta Informatica*, 39(3):143–209.

Lin, C. and Lu, S. (2011). Scheduling scientific workflows elastically for cloud computing. In *2011 IEEE International Conference on Cloud Computing (CLOUD)*, pages 746–747. IEEE.

Liu, H., Xu, D., and Miao, H. (2011). Ant colony optimization based service flow scheduling with various qos requirements in cloud computing. In *2011 First ACIS International Symposium on Software and Network Engineering (SSNE)*, pages 53–58. IEEE.

Nargunam, K. L. G. and Shajin, A. (2012). Compatibility of hybrid process scheduler in green it cloud computing environment. *International Journal of Computer Applications*, 55(5):27–33.

Ostermann, S., Prodan, R., and Fahringer, T. (2010). Dynamic cloud provisioning for scientific grid workflows. In *2010 11th IEEE/ACM International Conference on Grid Computing (GRID)*, pages 97–104. IEEE.

Pan, S., Zhu, L., and Qiao, J. (2019). An open sharing pattern design of massive power big data. In *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, pages 5–9. IEEE.

Pandey, S., Wu, L., Guru, S. M., and Buyya, R. (2010). A particle swarm optimization-based heuristic for scheduling workflow applications in cloud computing environments. In *2010 24th IEEE International Conference on Advanced Information Networking and Applications (AINA)*, pages 400–407. IEEE.

Petcu, D. (2010). Identifying cloud computing usage patterns. In *2010 IEEE International Conference on Cluster Computing Workshops and Posters (CLUSTER WORKSHOPS)*, pages 1–8. IEEE.

Ramakrishnan, L., Chase, J. S., Gannon, D., Nurmi, D., and Wolski, R. (2011). Deadline-sensitive workflow orchestration without explicit resource control. *Journal of Parallel and Distributed Computing*, 71(3):343–353.

Saeid Abrishami, Mahmoud Naghibzadeh, D. E. (2013). Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds. *Future Generation Computer Systems*, 29(1):158–169.

Tanaka, M. and Tatebe, O. (2012). Workflow scheduling to minimize data movement using multi-constraint graph partitioning. In *Proceedings of the 2012 12th IEEE/ACM International Symposium on Cluster,*

Cloud and Grid Computing (ccgrid 2012), pages 65–72. IEEE Computer Society.

Wu, Q., Yun, D., Lin, X., Gu, Y., Lin, W., and Liu, Y. (2013a). On workflow scheduling for end-to-end performance optimization in distributed network environments. In *Job Scheduling Strategies for Parallel Processing*, pages 76–95. Springer.

Wu, Z., Liu, X., Ni, Z., Yuan, D., and Yang, Y. (2013b). A market-oriented hierarchical scheduling strategy in cloud workflow systems. *The Journal of Supercomputing*, 63(1):256–293.

Xu, M., Cui, L., Wang, H., and Bi, Y. (2009). A multiple qos constrained scheduling strategy of multiple workflows for cloud computing. In *Parallel and Distributed Processing with Applications, 2009 IEEE International Symposium on*, pages 629–634. IEEE.

Yang, Y., Liu, K., Chen, J., Liu, X., Yuan, D., and Jin, H. (2008). An algorithm in swindew-c for scheduling transaction-intensive cost-constrained cloud workflows. In *eScience'08. IEEE Fourth International Conference on eScience, 2008.*, pages 374–375. IEEE.

Yu, J. and Buyya, R. (2005). A taxonomy of scientific workflow systems for grid computing. *Sigmod Record*, 34(3):44–49.